Modeling Narrative Discourse

David K. Elson

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2012

ABSTRACT

Modeling Narrative Discourse

David K. Elson

This thesis describes new approaches to the formal modeling of narrative discourse. Although narratives of all kinds are ubiquitous in daily life, contemporary text processing techniques typically do not leverage the aspects that separate narrative from expository discourse. We describe two approaches to the problem. The first approach considers the conversational networks to be found in literary fiction as a key aspect of discourse coherence; by isolating and analyzing these networks, we are able to comment on longstanding literary theories. The second approach proposes a new set of discourse relations that are specific to narrative. By focusing on certain key aspects, such as agentive characters, goals, plans, beliefs, and time, these relations represent a theory-of-mind interpretation of a text. We show that these discourse relations are expressive, formal, robust, and through the use of a software system, amenable to corpus collection projects through the use of trained annotators. We have procured and released a collection of over 100 encodings, covering a set of fables as well as longer texts including literary fiction and epic poetry. We are able to inferentially find similarities and analogies between encoded stories based on the proposed relations, and an evaluation of this technique shows that human raters prefer such a measure of similarity to a more traditional one based on the semantic distances between story propositions.

Table of Contents

List of Figures

List of Tables

Acknowledgments

It takes a village to write a thesis, especially one that strives to cross disciplines. Throughout my years at Columbia, I have been fortunate to come across so many curious, brilliant and generous colleagues. My gratitude goes out to all of them. I would like to send thanks in particular:

First and foremost, to Kathy McKeown, my advisor, who took me on and allowed me to take her research in new and unexpected directions with the utmost trust and confidence. This thesis would not exist if not for her expert guidance and patient support throughout its many turns. To Julia Hirschberg, my secondary advisor, whose enthusiasm for my thesis topic gave me the confidence to commit to it fully.

To the rest of my committee: Owen Rambow, who helped identify social network analysis in literature as a fruitful new approach; to Michael Collins, who provided invaluable advice on the machine learning aspects of my approach; and to Nicholas Dames, my co-author from the Department of English and Comparative Literature, who provided a trove of insight and guidance that brought the experiment to life. Also to Dragomir Radev, Franco Moretti, David Mimno and Graham Sack for their assistance with this experiment.

To Rebecca Passonneau and Inderjeet Mani for their earlier service on my committee; both provided crucial guidance at an important juncture that helped to focus and advance the Scheherazade experiment.

To Tanya Korelsky and the National Science Foundation for their generous support;[1] to Jack McGourty and the NSF GK-12 program, which previously supported me; to Shakira Lleras and Justine Thomas, the middle-school science teachers with whom I worked at

[1] This material is based on research supported in part by the U.S. National Science Foundation (NSF) under IIS-0935360. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

I.S. 143 for my GK-12 service; to Adam Cannon, Chris Murphy and Herb Chase for their guidance during my years as a preceptor teaching Data Structures and Algorithms to undergraduates. To Mark Riedl, for letting me explore other directions in computational creativity at the Institute for Creative Technologies during my internship there.

To Norma Graham for her excellent mentorship since my undergraduate days; to Livia Polanyi for her more recent, but equally excellent mentorship. To Judith Klavans for starting me on this path by hiring me as a research programmer, and for her encouragement in the intervening years. To David Plante and his creative writing students in the School of the Arts for welcoming me into their seminars to exchange ideas about narrative theory. To Marie-Laure Ryan for piquing my interest in the computational treatment of narrative.

To Fadi Biadsy, my office-mate, for enlivening my working environment and being ever free to engage in long and helpful technical discussions; to the many other fellow Ph.D. students who helped me through the process with both advice and distraction, including (but not limited to) Apoorv Agarwal, Mohamed Altantawy, Regina Barzilay, Daniel Bauer, Hila Becker, Ana Benitez, Arvid Bessen, Sasha Blair-Goldensohn, Bob Coyne, Wisam Dakka, Pablo Duboue, Noémie Elhadad, Frank Enos, Dave Evans, Elena Filatova, Jenny Finkel, Michel Galley, Joshua Gordon, Agustín Gravano, Aaron Harnly, Alpa Jain, Martin Jansche, Min-Yen Kan, Kevin Lerman, Jackson Liscombe, Wei-Yun Ma, Sameer Maskey, Smaranda Muresan, Ani Nenkova, Kristen Parton, Andrew Rosenberg, Sara Rosenthal, Barry Schiffman, Andrew Schlaikjer and Kapil Thadani. To the helpful administrators of the Department of Computer Science as well, especially Laura Furst and Daisy Nguyen.

To the formative evaluators, project students and others who assisted in developing and evaluating the Scheherazade tool: Deborah Aschkenes, David Berke, Cat Bohannon, Kyla Cheung, Allie Curry, Marshall Fox, Edmond Horsey, Eleanor Johnson, Ramya Joseph, Albert Ngo, Jonathan Rowe, Lucy Sheehan, Christine Smallwood, Erica Weaver, Jerry Weltman and Danny Wright.

And lastly, to my family, who never thought it crazy to pursue a Ph.D. about computational narratology—especially to my parents Norton and Sandy, my siblings Jeremy and Franny, and my uncle Alan and aunt Joan. And, of course, to Carly, who has helped me see the light at the end of the thesis. With her I can now start the next exciting chapter.

For My Parents

Chapter 1

Introduction

Narrative occurs with every other discourse type, including dialogue and multi-party interaction. From the time we are old enough to understand fairy tales, we learn of values, customs and mores from the stories we hear [Nelson, 1987]. In our day-to-day lives, narrative is the coin with which we exchange information and experience, from the news articles we read over breakfast [Fulton *et al.*, 2005], to the gossip we exchange around the water cooler, to the television shows and books into which we "transport" ourselves in the evening [Green *et al.*, 2004]. As Barthes [1975] puts it, narrative is "present at all times, in all places, in all societies... like life itself, it is there, international, transhistorical, transcultural."

Why is narrative such a pervasive mode of discourse? Philosophers and psychologists have advanced the idea that the story is a key structure for human thought. Dennett [1991] understands the notion of the self as a "center of narrative gravity," and our narrative selfhood as the product of an endless spinning of a "web" of stories by our consciousness. The progress of a child's development can be measured by the emergence of an autobiographical sense of self [Nelson, 1989; Nelson, 2003]. Bruner [1986] distinguishes between narrative and expository text, with the former triggering an "active search for meaning" on the part of a reader who draws drama out of the particulars of a related experience. He goes as far as to say that narrative organizes the structure of human experience [Bruner, 1991]. Empirical tests have long suggested that narrative is a key structural component of memory, since causal and temporal connections in a discourse—the basic structural scaffold of a story—lead to greater recall [Bartlett, 1932]. Narrative was likely the predominant form of oral discourse before writing was invented [Rubin, 1995]. In particular, narrative has long been a vehicle for reflecting on ethical questions, describing characters tangled in conflict and facing dilemmas where they must choose between competing values. Literary criticism often focuses on the ethical "code" underlying a text [Booth, 1989].

As the volume of human interaction that takes place online increases, so too does the volume of narrative discourse in machine-readable form. Project Gutenberg[1] and Google Books[2] have scanned and published millions of volumes across all subject areas, including large sections of literary fiction and non-fiction. Millions of people worldwide trade personal stories every day through social networking and blogging sites such as Twitter[3] and Blogger.[4] Google News,[5] NewsInEssence[6] and the Columbia Newsblaster project[7] collectively crawl and filter the thousands of news articles that compete for our attention and our compassion on a daily basis. Wikipedia,[8] the collaborative free encyclopedia, features thousands of articles with narrative discourse in many languages, including biographies, film plot summaries, and the historical overviews associated with places, objects, times and ideas.

Natural language processing is playing a large and crucial role in allowing us to understand, search, summarize and filter all these stories. Most of these tools currently in use, though, operate at the keyword or topic level, with no particular consideration given to what separates a narrative discourse from an expository discourse or even a list of disconnected facts. This sense of "storiness" has yet to be identified and exploited on a large scale. For instance, searching a news aggregator for the phrase "struggle against oppression" will return articles that contain instances of those three words, missing many articles that involve

[1] http://www.gutenberg.org

[2] http://books.google.com

[3] http://www.twitter.com

[4] http://www.blogger.com

[5] http://news.google.com

[6] http://www.newsinessence.com

[7] http://newsblaster.cs.columbia.edu

[8] http://www.wikipedia.org

such struggles without referring to them by this moniker (or any set of consistently applied keywords).

We are particularly interested in the similarities and analogies that occur between stories. On a cognitive level, we understand stories and events in the context of previous stories we have heard and previous events we have experienced. We find connections and relationships between the new and the old. This can be seen plainly in law and ethics, where case studies about the past are used as templates for understanding more current matters. It can also be seen in the many metaphors and allegories we use on a daily basis to connect new stories to old ones—some government's austerity measures threaten to "kill the goose that laid the golden eggs," an overly zealous individual may "cry wolf" too many times, a posturing politician is an "emperor with no clothes," a certain event opened "a Pandora's box." Aphorisms and idioms can resemble small narratives even if they did not originate in a myth or fable. "Out of the frying pan and into the fire," for instance, has all the hallmarks of a dramatic tale: danger, a plan to escape the danger, and a disastrous outcome to the plan resulting in more danger. We use these small analogies to better communicate the meanings of stories to one another. An aggregator of online natural language with narrative competence could find not only the many stories that involve struggles against oppression in some form or another, but also differing points of view on the same events that use contrasting narrative scaffolds: "freedom fighter" as opposed to "terrorist," "administration" as opposed to "government" and "regime," "social justice" as opposed to "socialism," and so on. Narrative analogies connect human expression across media at a level beyond lexical overlap.

The analysis of discourse concerns the relations between clauses and sentences that make a document more than the sum of its parts. To date, though, most work in the automatic analysis of discourse has focused on expository text rather than narrative text. The most commonly used models of discourse, Rhetorical Structure Theory (RST) [Mann and Thompson, 1988] and the Penn Discourse Treebank [Prasad *et al.*, 2008], deal in terms of subordinating conjunctions (*when*, *because*), coordinating conjunctions (*and*, *but*) and other relations that give discourse its coherence. These certainly appear in narrative texts, as do coreference and anaphora, which relate clauses and sentences together by the entities to which they repeatedly refer (such as people, places, and things). However, narratives also feature relations that do not appear in these models: between characters who are socially linked in a meaningful way, between a goal and its outcome, between an action and the strategic plan that the actor is attempting to fulfill, and more. We see these intra-textual links as being among the building blocks of "storiness" in a relatively unexplored corner of work of discourse.

We are also interested in the emerging field of **digital humanities**, which looks for connections and patterns within large corpora of literary texts (among other objectives). The value here is more intrinsic than extrinsic: What can we learn about a literary genre through statistical analysis? Traditional theory and criticism is based on the close study of a work or a small set of works, such as comparing an aspect of Dickens with an aspect of Austen. The advent of large scale corpora has made possible a mode of analysis which Moretti [2000a] calls a *distant read*, where thousands of books are automatically scanned and analyzed in an attempt to understand the long-term and large-scale trends can be seen through the aggregate study of thousands of novels published over the centuries since the printing press was introduced. This objective can be pursued at the word level—a recent tool called the Google Ngram Viewer[9] allows us to track the rising and falling prevalence of a word or phrase from the beginning of publishing to the present—but we believe that one can also find fruitful avenues of analysis by considering each text as a structured discourse, rather than as a collection of words.

This thesis is an exploration of narrative discourse relations and the ways in which they can reveal insights about a genre or a single text through manual annotation, automatic tagging and computational analysis. It is a search for a formal representation that can unlock information about structure and content, much in the way a formal model of syntax has allowed us to build high-accuracy syntactic parsers. We hypothesize that the better an algorithm can identify what sets a narrative apart from a collection of facts or a sequence of expository sentences, the better it can leverage these features to find patterns, connections, and analogies between stories. Such a tool will help us organize our thoughts and our writings, communicate with one another, and understand our culture at large.

[9] http://ngrams.googlelabs.com/

Outline

The central challenge to pursuing our goal is that the "meaning of a story" involves many intersecting factors. On a basic level, a narrative consists of at least two events that are presented in a temporal and causal sequence [Labov, 1972]; therefore, temporal and causal links are certainly the founding members of a set of narrative discourse relations. A story, by definition, evokes a story-world: the narrative reality being told that involves at least two time states, events that occur during those time states, and a functional relationship between those events. But, as we will see in Chapter 3, these relations alone have presented difficult computational challenges over the last few decades; no robust natural-language parser of time and causality has yet been built.

To make matters more complicated, narrative meaning has been persuasively argued to involve other factors as well: The agency of characters who participate in the story-world, who each act as conscious entities with independent minds and wills (having beliefs and goals); the interactions between characters (that is, the social network implied by a story); the world knowledge agreed upon by the teller and the receiver (such as the laws of physics and physiology that present danger to a character in freefall); and the moral point of the story (why is it being told at all?). Also important are the pragmatics of a narrative exchange: What is the teller's purpose? Is he trying to convince the receiver of a moral principle, to convey an emotional experience, or to pursue yet another goal? How does the receiver construct an image in her head of the story being told? Moreover, how does the receiver's search for meaning lead her to sympathize with a particular character, expect a certain outcome, or have an overall feeling of being "transported" into the story-world?

The consideration of all these factors in a language understanding system is too large a leap to be completed in a single thesis. There is instead a contrast between what can be practically accomplished and what the field should aspire to reach in the years to come. This thesis splits the difference by pursuing two approaches to the problem, one practical and one aspirational:

1. **Social network extraction in literature.** We build a system to extract features from literary novels, and apply machine learning tools to analyze those novels at the

scale of 10 million words (60 novels). Specifically, we use the structure of **quoted speech** found in the Victorian novel to determine the interactions between characters, and use those interactions to construct social networks. We then find the intrinsic value of these networks with respect to traditional literary theory. In particular, we consider theories that have been suggested by scholars of this genre about the relationships between the size of a community, the social network's interconnectedness, and the story's setting (between urban and rural). By comparing the extracted conversational networks to one another, we determine whether there is empirical evidence supporting these theories.

2. **Story Intention Graphs.** In the remainder of the thesis, we propose a new set of discourse relations that encodes the agent-oriented (theory of mind [Palmer, 2007]) meaning of a story. These relations cover not only time and causality, but also the presence of agents, objects and themes (through coreference relations), events, statives, goals, beliefs, plans, attempts to pursue goals, outcomes of attempts, and the affectual impacts of story-world events on each agent. This schemata, which we call the Story Intention Graph or SIG, is integrated with previously proposed sentence-level annotation schemes that identify the propositional structure of each sentence (predicates, thematic roles and arguments). We show that the SIG satisfies important criteria for a model of narrative discourse relations, including expressiveness, formality, and robustness to varying levels of semantic precision in a story's annotation.

   We present the automatic annotation of a narrative text into a SIG encoding as a worthwhile goal that would bring us closer to a meaningful computational understanding of narrative discourse, and show progress toward achieving that goal. To this end, we describe a software platform for manually annotating stories and extracting features from SIG encodings. We have used this tool, Scheherazade, to collect a corpus of 110 story encodings which we call **DramaBank** (analogous to the PropBank corpus [Kingsbury and Palmer, 2002], which includes predicate-argument annotation). We then describe a set of algorithms that find similarities and analogies between DramaBank encodings, and demonstrate the usefulness of the SIG schemata for finding thematic relationships between texts. An evaluation shows that SIG rela-

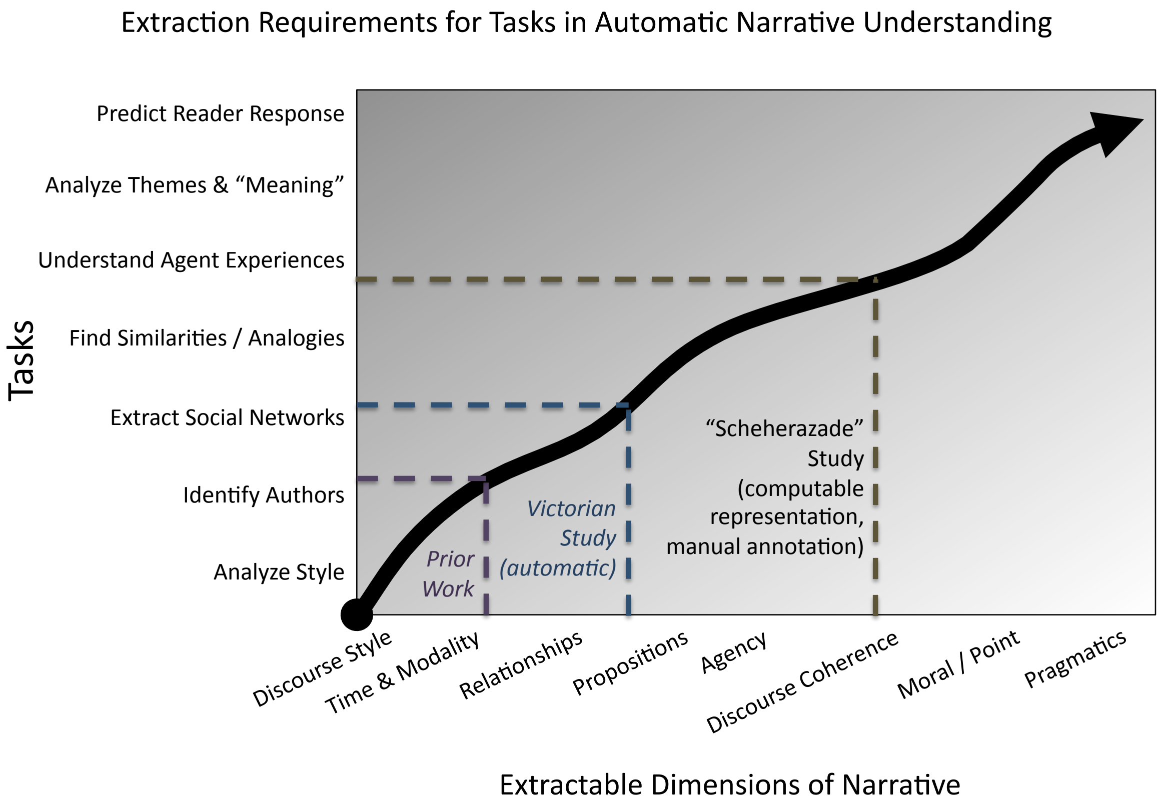


Figure 1.1: Extraction requirements for tasks in automatic narrative analysis.

tions enable us to describe similarities and analogies between stories more effectively than a method based on lexical and syntactic (propositional) similarities alone.

Figure 1.1 illustrates how these two approaches fit together in a larger sense. It plots the types of features we may aim to extract from a narrative along the horizontal axis, in generally increasing order of difficulty, against a set of tasks in narrative analysis enabled by those features along the vertical axis. Features that operate at the word level have allowed us to carry out analyses of writing style, which can be applied for authorship identification and similar tasks [Mostellar and Wallace, 1984]. The first part of this thesis investigates techniques for automatically detecting interpersonal relationships in a text, which allow us to examine a discourse or a genre with the methods of social network analysis. The latter part of the thesis looks forward to the automatic extraction of propositions, elements of agency and narrative discourse relations—which, as we show in Chapter 5, will allow us to find substantive similarities and analogies between stories.

Overview of Contributions

The contributions of this thesis are:

1. An approach for extracting conversational networks from literary fiction and an evaluation of previously proposed theories about the social networks found in the genre of 19th century British literature. This involves the application of machine learning tools to determine the most likely speaker for each direct quotation in a corpus of Victorian novels.

2. A novel set of discourse relations relating to the narrative structure of the discourse beyond social connectedness. These relations organize a text into a structure called a Story Intention Graph, or SIG, which consists of three interconnected sections:

   - The surface text (i.e., the original discourse),
   - A series of structured timelines describing the events and statives that take place in the story-world, and
   - A receiver's interpretation of the agentive meaning of the text, including the inner beliefs, goals and plans of the characters found in the story-world.

3. The construction, formative evaluation and release of a software platform, SCHEHERAZADE, which makes the process of annotating a story into a SIG encoding amenable to trained annotators. The platform includes a simple textual generation component that renders a discourse from an encoding, including a model of the relationship between English tense and aspect and a semantic representation of time.

4. A corpus of over 100 SIG encodings, collectively known as DramaBank. The 33 source texts include a set of Aesop's fables as well as a news article, the epic poem *Beowulf* and literary short fiction.

5. A set of algorithms for finding similarities and analogies between stories, by means of the SIG encodings that have been constructed for each story. An evaluation comparing three such methods, including one that operates by finding the semantic distances

between propositions, and two that rely on identifying analogous patterns of SIG relations.

The thesis is structured as follows: Chapter 2 describes the Victorian corpus experiment, with additional social networks illustrated in Appendix A. We introduce the SIG in Chapter 3, and continue this discussion in Appendix B with an overview of the types of narrative scenarios that can be represented by the SIG relations. We describe the implementation of the SCHEHERAZADE annotation platform, including a graphical interface and a textual discourse generation component, in Chapter 4. Chapter 5 describes the collection and analysis of the DramaBank corpus, as well as the three approaches to finding similarities and analogies. Finally, we conclude in Chapter 6.

Chapter 2

Literary Social Networks

As there are many perspectives from which one may examine narrative, there are many types of models with which one can encode a narrative discourse. Some models focus on a certain aspect of the discourse and use it as a thumbnail to describe the larger whole. A map of the city in which a story takes place, for example, can summarize a discourse by the spatial movements of its characters [Eco, 1995]. Similarly, a social network describes the relationships between characters (agents) that appear in a text. While a social graph does not tell the whole story, it falls under Moretti's [2000a] concept of the distant read—we trade off a detailed, contextual understanding of the particulars of each text, but gain in return the ability to analyze large groups and even entire genres. The idea of the distant read dovetails with the methodology of natural language processing; in essence, the task becomes one of information extraction at the level of discourse (as opposed to sentence or paragraph). By using a trained classifier to scan large quantities of text for key words and phrases, one can offer a perspective on literature that examines a far greater quantity of work than one can consider in a single survey of close reading. The insights of the distant read complement those of the close read, rather than displace them.

The notion of extracting social networks from literary texts offers a wealth of possible collaborations between computer scientists and literary experts. Studies about the nineteenth-century British novel, for instance, are often concerned with the nature of the community that surrounds the protagonist. Some theorists have suggested a relationship between the size of a community and the amount of dialogue that occurs, positing that "face to face time" diminishes as the number of characters in the novel grows. Others suggest that as the social setting becomes more urbanized, the quality of dialogue also changes, with more interactions occurring in rural communities than urban communities. Such claims have typically been made, however, on the basis of a few novels that are studied in depth. In this chapter, we aim to determine whether an automated study of a larger sample of nineteenth century novels supports these claims.

The following sections investigate the extraction of social networks from literature. We present a method to automatically construct a network based on dialogue interactions between characters in a novel. Our approach includes components for finding instances of quoted speech, attributing each quote to a character, and identifying when certain characters are in conversation. This allows us to construct a network where characters are vertices and edges signify an amount of bilateral conversation between those characters, with edge weights corresponding to the frequency and length of their exchanges. In order to evaluate the literary claims in question, we compute various characteristics of the dialogue-based social network and stratify these results by categories such as the novel's setting. For example, the density of the network provides evidence about the cohesion of a large or small community, and cliques may indicate a social fragmentation. Our results do not indicate that the majority of novels in this time period fit the suggestions provided by literary scholars, and we suggest an alternative explanation for our observations of differences across novels.

In contrast to previous approaches to social network construction, ours relies on a novel combination of pattern-based detection, statistical methods, and the adaptation of standard natural language processing tools for the literary genre. We carried out this work on a corpus of 60 nineteenth-century novels and serials, including 31 authors such as Dickens, Austen and Conan Doyle. In the following sections, we survey related work on social networks as well as computational studies of literature, and describe the relevant literary hypotheses in more detail. We then describe the methods we use to extract dialogue and construct networks, along with our approach to analyzing their characteristics. After we present our results, we discuss their significance from a literary perspective.

2.1 Related Work

Computer-assisted literary analysis has typically occurred at the word level. This level of granularity lends itself to studies of authorial style based on patterns of word use [Burrows, 2004], and researchers have successfully "outed" the writers of anonymous texts by comparing their style to that of a corpus of known authors [Mostellar and Wallace, 1984]. Determining instances of "text reuse," a type of paraphrasing, is also a form of analysis at the lexical level; it has recently been used to validate theories about the lineage of ancient texts [Lee, 2007]. The Google Ngram Viewer is based on a scanning of millions of volumes [Michel *et al.*, 2011].

Automatic analysis of literature using more semantically-oriented techniques has been rare, most likely because of the difficulty in automatically determining meaningful interpretations. Some exceptions include recent work on learning common event sequences in news stories [Chambers and Jurafsky, 2008a], an approach based on statistical methods, and the development of an event calculus for characterizing stories written by children [Halpin *et al.*, 2004], a knowledge-based strategy. On the other hand, literary theorists, linguists and others have long developed symbolic but non-computational models for novels. For example, Moretti [2005] has graphically mapped out texts according to geography, social connections and other variables.

There has been progress toward the automatic extraction of social networks representing connections between characters in discourse [Agarwal and Rambow, 2010], although typically not for the genre of literature. For example, the ACE program has involved entity and relation extraction in unstructured text [Doddington *et al.*, 2004]. Other recent work in social network construction has explored the use of more structured data such as email headers [McCallum *et al.*, 2007; Bird *et al.*, 2006], news articles [Tanev, 2007; Pouliquen *et al.*, 2008] and U.S. Senate bill co-sponsorship [Cho and Fowler, 2010]. In an analysis of discussion forums, Gruzd and Haythornthwaite [2008] explored the use of message text as well as posting data to infer who is talking to whom. In the present study, we also explore how to build a network based on conversational interaction, but we analyze the reported dialogue found in novels to determine the links. The kinds of language that are used to signal such information differ between the two forms. In discussion forums, people tend to use addresses such as “Hey, Sally,” while in novels, a system must determine both the speaker and the intended recipient of a dialogue act. This is a significantly different problem.

2.2 Hypotheses

Within literary studies, there are many theories about the relation between novelistic form (the workings of plot, characters, and dialogue, to take the most basic categories) and changes to real-world social milieux.[1] Many of these theories center on nineteenth-century European fiction; innovations in novelistic form during this period, as well as the rapid social changes brought about by revolution, industrialization, and transport development, have traditionally been linked. These theories, however, have used only a select few representative novels as proof; as Moretti put it, “if we set today’s canon of nineteenth-century British novels at two hundred titles (which is a very high figure), they would still be only about 0.5 percent of all published novels” [Moretti, 2000b]. By using statistical methods, it is possible to broaden an analysis to include hundreds or thousands of texts rather than one or several. We believe these methods are essential to testing the validity of some core theories about social interaction and its representation in literary genres like the novel.

Major versions of the theories about the social worlds of nineteenth-century fiction tend to center on characters, in two specific ways: how many characters novels tend to have, and how those characters interact with one another. These two properties of novels are usually explained with reference to a novel’s setting. From the influential work of the Russian critic Mikhail Bakhtin to the present, a consensus has emerged that as novels are increasingly set in urban areas, the number of characters and the quality of their interaction change to suit the setting. Bakhtin’s term for this causal relationship was *chronotope*: the “intrinsic interconnectedness of temporal and spatial relationships that are artistically expressed in literature,” in which “space becomes charged and responsive to movements of time, plot, and history” [Bakhtin, 1981, 84]. In Bakhtin’s analysis, different spaces have different

[1]This chapter describes joint work previously published with Kathleen McKeown and Nicholas Dames [Elson and McKeown, 2010; Elson *et al.*, 2010]. The literary insights in this chapter (Sections 2.2 and 2.7.2) are those of Prof. Dames.

social and emotional potentialities, which in turn affect the most basic aspects of a novel's aesthetic technique.

After Bakhtin's invention of the chronotope, much literary criticism and theory devoted itself to describing qualities of specific chronotopes, particularly those of the village or rural environment and the city or urban environment. Following a suggestion of Bakhtin's that the population of village or rural fictions is modeled on the world of the family, made up of an intimately related set of characters, many critics analyzed the formal expression of this world as constituted by a small set of characters who express themselves conversationally. Raymond Williams used the term "knowable communities" to describe this world, in which face-to-face relations among a restricted set of characters are the primary mode of social interaction [Williams, 1975, 166].

By contrast, the urban world, in this traditional account, is both larger and more complex. To describe the social-psychological impact of the city, Franco Moretti argues, protagonists of urban novels "change overnight from 'sons' into 'young men': their affective ties are no longer vertical ones (between successive generations), but horizontal, within the same generation. They are drawn towards those unknown yet congenial faces seen in gardens, or at the theater; future friends, or rivals, or both" [Moretti, 1999, 65]. The result is two-fold, with more characters and more interactions, but less actual conversation. As literary critic Terry Eagleton argues, the city is where "most of our encounters consist of seeing rather than speaking, glimpsing each other as objects rather than conversing as fellow subjects" [Eagleton, 2005, 145]. Moretti argues in similar terms. For him, the difference in number of characters is "not just a matter of quantity... it's a qualitative, morphological one" [Moretti, 1999, 68]. As the number of characters increases, Moretti argues (following Bakhtin in his logic), social interactions of different kinds and durations multiply, displacing the family-centered and conversational logic of village or rural fictions. "The narrative system becomes complicated, unstable: the city turns into a gigantic roulette table, where helpers and antagonists mix in unpredictable combinations" [Moretti, 1999, 68]. This argument about how novelistic setting produces different forms of social interaction is precisely what our method seeks to evaluate.

We assembled a representative corpus of Victorian novels in order to test two hypotheses which are derived from these theories:

1. That there is an inverse correlation between the amount of dialogue in a novel and the number of characters in that novel. One basic, shared assumption of these theorists is that as the network of characters expands—as a quantitative change becomes qualitative, in Moretti's words—the amount and importance of dialogue *decreases*. With a method for extracting conversations from texts, it is possible to test this hypothesis against our corpus.

2. That a significant difference in the nineteenth-century novel's representation of social interaction is geographical: Novels set in urban environments depict complex but loose social networks, in which numerous characters share little conversational interaction, while novels set in rural environments inhabit more tightly bound social networks, with fewer characters sharing much more conversational interaction. This hypothesis is based on the contrast between Williams's rural "knowable communities" and the sprawling, populous, less conversational urban fictions or Moretti's and Eagleton's analyses. If true, this would suggest that the inverse relationship of the first hypothesis (more characters means less conversation) is correlated to, and perhaps even caused by, the geography of a novel's setting. The claims about novelistic geography and social interaction have usually been based on comparisons of a few select novelists (especially Austen and Dickens). Do they remain valid when tested against a larger corpus?

2.3 Overview of Corpora and Methodology

In order to test these hypotheses, we developed a novel approach to extracting social networks from literary texts themselves, building on existing analysis tools. First, we defined "social network" as "conversational network" for purposes of evaluating these literary theories. In a conversational network, vertices represent characters (assumed to be named entities) and edges indicate at least one instance of dialogue interaction between two characters over the course of the novel. The weight of each edge is proportional to the amount of interaction. We define a conversation as a continuous span of narrative time featuring a set of characters in which the following conditions are met: The characters are in the same place at the same time; they take turns speaking; they are mutually aware of one another; and each character's speech is intended for the other to hear.

There are, of course, significant differences between a social network in the conventional sense, and a conversational network as defined in terms of face-to-face dialogue. Characters who are strongly connected by social bonds such as family, race, class or cohabitation may nonetheless never engage in face-to-face conversation, so this metric carries with it a certain loss of "recall" compared to the more broadly defined ACE task. However, we believe that a conversational network is sufficient to capture Eagleton's distinction between "glimpsing each other as objects" and "conversing as fellow subjects," as well as Moretti's interpretation of the city as a "roulette table" where characters "mix" in unpredictable ways—especially given that, by and large, the novels were written prior to the invention of the telephone.

The next three sections describe our pipeline:

1. **Character Identification.** Identify the characters present in the text.
2. **Quoted Speech Attribution.** Given a set of characters mentioned in the text, and a sequence of spans of quoted speech, determine which character (if any) is speaking each quote.
3. **Conversational Networks Construction.** Identify the salient conversations that exist in the text, and use this information to construct a conversational network that describes the overall social structure found in the novel.

We ran our experiments on a corpus of 60 novels which we call the **LSN corpus** (for Literary Social Networks). With the guidance of Nicholas Dames, an expert in the Victorian novel in the Department of English & Comparative Literature, we selected these texts to represent the genre of Victorian fiction as a whole and to include contrasts along several categories: authorial (novels from the major canonical authors of the period), historical (novels from each decade), generic (from the major sub-genres of nineteenth-century fiction, such as historical and social), sociological (set in rural, urban, and mixed locales), and with respect to perspective (narrated in first-person and third-person form).

Author/Title/Year	Persp.	Setting	Author/Title/Year	Persp.	Setting
Ainsworth, *Jack Sheppard* (1839)	3rd	urban	Gaskell, *North and South* (1854)	3rd	urban
Austen, *Emma* (1815)	3rd	rural	Gissing, *In the Year of Jubilee* (1894)	3rd	urban
Austen, *Mansfield Park* (1814)	3rd	rural	Gissing, *New Grub Street* (1891)	3rd	urban
Austen, *Persuasion* (1817)	3rd	rural	Hardy, *Jude the Obscure* (1894)	3rd	mixed
Austen, *Pride and Prejudice* (1813)	3rd	rural	Hardy, *The Return of the Native* (1878)	3rd	rural
Braddon, *Lady Audley's Secret* (1862)	3rd	mixed	Hardy, *Tess of the d'Ubervilles* (1891)	3rd	rural
Braddon, *Aurora Floyd* (1863)	3rd	rural	Hughes, *Tom Brown's School Days* (1857)	3rd	rural
Brontë, Anne, *The Tenant of Wildfell Hall* (1848)	1st	rural	James, *The Portrait of a Lady* (1881)	3rd	urban
Brontë, Charlotte, *Jane Eyre* (1847)	1st	rural	James, *The Ambassadors* (1903)	3rd	urban
Brontë, Charlotte, *Villette* (1853)	1st	mixed	James, *The Wings of the Dove* (1902)	3rd	urban
Brontë, Emily, *Wuthering Heights* (1847)	1st	rural	Kingsley, *Alton Locke* (1860)	1st	mixed
Bulwer-Lytton, *Paul Clifford* (1830)	3rd	urban	Martineau, *Deerbrook* (1839)	3rd	rural
Collins, *The Moonstone* (1868)	1st	urban	Meredith, *The Egoist* (1879)	3rd	rural
Collins, *The Woman in White* (1859)	1st	urban	Meredith, *The Ordeal of Richard Feverel* (1859)	3rd	rural
Conan Doyle, *The Sign of the Four* (1890)	1st	urban	Mitford, *Our Village* (1824)	1st	rural
Conan Doyle, *A Study in Scarlet* (1887)	1st	urban	Reade, *Hard Cash* (1863)	3rd	urban
Dickens, *Bleak House* (1852)	mixed	urban	Scott, *The Bride of Lammermoor* (1819)	3rd	rural
Dickens, *David Copperfield* (1849)	1st	mixed	Scott, *The Heart of Mid-Lothian* (1818)	3rd	rural
Dickens, *Little Dorrit* (1855)	3rd	urban	Scott, *Waverley* (1814)	3rd	rural
Dickens, *Oliver Twist* (1837)	3rd	urban	Stevenson, *The Strange Case of Dr. Jekyll and Mr. Hyde* (1886)	1st	urban
Dickens, *The Pickwick Papers* (1836)	3rd	mixed	Stoker, *Dracula* (1897)	1st	urban
Disraeli, *Sybil, or the Two Nations* (1845)	3rd	mixed	Thackeray, *History of Henry Esmond* (1852)	1st	urban
Edgeworth, *Belinda* (1801)	3rd	rural	Thackeray, *History of Pendennis* (1848)	1st	urban
Edgeworth, *Castle Rackrent* (1800)	3rd	rural	Thackeray, *Vanity Fair* (1847)	3rd	urban
Eliot, *Adam Bede* (1859)	3rd	rural	Trollope, *Barchester Towers* (1857)	3rd	rural
Eliot, *Daniel Deronda* (1876)	3rd	urban	Trollope, *Doctor Thorne* (1858)	3rd	rural
Eliot, *Middlemarch* (1871)	3rd	rural	Trollope, *Phineas Finn* (1867)	3rd	urban
Eliot, *The Mill on the Floss* (1860)	3rd	rural	Trollope, *The Way We Live Now* (1874)	3rd	urban
Galt, *Annals of the Parish* (1821)	1st	rural	Wilde, *The Picture of Dorian Gray* (1890)	3rd	urban
Gaskell, *Mary Barton* (1848)	3rd	urban	Wood, *East Lynne* (1860)	3rd	mixed

Table 2.1: Properties of the nineteenth-century British novels and serials included in the LSN corpus.

The novels, as well as important metadata we assigned to them (formal perspective and setting), are shown in Table 2.1. For purposes of these labels, we define *urban* to mean set in a metropolitan zone, characterized by multiple forms of labor (not just agricultural). We conversely define *rural* to describe texts that are set in a country or village zone, where agriculture is the primary activity, and where land-owning, non-productive, rent-collecting gentry are socially predominant. We also explored the other properties that we identified, such as sub-genre, but focus on the results regarding setting and perspective. We obtained electronic encodings of the texts from Project Gutenberg.[2] All told, these texts total more

[2] http://www.gutenberg.org

Author	Title	Year	# Quotes	Quotes attributed
Jane Austen	*Emma**	1815	549	546
Charles Dickens	*A Christmas Carol*	1843	495	491
Gustave Flaubert	*Madame Bovary**	1856	514	488
Mark Twain	*The Adventures of Tom Sawyer**	1876	539	478
Sir Arthur Conan Doyle	"The Red-Headed League" "A Case of Identity" "The Boscombe Valley Mystery" "A Scandal in Bohemia"	1890 1888 1888 1888	524	519
Anton Chekhov	"The Steppe" "The Lady with the Dog" "The Black Monk"	1888 1899 1894	555	542

Table 2.2: Makeup of the QSA corpus. * indicates that excerpts were used.

than 10 million words.

For our work in quoted speech attribution, we curated a separate development corpus called the **QSA corpus**. This corpus includes some of the same texts as the LSN corpus, but samples texts from other genres as well. We compiled passages of 11 works by Chekhov, Flaubert, Twain, Austen, Dickens and Conan Doyle that appeared between 1815 and 1899. Each author was influential in popularizing the form of the novel (or, in the cases of Chekhov and Conan Doyle, the short story) as a medium distinct from the more well-established play or poem. However, these works still hearken back to the older form, in that like a play, they consist of extended scenes of dialogue between two or more individuals in a scene. These texts have a large proportion of quoted speech (dialogue and internal monologue).

Four authors represented in the QSA wrote in English, one in Russian (translated by Constance Garnett) and one in French (translated by Eleanor Marx Aveling); two authors contribute short stories and the rest novels (while Dickens often wrote in serial form, *A Christmas Carol* was published as a single novella). Excerpts were taken from *Emma*, *Madame Bovary* and *The Adventures of Tom Sawyer*. In all, the QSA corpus consists of about 111,000 words including 3,176 instances of quoted speech (which we define as a span of text within a paragraph that falls between opening and closing quotation marks).

As its name implies, the purpose of the QSA corpus is to allow us to train our model for attributing quoted speech acts to their respective speakers. This is a "gold standard" corpus that allows us to interpret this problem as a supervised learning task. To obtain ground-truth annotations of which characters were speaking or thinking which quotes, we conducted an online survey via Amazon's Mechanical Turk distributed work program. For each quote, we asked 3 annotators to independently choose a speaker from the list of contextual candidates—or, choose "spoken by an unlisted character" if the answer was not available, or "not spoken by any character" for non-dialogue cases such as sneer quotes. (We include attributed thoughts and monologues in our definition of "dialogue.") We describe below the method by which we extract candidate speakers, including named entities and nominals, from the text. Up to 15 candidate speakers were presented for each quote from up to 10 paragraphs preceding the quote (including the quote's paragraph itself). When two definite noun phrases referred to the same person (e.g., "Harriet" and "Emma's friend"), annotators were instructed to choose the reference "most strongly associated" with the quote in question. Annotators typically chose the closest mention, except in cases where a more complete mention was nearby.

Of the 3,578 quotes in the survey results, 2,334 (about 65%) had unanimous agreement as to the identity of the speaker, and 1,081 (another 30%) had a 2-vote majority which was assumed to be the correct answer. The remaining 4.5% had a total 3-way tie, often in cases where multiple coreferent names were offered for the same speaker. We excluded these cases from our corpus, as coreference is not our main focus. To normalize for poor annotator performance, each annotator was graded according to the rate at which she agreed with the majority. If this rate fell below 50%, we omitted all the annotator's ratings; this affected only 2.6% of the votes.

We also excluded from evaluation the 239 quotes (7%) where a majority agreed that the correct speaker was not among the options listed. This includes the 3% of quotes for which the correct speaker was never considered as a candidate because our pipeline failed to identify it (a genuine recall issue, most often associated with unnamed entities such as *the porter* that did not appear to be potential speakers). In the remaining 4%, the passage we presented to annotators did not extend far back enough to determine the speaker (an artifact of our annotation methodology). Annotators also agreed that 112 of the quotes (3.5%) were non-dialogue text. We set out to detect such cases alongside quotes with speakers.

We put aside one-third of the QSA corpus for use in developing our method, and left the remainder for training and testing. We have publicly released these data to encourage further work.[3] We used the development segment of the QSA corpus to build and test our character identifier as well as a pre-processing script that normalizes formatting, detects headings and chapter breaks, removes metadata, and identifies likely instances of quoted speech—those spans of text that fall between quotation marks, which we assume to be a superset of the quoted speech present in the text. We did not hand-annotate the LSN corpus or the testing segment of the QSA corpus, with two exceptions: to correct OCR errors left over from the scanning process, and, in cases where single quotes are used to delimit quoted speech instead of double quotes, to disambiguate such delimiters from apostrophes used in colloquial abbreviations (e.g., *drinkin'*).

2.4 Character Identification

The first challenge is to "chunk" (identify and delimit) mentions of characters from the text. It is not enough to compile a list of names; we must annotate the text to identify each mention of each character, so we can have a better idea about when each character is speaking. This is a part of the Named Entity Recognition (NER) task, as previously articulated in by ACE program [Doddington *et al.*, 2004] and elsewhere.

Character identification in novels is made complicated by the fact that there are myriad ways in which authors refer to characters. A named entity (e.g., *Ebenezer Scrooge*) is one way. However, there can be aliases and variations (*Mr. Scrooge*). Authors also use pronouns and descriptive nominals (*he*, *she*, *the old man*) to refer to given characters known by proper names. Nondescript characters, by definition, are only given nominals (*the porter*, *the clerk*). To identify all the characters in the text, then, we not only have to search across proper nouns, nominals and pronouns, but link together those mentions which co-refer to the same entity.

We attempted to apply two recent named entity recognition and coreference systems to the QSA corpus: ACEJet [Grishman *et al.*, 2005] and Reconcile [Stoyanov *et al.*, 2010], both of which have been applied to the ACE task. However, both were designed more for news prose than for literary fiction, which presented substantial challenges to their adoption

[3] http://www.cs.columbia.edu/nlp/tools.cgi

for this task: Neither system was able to process texts at the novella length or longer, given our available resources, as this is several orders of magnitude longer than a typical news article; also, both tools were aggressive in their merging of mentions into entities, which led to unacceptable precision losses. For instance, gender distinctions as implied by titles were sometimes ignored, such that *Mr. Weston* and *Mrs. Weston* were merged into a single entity in the case of *Emma*. We believe the complexity of literary prose, compared to the more simple syntactic structures found in newswire, calls for a new approach that is built on a distinct set of assumptions.

In order to maintain a high precision, we developed a custom pipeline for literary fiction that breaks the character identification task into three stages: First, to chunk named entities, pronouns and nominals; second, to perform coreference at a high precision, but only between named entities; and third, to roll the more difficult coreference task—pronoun resolution—into the larger quoted speech attribution task, as we do not require a general solution to this problem to address the task at hand (finding the named entity or nominal responsible for each instance of quoted speech).

Pronouns are easy to detect, as there is a closed set of words for which we can search (as we are limiting our investigation to English). Named entities and nominals are more difficult. Fortunately, chunking named entities is a task that carries over cleanly from other discourse types, such as news, so we were able to leverage publicly available tools for the task. We processed each novel with the Stanford Named Entity Recognizer [Finkel *et al.*, 2005], which applies Conditional Random Field (CRF) sequence models to the vector of words found in a document. The system was trained on the data set for the shared named entity recognition task published at the Seventh Conference on Computational Natural Language Learning (CoNLL),[4] a collection of news articles from the Reuters Corpus.[5] Each named entity is classified as either a Person, Location or Organization. In adapting the Stanford NER finder to the QSA and LSN corpora, we found that the tool sometimes classified two identical mentions in two different classes, as they appeared in different contexts (one context might suggest that *Darcy* is a person, where another appears to refer to him as

[4]Available at `http://www.cnts.ua.ac.be/conll2003/ner/#CN03`

[5]See `http://trec.nist.gov/data/reuters/reuters.html`

a place). In these cases, we assigned the class that took a plurality of "votes" across all identical mentions. We then removed all Location entities from our list of characters.

Using the development segment of the QSA corpus as a testbed, we developed a custom heuristic for extracting nominals. This method scans the pre-processed text against a regular expression that searches each line for two close (but not necessarily adjacent) tokens: a determiner and a head noun. We compiled lists of determiners and head nouns using a subset of the development corpus—determiners included the normal *a* and *the*, as well as possessives that include head nouns (e.g., *her father*, *Isabella's husband*) and both ordinal and cardinal numbers (*two women*). The text that falls between the determiner and the head noun is assumed to be a modifier, although we manually tuned the regular expressions to separate legitimate modifiers from noise. A modifier can either be a single word, or two words separated by a comma.

We compiled a list of valid head nouns by adapting a subset of the taxonomy of English words offered by WordNet [Fellbaum, 1998]. Specifically, we chose subtrees that could potentially describe an animate agent, including organisms (*the stranger*), imaginary beings and spiritual beings. This required some filtering, as the WordNet "organism" hierarchy includes many words not typically used as nouns. For instance, *heavy* is typically an adjective, but it can refer to "an actor who plays villainous roles." We trained a rule-based classifier [Cohen, 1995] on a subset of the development segment of the QSA corpus (based on our own annotations) to filter out such undesirable nouns and decrease the noise generated by the nominal chunker. Features included counts of WordNet senses for the word as an adjective, a noun and a verb, as well as the noun senses' "sense numbers." In the latter case, WordNet assigns a number to each sense to rank its prevalence in the various corpora which have been tagged against the lexicon. Words whose noun senses appeared frequently in WordNet's semantic concordance texts, especially relative to their non-noun senses, were allowed to be head nouns. The list numbered some 20,000 nouns, from *aardvark* to *Zulu*, including a fair number of compound nouns. Table 2.3 shows excerpts from Dickens, Flaubert, Chekhov and Twain (clockwise from top left), including names and nominals in bold that our system identified as character mentions outside of quoted speech.

We limited our work in coreference to grouping together named entity mentions that

"A merry Christmas, uncle! God save you!" cried **a cheerful voice**. It was **the voice** of **Scrooge's nephew**, who came upon him so quickly that this was the first intimation he had of his approach. "Bah!" said **Scrooge**, "Humbug!" He had so heated himself with rapid walking in the fog and frost, this nephew of Scrooge's, that he was all in a glow; his face was ruddy and handsome; his eyes sparkled, and his breath smoked again. *"Christmas a humbug, uncle!"* said **Scrooge's nephew.** "You don't mean that, I am sure?"	"And," said **Madame Bovary**, taking her watch from her belt, "take this; you can pay yourself out of it." But **the tradesman** cried out that she was wrong; they knew one another; did he doubt her? What childishness! She insisted, however, on his taking at least the chain, and **Lheureux** had already put it in his pocket and was going, when she called him back. *"You will leave everything at your place. As to the cloak"*—she seemed to be reflecting—"do not bring it either; you can give me the maker's address, and tell him to have it ready for me."
"Well, I do, too– LIVE ones. But I mean dead ones, to swing round your head with a string." "No, I don't care for rats much, anyway. What I like is chewing-gum." "Oh, I should say so! I wish I had some now." *"Do you? I've got some. I'll let you chew it awhile, but you must give it back to me."*	He beckoned coaxingly to **the Pomeranian**, and when **the dog** came up to him he shook his finger at it. **The Pomeranian** growled: **Gurov** shook his finger at it again. **The lady** looked at him and at once dropped her eyes. "He doesn't bite," she said, and blushed. *"May I give him a bone?"* he asked; and when she nodded he asked courteously, "Have you been long in Yalta?"

Table 2.3: Four samples of output that show the extracted character names and nominals (in bold) and quoted speech fragments (in italics).

refer to the same individual (as opposed to pronoun or nominal anaphora resolution). Our heuristic for this task is based on work we previously published in the domain of scholarly monographs about art and architecture [Davis *et al.*, 2003]. This approach finds named entity mentions that are variations of one another, grouping them into clusters that assume transitivity (in that mentions that are variations of the same entity are assumed to be variations of one another). The clustering process is as follows:

1. For each named entity, we generate variations on the name that we would expect to see in coreferent mentions. Each variation omits certain parts of multi-word names, respecting titles and first/last name distinctions. For example, *Mr. Sherlock Holmes* may refer to the same character as *Mr. Holmes*, *Sherlock Holmes*, *Sherlock* and *Holmes*. (We found that in this literary genre, feminine titles could not be removed without confusing the women's names with those of their male relatives.)

2. For each named entity, we compile a list of other named entities that may be coreferent mentions, either because they are identical or because one is an expected variation on the other.

3. We then match each mention with the most recent of its possible coreferent mentions. In aggregate, this creates a cluster of mentions for each character.

Though we also group together identical nominals as referring to the same entity, we do not attempt to find coreference between nominals, pronouns and named entities. That is, we do not perform anaphora resolution as a discrete task. Instead, we roll this ambiguity into the input for our next larger task, quoted speech attribution. When faced with such input, as we will see, the QSA solver chooses the most likely speaker from among several nearby mentions.

In order to make quoted speech attribution easier, we also pre-process the texts with an automatic tagger that assigns a gender (male, female, plural, or unknown) to as many named entities as possible. We do this first by finding mentions with gendered titles (e.g., *Mr.*), gendered head words (*nephew*) and first names as given in a gendered name dictionary (*Emma*). Then, we assume that each named entity has one gender that is shared transitively by all of its mentions. *Mr. Scrooge*, for instance, assigns a "male" tag to itself and forwards this tag to *Scrooge*, which assigns it further to *Ebenezer Scrooge*, which assigns it finally to *Ebenezer* (though redundantly, as the gender dictionary knows this name to be male). All mentions start out with the "unknown" tag; if two mentions for the same entity are tagged with opposing genders by this approach, we take a vote among all the mentions with assigned genders, and apply the gender with the plurality of votes to each mention. This assumes that all mentions refer to an entity with a consistent gender.

Although we did not conduct a formal evaluation of this component in isolation, its output was used to give the annotators of the QSA corpus "candidate" speakers for each quote. As we mentioned earlier, only in 3% of quotes was there a recall issue in which the speaker was not extracted by our tool and presented as a candidate. Meanwhile, the precision of the named entity coreference aspect is incorporated into the forthcoming evaluation of the overall social network extraction pipeline. In the future, we will apply these techniques to other genres and determine how well they perform in various contexts. Two areas to address are language independence (i.e., these steps do not work on French or German texts) and more wide-ranging, automatic coreference. One particular irony of the limited-coreference approach we have described is that characters who change names, or are called two different names at different times, are taken to be separate individuals—including, in the LSN corpus, a Dr. Jekyll on one page and a Mr. Hyde on the next.

2.5 Quoted Speech Attribution

Understanding the semantics of attributing direct and indirect speech acts to their speakers is important for tasks beyond social network extraction, such as opinion mining [Balahur *et al.*, 2009], discourse [Redeker and Egg, 2006] and even the automatic graphical visualization of a scene [Kurlander *et al.*, 1996]. This section addresses the problem of attributing instances of quoted speech to their respective speakers in narrative discourse, using the QSA corpus outlined in 2.3.

The baseline approach to this task is to find named entities near the quote and assign the quote to the one that is closest (especially if there is a speech verb nearby). However, even in the straightforward prose by these authors (compared to that of modernist authors), in many instances there is a large distance between the quote and its speaker. For example, in the following passage from Austen's *Emma*, there are several named entities near the quote, and correct attribution depends on an understanding of syntax and (to a lesser extent) the semantics of the scene:

> "Take it," said **Emma**, smiling, and pushing the paper towards **Harriet**– *"it is for you. Take your own."*

The quote "it is for you. Take your own" is preceded by two proper names in the paragraph, *Emma* and *Harriet*, of which the correct speaker is the farther of the two. In other cases, such as extended conversations, the quoted speech and the nearest mention of its speaker may be separated by 15, 20 or an even greater number of paragraphs.

2.5.1 Related Work

The pragmatics of quoted and indirect speech in literature have long been studied [Voloshinov, 1971; Banfield, 1982], but the application of natural language processing to literature with respect to this task is limited by comparison; most work in quoted speech identification and attribution has been focused on the news domain. Most recently, Sarmento and Nunes [2009] present a system for extracting and indexing quotes from online news feeds. Their system assumes that quotes fall into one of 19 variations of the expected syntactic construction `[Name] [Speech Act] [Quote]`, where *Speech Act* is one of 35 selected verbs and *Name* is a full mention (anaphoric references are not allowed). Pouliquen et al. [2007] take a similar approach in their news aggregator, identifying both universal and language-specific templates for newswire quotes against which online feeds are matched; Sagot et al. [2010] take a similar tack for French news articles. This method trades off recall for precision, since there are many syntactic forms a quote may take. Unfortunately, the tradeoff is not as favorable for literary narrative, which is less structured than news text in terms of attributing quoted speech. For example, a quote often appears by itself in a paragraph. Our approach augments the template approach with a supplementary method based on statistical learning.

The work targeting literature has covered character and point-of-view identification [Wiebe, 1990] as well as quoted speech attribution in the domain of children's literature for purposes of building a text-to-speech system [Zhang *et al.*, 2003]. Mamede and Chaleira [2004] work with a set Portuguese children's stories in their heavily rule-based approach to this task; we aim to be less reliant on rules for processing a larger corpus. Glass and Bangay [2007] focus on finding the link between the quote, its speech verb and the verb's agent. Compared to this work, we focus more on breadth (recall), as we include in our evaluation quotes that do not have speech verbs nearby.

2.5.2 Methodology

Our method for quoted speech attribution is as follows:

1. **Preprocessing.** We identify all named entities, pronouns and nominals that appear in the passage of text preceding the quote in question. These are the *candidate* speakers. For building the statistical models, they match the candidates provided to our annotators. We replace certain spans of text with backoff symbols, and clean or normalize other parts.

2. **Classification.** The second step is to classify the quote into one of a set of syntactic categories. This serves to cluster together scenarios where the syntax strongly implies a particular solution. In some cases, we choose a candidate solely based on its syntactic category.

3. **Learning.** The final step is to extract a feature vector from the passage and send it to a trained model specific to its syntactic category. There are actually n vectors compiled, one for each candidate speaker, that are considered individually. The model predicts the probability that each candidate is a *speaker*, as opposed to a *non-speaker*, then attributes the quote to the highest-probability candidate.

2.5.3 Encoding, cleaning, and normalizing

Unlike discourse tasks such as part-of-speech tagging and named entity recognition, we do not have the convenience of a small, closed set of possible tags that apply to every document. That is, while each English document will use the same small set of part-of-speech tags (noun, verb and so on), each novel will feature a different set of characters possibly numbering into the hundreds or thousands. This complicates our learning model; predictive modeling only works if the same set of possible tags is available during both training and testing.

For this reason, before we extract features for each candidate-quote data point, we encode the passage between the candidate and the quote according to a backoff model. Our purpose here is to increase the amount of data that subscribes to similar patterns by substituting generic words and phrases for specific ones. The steps include:

1. Replacing the quote and character mention in question (the *target quote* and *target character*), as well as other quotes and characters, with symbols: <TARGET_QUOTE>,

<TARGET_CHARACTER>, <OTHER_QUOTE> and <OTHER_CHARACTER>.

2. Replacing verbs that indicate verbal expression or thought with a single symbol, <EXPRESS_VERB>. We compiled the list of expression verbs by taking certain WordNet subtrees, similar to the manner in which we compiled character head nouns. We selected the subtrees based on the development corpus; they include certain senses of *express*, *think*, *talk* and *interrupt*, among others. There are over 6,000 words in this list in all, including various conjugated forms of each verb.

3. Removing extraneous information, in particular adjectives, adverbs, and adverbial phrases. We identified these by processing the passage with the MXPOST part-of-speech tagger [Ratnaparkhi, 1996].

4. Removing paragraphs, sentences and clauses where no information pertaining to quoted speech attribution seems to occur (e.g., no quotes, pronouns or names appear).

2.5.4 Dialogue chains

One crucial aspect of the quote attribution task is that an author will often produce a sequence of quotes by the same speaker, but only attribute the first quote (at the head of the *dialogue chain*) explicitly. The effect of this discourse feature is that instances of quoted speech lack conditional independence. That is, the correct classification of one quote often depends on the correct classification of at least one previous quote. We read the text in a linear fashion and attribute quotes in a cumulative fashion, maintaining a discourse model that includes the currently speaking characters. For example:

> "Bah!" said Scrooge, "Humbug!"

The added "Humbug" is implied to be spoken by the same speaker as "Bah." In general, the reader assumes that an "added" quote is spoken by the previous speaker, and that if several unattributed quotes appear in sequential paragraphs, they are two "intertwined" chains with alternating speakers. This model of reading is not tied to these authors or to this genre, but is rather a common stylistic approach to reporting conversational dialogue.

Syntactic category	Definition
Backoff	n/a
Added quote	<OTHER_QUOTE by PERSON_1> <TARGET_QUOTE>
Apparent conversation	<OTHER_QUOTE by PERSON_1> <OTHER_QUOTE by PERSON_2> <TARGET_QUOTE>
Quote-Said-Person trigram	<TARGET_QUOTE> <EXPRESS_VERB> <PERSON_1>
Quote alone	Quote appears by itself in a paragraph but "Apparent conversation" (multiple subsequent unattributed quotes) does not apply.
Anaphora trigram	<TARGET_QUOTE> <PRONOUN> <EXPRESS_VERB>
Quote-Person-Said trigram	<TARGET_QUOTE> <PERSON_1> <EXPRESS_VERB>

Table 2.4: The most prevalent syntactic categories found in the development section of the QSA corpus.

We model this dependence in both development and testing. In training statistical learners, we incorporate the annotations of speakers into the input features for subsequent quotes. In other words, for each quote, the learner sees the speaker of the previous quote. As the system processes a new text online, it attributes quotes cumulatively from the front of the text to the back, just as a human reader would. During the backoff encoding, we include the identity of each previous speaker in its respective <OTHER_QUOTE> tag (see Table 2.4). This technique has the potential to propagate an error in attributing the "head" quote of a chain to the entire chain; in the present study we evaluate each quote under the ideal condition where previous quotes are correctly identified. We leave for future work the investigation of techniques for repairing discourse-level attribution errors.

2.5.5 Syntactic categories

Our next step is to classify the quotes and their passages in order to leverage two aspects of the semantics of quoted speech: dialogue chains and the frequent use of expression verbs. A pattern matching algorithm assigns to each quote one of five syntactic categories (see Table 2.4):

- **Added quote.** Intended for links in dialogue chains, this category covers quotes that immediately follow other quotes without paragraph breaks (e.g., "Humbug!").

Syntactic category	Rate	Prediction	Accuracy
Backoff	.19	n/a	
Added quote	.19	PERSON_1	.95
Apparent conversation	.18	PERSON_1	.96
Quote-Said-Person trigram	.17	PERSON_1	.99
Quote alone	.14	n/a	
Anaphora trigram	.10	n/a	
Quote-Person-Said trigram	.02	PERSON_1	.92

Table 2.5: For each syntactic category, its prevalence in the training/development corpus, the applicable prediction (if any), and the accuracy of the prediction on the development corpus.

- **Quote alone.** A quote appears by itself in a paragraph, without an attribution. In a subcategory, **apparent conversation**, two previous paragraphs begin with quotes that are either also alone or followed by sentences without quoted speech. This case is designed to correspond to dialogue chains.

- **Character trigram.** This is a sequence of three adjacent tokens: a character mention, an expression verb and a span of quoted speech. There are six subcategories, one for each permutation (e.g., *"Bah!" said Scrooge* would be in the **Quote-Said-Person** subcategory, where "Said" refers to any expression verb and "Person" refers to a character mention).

- **Anaphora trigram.** There are six subcategories here that correspond to the six character trigrams, except that a pronoun takes the place of a character mention. Each subcategory is coded with the gender implied by the pronoun (male, female or plural speaker).

- **Backoff.** This catch-all category covers all quotes that are not covered by another category.

Two of these categories automatically imply a speaker for the quote. In *Added quote*, the speaker is typically the same as the one who spoke the preceding quote, and in character trigram categories, the mentioned character is presumably the speaker. We shall see that these implied answers are highly accurate and sometimes obviate the need for machine learning for their respective categories. We divide the remaining cases into three data sets for learning: *Backoff*, *Quote alone*, and any of the *Anaphora* trigrams. During online quote attribution, the syntactic classifier acts as a "router" that directs each quote to either an implied answer or one of the trained models.

While we implemented the classifier using our development corpus, the syntactic categories are not designed for these authors or for nineteenth century texts in particular. While they appear to cover the conventional form of dialogue expression in Western literature, genres such as epic poetry or modernism vary in form to the point where our system would place most quotes in the backoff category. The forthcoming machine learning approaches can then be retrained on a particular genre or style.

Table 2.3, which showed excerpts from Dickens, Flaubert, Chekhov and Twain (clockwise from top left), also gives examples of different syntactic categories. The four italicized quotes represent *Quote-Said-Person*, *Backoff*, *Quote-He-Said* (that is, *Anaphora*), and *Apparent conversation*, respectively.

2.5.6 Feature extraction and learning

To build the predictive models, we extract a feature vector $\vec{f}$ for each candidate-quote pair. The features include:

- The distance (in words) between the candidate and quote
- The presence and type of punctuation between the candidate and quote (including paragraph breaks)
- Among the character mentions found near the quote, the ordinal position of the candidate outward from the quote (in the anaphora cases, only gender-matching characters are counted)
- The proportion of the recent quotes that were spoken by the candidate (as annotated)
- Number of names, quotes, and words in each paragraph
- Number of appearances of the candidate
- For each word near the candidate and the quote, whether the word is an expression verb, a punctuation mark, or another character

- Various features of the quote itself, including the length, the position in the paragraph, and the presence or absence of character mentions inside the quote

Because this problem is one of choosing between candidates, we explore several ways of comparing each candidate's feature vector to those of its competitors within a set for a single quote. Specifically, we calculate the average value for each feature across the set and assemble a vector $\vec{f}_{mean}$. We then replace the absolute values for each candidate ($\vec{f}$) with the *relative distance* in value for each feature from the set norm, $|\vec{f} - \vec{f}_{mean}|$. We similarly experiment with sending $|\vec{f} - \vec{f}_{median}|$, $|\vec{f} - \vec{f}_{product}|$, $|\vec{f} - \vec{f}_{max}|$ and $|\vec{f} - \vec{f}_{min}|$ to the learners.

We applied three learners to the data. Each creates a model for predicting *speaker* or *non-speaker* given any candidate-quote feature vector. Namely, they are J48, JRip and a two-class logistic regression model with a ridge estimator, all as available in the WEKA Toolkit [Hall *et al.*, 2009]. Because these give binary labels and probability scores for each candidate separately, the final step is to **reconcile** these results into a single decision for each quote. We tried four alternate methods:

- In the **label** method, we simply scan all candidates for one that has been classified *speaker*. If more than one candidate is classified *speaker*, the attribution remains ambiguous as no speaker is identified. If no speaker is found, the quote is determined to be non-dialogue. Overattributions (where a speaker is given to non-dialogue), underattributions (where no speaker is identified for dialogue) and misattributions (where the wrong speaker is identified) all count as errors.

- The **single probability** method discards the labels and simply uses the probability, supplied by each classifier, that each candidate belongs in the *speaker* class. When these probabilities are ranked, the candidate with the highest probability is taken as the speaker—-unless the probability falls below a certain threshold, in which case we conclude the quote is non-dialogue (no speaker). We vary the threshold t as a parameter.

- The **hybrid** method works the same as the "label" method, except in case more than one candidate is labeled as *speaker*, the algorithm backs off to the single-probability method to find the best choice.

Syntactic category	Rate	Solver	Feature vector	Reconciliation method	Acc.
Quote-Said-Person	.22	**Category prediction**			**.99**
		Logistic+J48	$\vec{f} - \vec{f}_{min}$	Max. ($t = .02$)	.96
Added quote	.19	**Category prediction**			**.97**
		J48	$\vec{f}$	Hybrid	.97
Backoff	.18	**Logistic+J48+JRip**	$\vec{f}$	Mean ($t = .08$)	**.64**
Quote alone	.16	**Logistic+J48+JRip**	$\vec{f} - \vec{f}_{mean}$	Mean ($t = .03$)	**.63**
		Logistic+J48+JRip	$\vec{f}$	Mean ($t = .07$)	.65
Apparent conversation	.12	**JRip**	$\vec{f} - \vec{f}_{min}$	Hybrid	**.93**
		JRip	$\vec{f} - \vec{f}_{mean}$	Hybrid	.94
		Category prediction			.91
Anaphora trigram	.09	**Logistic**	$\vec{f} - \vec{f}_{mean}$	Mean ($t = .01$)	**.63**
		Logistic+JRip	$\vec{f} - \vec{f}_{median}$	Mean ($t = .02$)	.64
Quote-Person-Said	.04	**JRip**	$\vec{f}$	Hybrid	**.97**
		Logistic+JRip+J48	$\vec{f}$	Mean/median ($t = .02$)	.94
		Category prediction			.93
Overall	1.0	In bold above			**.83**
Baseline	1.0	Most recent			.45
Baseline	1.0	Closest			.52

Table 2.6: Performance of both category predictions and trained models on the test set of the QSA corpus, for each syntactic category.

- The **combined probability** method works the same as the single probability method, except the probability of each candidate being *speaker* is derived by combining two or three of the probabilities given by the classifiers. We ran all permutations of classifiers and combined their results in four ways: mean, median, product and maximum, as suggested by Kittler et al. [1998].

2.5.7 Results and discussion

Table 2.6 shows the performance on the attribution task of both the category predictions and the trained models over the test set, with the latter using 10-fold cross-validation. Only the top-performing classifier permutations are shown for each category. For example, a combination of logistic regression, J48 and JRip, whose input features were absolute (rather than relative) and whose output probabilities were averaged before they were ranked, was trained and tested on all data in the backoff class. It correctly identified the speaker (or lack of speaker) with 64% accuracy. Parameter tuning was done independently for each category. We achieved particularly high learning results in the categories where the speaker can be determined by the category alone (such as *Added quote*); the decision tree learners are effectively deriving rules similar to those that we coded manually, obviating the need for hard-coded category predictions.

The *Rate* column in Table 2.6 shows the prevalence of each syntactic category in the testing corpus; these proportions differ only slightly from those in the development corpus (Table 2.4). When we sum the accuracy scores and weigh each according to their rates in the test set, we find an overall accuracy of .83. To ensure that we are not optimizing our classifier parameters for the test set, we separated out the parameter tuning process by having the test set adopt the classifier permutations that performed the best on the development set (one for each syntactic category). The overall accuracy over the test set with these learners was .80, suggesting that the classifier parameters are not overfitting the data. For purpose of comparison, a baseline that attributes a quote to the most recently seen character gives the correct speaker in only 45% of cases. A smarter baseline that takes the closest occurring character, whether it appears before or after the quote, has an accuracy of only .52. Our results clearly show a significant improvement over these baselines.

2.6 Conversational Network Construction

We applied the results from our character identification and quoted speech attribution methods toward the construction of conversational networks from literature. We derived one network from each text in our corpus.

We first assigned vertices to character entities that are mentioned repeatedly throughout the novel. Coreferent mentions of the same entity (such as *Mr. Darcy* and *Darcy*) were grouped into the same vertex. We found that a network that included incidental or single-mention named entities became too noisy to function effectively, so we filtered out the entities that are mentioned fewer than three times in the novel or are responsible for less than 1% of the named entity mentions in the novel.

We assigned weighted, undirected edges between vertices that represent *adjacency* in quoted speech fragments. Specifically, we set the weight of each undirected edge between two character vertices to the total length, in words, of all quotes that either character speaks from among all pairs of adjacent quotes in which they both speak—implying face to face conversation. We empirically determined that the most accurate definition of "adjacency" is one where the two characters' quotes fall within 300 words of one another with no attributed quotes in between. When such an adjacency is found, the length of the quote is added to the edge weight, under the hypothesis that the significance of the relationship between two individuals is proportional to the length of the dialogue that they exchange. Finally, we normalized each edge's weight by the length of the novel.

A sample network, automatically constructed in this manner from Jane Austen's *Mansfield Park*, is shown in Figure 2.1. The width of each vertex is drawn to be proportional to the character's share of all the named entity mentions in the text (so that protagonists, who are mentioned frequently, appear in larger ovals). The width of each edge is drawn to be proportional to its weight (total conversation length). The figure was rendered by Graphviz, an open-source graph visualization package, using a stress-majorization approach [Gansner and North, 2000]. Additional samples are provided in Appendix A.

We also experimented with two alternate methods for identifying edges, for purposes of comparing against a baseline:

1. The "correlation" method divides the text into 10-paragraph segments and counts the number of mentions of each character in each segment (excluding mentions inside quoted speech). It then computes the Pearson product-moment correlation coefficient for the distributions of mentions for each pair of characters. These coefficients are used for the edge weights. Characters that tend to appear together in the same areas of the novel are taken to be more socially connected, and have a higher edge weight.

2. The "spoken mention" method counts occurrences when one character mentions another in his or her quoted speech. These counts, normalized by the length of the text, are used as edge weights. The intuition is that characters who mention one another are more likely to be socially connected.

To check the accuracy of our method for extracting conversational networks, we conducted an evaluation involving four of the novels (*The Sign of the Four*, *Emma*, *David*

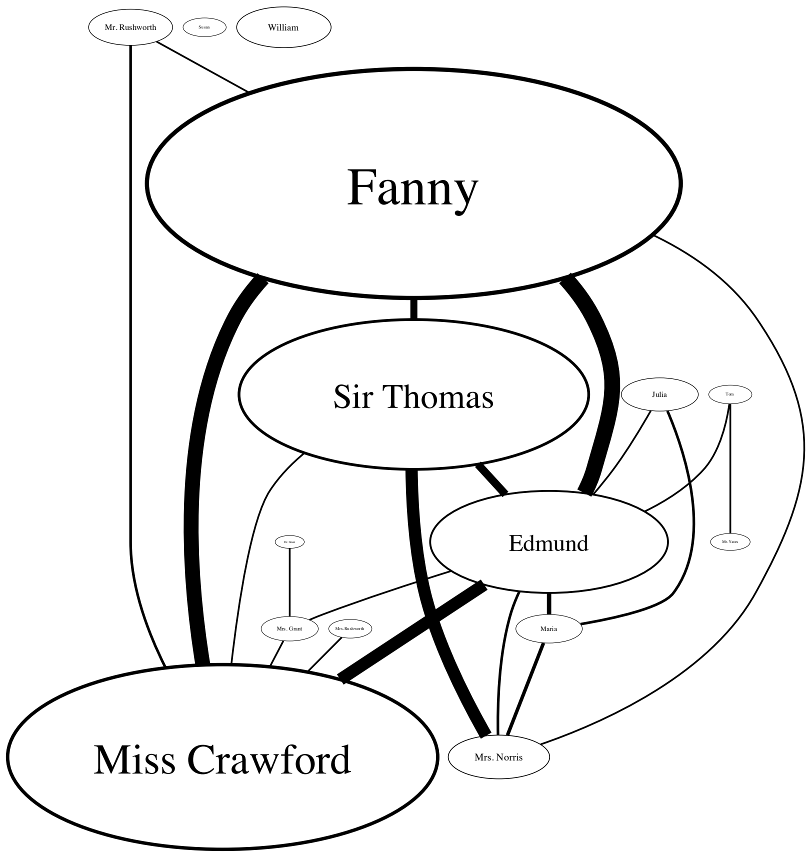


Figure 2.1: Automatically extracted conversation network for Jane Austen's *Mansfield Park*.

Copperfield and *The Portrait of a Lady*). We did not use these texts when developing our method for identifying conversations. For each text, we randomly selected 4-5 chapters from among those with significant amounts of quoted speech, so that all excerpts from each novel amounted to at least 10,000 words. We then asked three annotators to identity the conversations that occur in all 44,000 words. We requested that the annotators include both direct and indirect (unquoted) speech, and define "conversation" as in the beginning of Section 2.3, but exclude "retold" conversations (those that occur within other dialogue).

We processed the annotation results by breaking down each multi-way conversation into all of its unique two-character interactions (for example, a conversation between four people indicates six bilateral interactions). To calculate inter-annotator agreement, we first compiled a list of all possible interactions between all characters in each text. In this model, each annotator contributed a set of "yes" or "no" decisions, one for every character pair.

Method	Precision	Recall	F
Speech adjacency	.95	.51	.67
Correlation	.21	.65	.31
Spoken-mention	.45	.49	.47

Table 2.7: Precision, recall, and F-measure of three methods for detecting bilateral conversations in literary texts.

We then applied the kappa measurement for agreement in a binary classification problem [Cohen, 1960]. In 95% of character pairs, annotators were unanimous, which is a high agreement of $k = .82$.

The precision and recall of our method for detecting conversations is shown in Table 2.7. Precision was .95; this indicates that we can be confident in the specificity of the conversational networks that we automatically construct. Recall was .51, indicating a sensitivity of slightly more than half. There were several reasons that we did not detect the missing links, including indirect speech, quotes attributed to anaphoras or coreferent nominals, and large conversations in which not all participants speak in turn with each of her peers.

To calculate precision and recall for the two baseline social networks, we set a threshold t to derive a binary prediction from the continuous edge weights. The precision and recall values shown for the baselines in Table 2.7 represent the highest performance we achieved by varying t between 0 and 1 (maximizing F-measure over t). Both baselines performed significantly worse in precision and F-measure than our quoted-speech adjacency method for detecting conversations.

2.7 Data Analysis

We extracted features from the conversational networks that emphasize the complexity of the social interactions found in each novel:

1. The number of characters and the number of speaking characters
2. The variance of the distribution of quoted speech (specifically, the proportion of quotes spoken by the n most frequent speakers, for $1 \leq n \leq 5$)
3. The number of quotes, and proportion of words in the novel that are quoted speech

4. The number of 3-cliques and 4-cliques in the social network

5. The *average degree* of the graph, defined as

$$\frac{\sum_{v \in V} |E_v|}{|V|} = \frac{2|E|}{|V|} \tag{2.1}$$

where $|E_v|$ is the number of edges incident on a vertex v, and $|V|$ is the number of vertices. In other words, this determines the average number of characters connected to each character in the conversational network ("with how many people on average does a person converse?").

6. A variation on *graph density* that normalizes the average degree feature by the number of characters:

$$\frac{\sum_{v \in V} |E_v|}{|V|(|V|-1)} = \frac{2|E|}{|V|(|V|-1)} \tag{2.2}$$

By dividing again by $|V| - 1$, we use this as a metric for the overall connectedness of the graph: "With what *percent* of the entire network (besides herself) does each person converse, on average?" The weight of the edge, as long as it is greater than 0, does not affect either the network's average degree or graph density.

2.7.1 Results

We derived results from the data in two ways. First, we examined the strengths of the correlations between the features that we extracted (for example, between number of character vertices and the average degree of each vertex). We used Pearson's product-moment correlation coefficient in these calculations. Second, we compared the extracted features to the metadata we previously assigned to each text (e.g., urban vs. rural).

Hypothesis 1, which we described in Section 2.2, claims that there is an inverse correlation between the amount of dialogue in a nineteenth-century novel and the number of characters in that novel. We did not find this to be the case. Rather, we found a weak but positive correlation (r=.16) between the number of quotes in a novel and the number of characters (normalizing the quote count for text length). There was a stronger positive

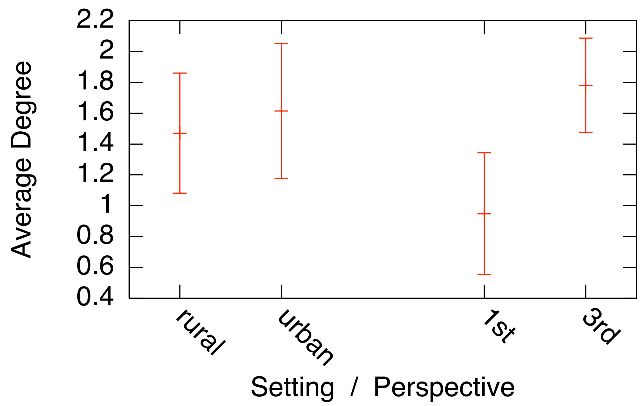


Figure 2.2: The average degree for each character as a function of the novel's setting and its perspective.

correlation (r=.50) between the number of unique speakers (those characters who speak at least once) and the normalized number of quotes, suggesting that larger networks have more conversations than smaller ones. But because the first correlation is weak, we investigated whether further analysis could identify other evidence that confirms or contradicts the hypothesis.

Another way to interpret hypothesis 1 is that social networks with more characters tend to break apart and be less connected. However, we found the opposite to be true. The correlation between the number of characters in each graph and the average degree (number of conversation partners) for each character was a positive, moderately strong r=.42. This is not a given; a network can easily, for example, break into minimally connected or mutually exclusive subnetworks when more characters are involved. Instead, we found that networks tend to stay close-knit regardless of their size: Even the density of the graph (the percentage of the community that each character talks to) grows with the total population size at r=.30. Moreover, as the population of *speakers* grows, the density is likely to increase at r=.49. A higher number of characters (speaking or non-speaking) is also correlated with a higher *rate* of 3-cliques per character (r=.38), as well as with a more balanced distribution of dialogue (the share of dialogue spoken by the top three speakers decreases at r=−.61). This evidence suggests that in nineteenth-century British literature, the larger communities are no less connected than the small ones.

Hypothesis 2, meanwhile, posits that a novel's setting (urban or rural) would have

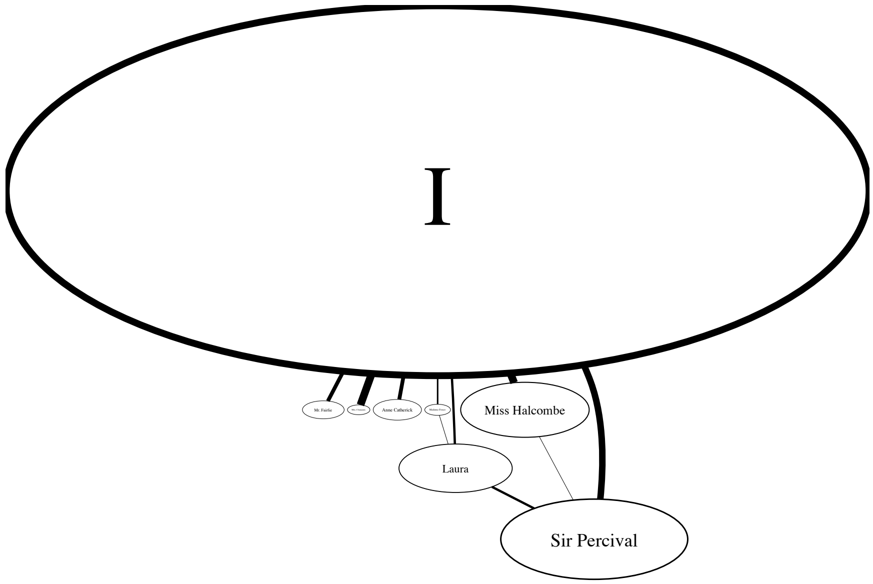


Figure 2.3: Conversational networks for first-person novels like Collins's *The Woman in White* are less connected due to the structure imposed by the perspective.

an effect on the structure of its social network. After defining a "social network" as a conversational network, we did not find this to be the case. Surprisingly, the numbers of characters and speakers found in the urban novel were *not* significantly greater than those found in the rural novel. Moreover, each of the features we extracted, such as the rate of cliques, average degree, density, and rate of characters' mentions of other characters, did not change in a statistically significant manner between the two genres. For example, Figure 2.2 shows the mean over all texts of each network's average degree, separated by setting into urban and rural with intervals representing 95% confidence. The increase in degree seen in urban texts is not significant.

Rather, the only metadata variable that *did* impact the average degree with any significance was the text's perspective. Figure 2.2 also separates texts into first- and third-person tellings and shows the means and confidence intervals for the average degree measure. Stories told in the third person had much more connected networks than stories told in the first person: not only did the average degree increase with statistical significance (by the homoscedastic t-test to $p < .005$), so too did the graph density ($p < .05$) and the rate of 3-cliques per character ($p < .05$).

We believe the reason for this can be intuited with a visual inspection of a first-person graph. Figure 2.3 shows the conversational network constructed for Collins's *The Woman in White*, which is told in the first person. Not surprisingly, the most oft-repeated named entity in the text is *I*, referring to the narrator. More surprising is the *lack* of conversational connections between the auxiliary characters. The story's structure revolves around the narrator and each character is understood in terms of his or her relationship to the narrator. Private conversations between auxiliary characters would not include the narrator, and thus do not appear in a first-hand account. An "omniscient" third person narrator, by contrast, can eavesdrop on any pair of characters conversing. This highlights the importance of detecting reported and indirect speech in future work (which is difficult, e.g., [Banfield, 1973; Oltean, 1993]), as a first-person narrator may hear about other connections without witnessing them directly.

2.7.2 Literary Interpretation of Results

Our data, therefore, do not confirm hypothesis 1. They also suggest, in relation to hypothesis 2 (also not confirmed by the data), a strong reason why.

One of the basic assumptions behind hypothesis 2, that urban novels contain more characters in a manner mirroring the masses of nineteenth-century cities, is not borne out by our data. Our results do, however, strongly correlate a point of view (third-person narration) with more frequently connected characters, implying tighter and more talkative social networks.

We would propose that this suggests that the form of a given novel (the standpoint of the narrative voice, i.e., whether the voice is "omniscient" or not) is far more predictive of the kind of social network described in the novel than where it is set or even the number of characters involved. While standard accounts of nineteenth-century fiction follow Bakhtin's notion of the chronotope in their emphasis on the content of the novel as determinative (such as where it is set, and whether the novel fits within a genre of "village" or "urban" fiction), we have found that content to be surprisingly irrelevant to the shapes of the social networks. Bakhtin's influential theory (like its detailed re-workings by Williams, Moretti, and others) suggests that as the novel becomes more urban, its form changes to accommodate the looser, more populated, less conversational networks of city life. Our data suggest the opposite: that the "urban novel" is not as strongly distinctive a form as has been asserted, and that in fact it can look much like the village fictions of the century, as long as the same method of narration is used.

This conclusion leads to some further considerations. We are suggesting that the important element of social networks in nineteenth-century fiction is not where the networks are set, but from what standpoint they are imagined or narrated. We must remember that these authors set out to tell stories, rather than to document a time and a place in terms of its social connectedness. A "small" story, one focused on a single individual and his or her own connections, does not become large simply because are unseen others behind the walls. Narrative voice, in sum, trumps setting.

2.8 Conclusion

In this chapter, we presented a method for characterizing a corpus of literary fiction by extracting the network of social conversations that occur among its characters. This allowed us to take a systematic and wide look at a large corpus of texts, an approach which complements the narrower and deeper analysis performed by literary scholars and can provide evidence for or against some of their claims. In particular, we described a high-precision method for detecting face-to-face conversations between two named characters in a novel, and showed that as the number of characters in a novel grows, so too do the cohesion, interconnectedness and balance of their social network. In addition, we showed that the form of the novel (first- or third-person) is a stronger predictor of these features than the setting (urban or rural). As a model of narrative discourse, the conversational network is feasible and intrinsically useful to the study of literature. In the next chapter, we will supplement these results by introducing an alternate model that captures a separate and larger set of narrative relations.

Chapter 3

Story Intention Graphs

In Chapter 2, we demonstrated methods for identifying a particular type of relationship in narrative discourse (conversational networks). We showed this to be a descriptive aspect of a corpus of Victorian novels. A story, though, cannot be completely described by its social structure alone. There is more that separates stories from non-stories, as not every document with quoted speech is necessarily a story (transcripts of meetings and email exchanges being two counterexamples).

We now turn our attention to a separate discourse model, one that broadens our view of "storiness" beyond quoted speech. We first ask: What kind of model can best capture the essence of a story? What *is* that essence? We then ask: Can we define a set of intratextual relations specific to our idea of the essence of narrative discourse?

This chapter attempts to answer these questions, first by defining our goals for a narrative representation (Section 3.1), then by reviewing prior models of narrative discourse (Section 3.2), then by describing a new representation (Section 3.3). Our contribution is a set of discourse relations which we collectively call the Story Intention Graphs (SIG). In Chapter 4, we describe the platform and annotation tool we have developed for building and managing a corpus of SIG encodings based on well-known narratives. Finally, Chapter 5 shows that SIGs are an effective formalism for reasoning about stories and their connections to one another.

3.1 Goals For A New Representation

Narrative is the structural scaffold for describing human experience, and so interest in describing narrative structure cuts across a swath of disciplines from the humanities to social sciences and artificial intelligence. People have told stories to one another since before the invention of writing, and some of our earliest recorded thinkers found the proper structure of a story to be worthy of inquiry [Aristotle, 1961]. The notion that a textual narrative invokes any regimented structure at all has swung in and out of fashion repeatedly over the last century among literary theorists, with post-structuralism being a dissenting view. As in any artistic medium, the idea that there are rules that govern "proper" storytelling is an invitation for avant-garde artists to break the rules and further explore the creative space. Still, for centuries, literary theorists have followed Aristotle in describing the form and content of story, with respect to genre, medium, author, time period, meter, and many other aspects. Modeling narrative is often a means to explore an aspect of the human condition. Some symbolic models, for instance, carry the intention of mimicking our cognitive processes [Graesser *et al.*, 1994]; others use structure as a tool for comparing cultures by their mythologies [Campbell, 1949].

Our purposes are pragmatic by comparison. As we mentioned in the Chapter 1, we aim to find computational methods to look beyond the surface form of a text to compare and contrast stories based on content (as opposed to style). Two stories, even very short ones, may have similar distributions of words at the surface level, and yet convey very different meanings as narrative artifacts. Consider E. M. Forster's [1990, 87] classic distinction between a non-story and a story:

1. The king died. Then the queen died.
2. The king died. Then the queen died of grief.

Both tell a sequential series of events, both involve the same actions (two deaths), and the word overlap is almost total. While the topic of death is inherently dramatic, (2) is more of a story than (1) because it relates the two events in a coherent manner. It is not simply a matter of attributing causation to the queen's death; "the queen died of old age" is not a suitable replacement for "the queen died of grief," which brings to mind images of longing, of a figure bereft to the point of physiological breakdown. Forster also considered this a type of mystery plot: "The queen died, no one knew why, until it was discovered that it was through grief at the death of the king." Here, as well as in detective stories, the presence of a thematically relevant, unanswered question piques interest in the receiver.

Such facets are examples of what we call **thematic content**: those qualities that make a story interesting, tellable, memorable, and otherwise evocative of a receiver's emotional investment. Thematic content is the *point* that the teller is trying to convey to the reader [Wilensky, 1982] that separates a story from any chronologically ordered list of events. We search for thematic content when we reflect on a story and think, "What message did it send? Why did I care?" Sometimes, as in poetry, the answer lies closer to the words and the images or patterns they invoke. We are interested in the embedded meaning that persists even when a story is paraphrased, transmuted from one medium to another, or passed from one generation to the next. As we have noted, natural language processing tends to focus on lexical and syntactic features of language; even when artificial intelligence attempts to parse out the deepest meanings of textual stories, the thematic aspect is usually secondary to other concerns [Hidi and Baird, 1986]. With a proper representation, we can identify not only the thematic content of a single story, but the analogical connections between stories: those aspects of thematic content which are common across a group or a genre.

To take a longer example than Forster's, consider the two fables in Table 3.1: "The Wily Lion" and "The Fox and the Crow".[1] We intuit that both are stories, in that they convey meaning beyond the sum of the actions that are described. Both feature discourse relations that set them apart from non-narrative discourse, such as that between an action and the goal that the action is undertaken to fulfill. They are quite analogous: in both cases, a "predator" schemes to obtain something it wants from a "prey" animal who falls for a deception. Both may convey a certain ethical message, advising readers to be conscious of the possibility of ulterior motives when suspect individuals begin to purvey flattery. They take place in a world not totally like our own—ours has no talking lions or foxes—but

[1] The full texts of all of the fables attributed to Aesop that we use in this thesis, as translated by Jones [1912], are reproduced in Appendix D. "The Wily Lion" is fable P469 from the Perry index of these fables [Perry, 2007]; "The Fox and the Crow" is P124.

A Lion watched a fat Bull feeding in a meadow, and his mouth watered when he thought of the royal feast he would make, but he did not dare to attack him, for he was afraid of his sharp horns.
Hunger, however, presently compelled him to do something: and as the use of force did not promise success, he determined to resort to artifice.
Going up to the Bull in friendly fashion, he said to him, "I cannot help saying how much I admire your magnificent figure. What a fine head! What powerful shoulders and thighs! But, my dear friend, what in the world makes you wear those ugly horns? You must find them as awkward as they are unsightly. Believe me, you would do much better without them."
The Bull was foolish enough to be persuaded by this flattery to have his horns cut off; and, having now lost his only means of defense, fell an easy prey to the Lion.

A Crow was sitting on a branch of a tree with a piece of cheese in her beak when a Fox observed her and set his wits to work to discover some way of getting the cheese.
Coming and standing under the tree he looked up and said, "What a noble bird I see above me! Her beauty is without equal, the hue of her plumage exquisite. If only her voice is as sweet as her looks are fair, she ought without doubt to be Queen of the Birds."
The Crow was hugely flattered by this, and just to show the Fox that she could sing she gave a loud caw. Down came the cheese, of course, and the Fox, snatching it up, said, "You have a voice, madam, I see: what you want is wits."

Table 3.1: "The Wily Lion" (top) and "The Fox and the Crow".

that does not stop us from seeing how the world *is* like our own and how the events might somehow relate to our own experiences. We will use "The Wily Lion" repeatedly throughout this chapter as a keystone as we examine prior attempts to develop a descriptive symbolic language for narrative discourse, and then introduce our own.

An approach to finding and describing story analogies like these would provide both intrinsic and extrinsic benefits. Intrinsically, it would shed light on the nature of the stories themselves, as well as on the discourse comprehension processes that allow us to find meaning in them. Extrinsically, a system able to extract narrative themes and tropes from discourse would help us better organize our own narrative and those we experience online or through other media. It could, for instance, let us search news articles, blog posts, and historical literature for all narratives that fit certain parameters—a search that would return entirely different results than one based on keywords. It could find opposing viewpoints on the same story and help us better understand how, in areas such as politics, two groups can see the same events as belonging to diametrically opposed and difficult-to-reconcile narratives.

Any attempt to build such a system must begin with a complex decision: How should the story be represented symbolically? Which aspects of narrative do we choose to "reify" (formally represent as a narrative primitive) because we expect them to recur from story to story? Any system of symbols implicitly commits to a particular structuralist reading of narrative. If we are to go beyond keyword searching, we must first assert that there is narrative meaning to be found beyond the surface word, and describe a formal representational schemata that captures that meaning. As narrative touches many aspects of logic, language, and culture, this is not a simple task. Let us start by drawing certain boundaries around the space of possible models.

We would like our representation to satisfy three criteria:

- First, we seek a **robust** model, emphasizing the key elements of a narrative rather than attempting to model the entire semantic world of the story at a high precision. Most of the work in story understanding today occurs in the planning or logic communities; of particular note are the long-term efforts at modeling stories in first-order logic [Mueller, 2004; Mueller, 2006; Zarri, 1997; Zarri, 2010] and other formal representations for plans and strategies [Hobbs and Gordon, 2005; Christian and Young, 2003; Riedl and Young, 2004]. The systems based on these models can compute complex inferences about the story-world and project what can or should happen at some future point in a story. To accomplish this, their models strongly emphasize commonsense knowledge and rule-based inference. Such an approach can come close to addressing the subject matter we are trying to model. In one case, a rule-based system was able to answer questions about the ethics of the stories it read [Reeves, 1991]. It could indicate when people were acting selfishly or selflessly. Mueller's work similarly uses a theorem prover and a database of commonsense axioms to make inferences beyond the first-order assertions that are provided in the story. This is useful for certain *why* explanations, such as knowing in "The Three Little Pigs" that the house made of straw fell because it was weaker than the house made of bricks.

  The drawback of this approach is that the knowledge base tends to be rigid and narrow compared to the information found even in a single novel. No commonsense theorem prover has very wide coverage today, and it is currently infeasible to extract a model of such precision from a discourse automatically. For our purposes, we need to model less about *what exactly is happening* than *why it makes for an interesting story*.

This is not to say that we are completely against first-order logic; as we will see, we use this formalism as a way of enriching certain aspects of our representation when they are essential to the thematic content we are trying to capture. For example, our model does not intrinsically understand the differences between ethnic races, but if such a difference is relevant to the discourse relations we are targeting in a particular story (e.g., an important motivation in a story about prejudice), such semantics can be represented within the scope of that story. As Smith and Hancox [2001] noted when describing their own criteria for a narrative representation, different individuals can interpret a story at different levels of specificity. Our model will provide a framework for a descriptive first-order representation that allows the precision of the representation to scale up or down as needed. This approach to being tolerant of partial encodings is what we mean by robustness.

- Second, we seek an **expressive and computable** model of thematic content. As a counterbalance to the first criterion, we wish to find a model that is formal enough to allow us to find analogies, identify patterns, and design summaries of narrative content. A model that uses lexical features to distinguish stories from non-stories [Gordon and Swanson, 2009], for example, is useful for building corpora of stories found online [Swanson and Gordon, 2008], but is only the starting point for distinguishing stories from one another. One recent attempt to find discourse patterns in news stories [Chambers and Jurafsky, 2008a] uses lexical and semantic similarities among words in each story to find "narrative event chains," in which the same kinds of verbs are applied in the same relative ordering to the same set of protagonists among a collection of articles. While this is a form of analogy, it is specific to news as a genre of narrative—one in which the behavior of the "focalizer" (narrating agent [Genette, 1983; Bal, 1997]) is distinctive. While a journalist typically tries to tell the facts underlying the story transparently, a writer of fiction will strategically withhold, re-order of obfuscate information for thematic effect. Thus, the key challenge is not just to determine how many relations to extract for representing cross-domain narratives, but to choose the *right* relations that will strike a balance between wide coverage and semantic precision for this particular task. A story that a reasonable reader would find to have thematically engaging content should have a corresponding encoding under the model that represents those aspects which make the story engaging.

- Finally, we seek a model that is **accessible**. That is, it should be amenable to manual annotation by human subjects for purposes of building a data bank of narratives. The annotation methodology must be simple and well-documented enough for trained annotators, or even lay readers, to learn. The model should be open-domain, rather than assume a particular input corpus or genre.

  Our inspiration here is prior large-scale annotation projects such as the Penn Treebank, which provides syntactic markup of the Wall Street Journal corpus [Marcus *et al.*, 1993], and PropBank, which provides semantic role labeling for individual sentences [Kingsbury and Palmer, 2002]. The Penn Discourse Treebank provides a corpus of documents annotated according to a model of rhetorical relations [Prasad *et al.*, 2008], as does RSTBank [Carlson *et al.*, 2003]. The TimeBank corpus does the same for temporal relations [Pustejovsky *et al.*, 2003b], according to the TimeML model [Mani and Pustejovsky, 2004].

  In this spirit, we will present a schemata that we have used to collect a corpus we call **DramaBank**. Publicly released,[2] it consists of textual narratives annotated with the discourse relations that we find to represent thematic content. It is our hope that DramaBank will enable further work on narrative discourse parsing, as the Penn Discourse Treebank corpus has enabled recent work on discourse parsers for more expository texts [Lin *et al.*, 2010]. We discuss DramaBank in Chapter 5.

Let us now walk through some of the models of narrative content that have previously been proposed, and consider how well they meet our needs.

[2] http://www.cs.columbia.edu/~delson

3.2 A Brief History of Narrative Modeling

We discuss prior narrative models in three categories. First, we consult cognitive psychology and settle on a particular approach to modeling narrative meaning. Second, we review discourse models in general, with an emphasis on story grammars. Finally, we review prior models used in artificial intelligence and natural language processing.

3.2.1 Foundations in Cognitive Psychology

The most crucial starting point for teasing apart the meaning of a narrative is the distinction between what the story *is* and how the story is *told*. This distinction goes by many names. Todorov [1966] named the plane of story content *histoire* and that of style and point-of-view *discours*. Russian Formalism uses the terms *fabula* and *sjuzhet*, respectively. Genette [1972] refers to the *diegetic level* and *extradiegetic level*. Bal [1997] makes a three-level distinction: the *fabula*, which she defines as "a series of logically and chronologically related events that are caused or experienced by actors; the *story*, in which a narrator (perceiver) selects some elements of the *fabula* to convey and omits others; and the *text*, where words are chosen convey the story in a discourse." Whichever set of terms is used, the effect is the same. A narrative discourse is a lens that focuses attention on a particular combination of events that transpire in a constructed world. Sometimes the narrator purports that the *fabula* corresponds to our reality, as in a news article by a reliable journalist; in fiction, the world of the story borrows many elements from the actual world but invents others [Eco, 1995].

Much of what happens in a *fabula* remains unsaid, or at least withheld from the reader. Sometimes this is dramatically essential. It would defeat the purpose of a mystery novel for the identity of the killer to be revealed in the first chapter, when the murder takes place. Rather, this crucial fact is omitted by the narrator, who invites the reader to follow the detective's investigation toward a solution which satisfies all the facts in a causally plausible fashion. Usually, though, these omissions are made for purposes of narrative economy, because they have no bearing on the thematic content of the story. The sentence "John drove to the store," for example, omits certain elements of its *fabula*: what kind of vehicle John drove, where he was driving from, the time of day, the time of year, the duration of the trip, and so on. When the missing facts are needed to give a story *coherence*, or intra-connectedness as a united discourse, we infer them. For instance, in the Forster citation above, we may infer that the queen died "of grief" in order to connect the two sentences. The thematic content of the story builds off of such inferences, which look beyond the words of the text in search of particular facets of discourse coherence. We strive to understand why the various parts of a story were included in the discourse by the narrating agent, and how all the parts fit together coherently. When we find such coherence, the story takes on a meaning greater than the sum of its sentences.

But what, in general, are the facets of meaning that we search for? What makes a story more interesting than a set of chronological facts in a *fabula*? In short, what are the "primitive" symbols of tellable narratives that we can reify in a representation that will allow us to find thematic similarities, differences, patterns and analogies?

Cognitive psychology has been examining these questions in a search for an understanding of the way the mind comprehends discourse. Narrative, as opposed to expository or persuasive discourse, has been a common testbed for understanding the way inferences are generated during reading [Graesser *et al.*, 1991a]. Using methods such as question-answering response time and "think-aloud," when subjects articulate their inferential processes while reading, they track the way various elements of the story are retained in working memory, discarded, linked together and inferred from background knowledge. As story interpretation is subjective to each receiver, their findings on the process of reading are relevant to the design of our schemata.

Most discourse psychologists subscribe to a three-layer model of discourse representation analogous to Bal's text-story-*fabula* system [Graesser *et al.*, 1997]. First proposed by van Dijk, a linguist, and Kintsch, a psychologist [van Dijk and Kintsch, 1983], the model begins with a *surface code* which (like Bal's text layer) recalls the words on the page. In cases like poetry where wording is important, this is retained; in news articles and most other forms of narrative, it is quickly discarded in favor of representations of deeper meaning. The second layer, the *textbase*, contains propositions that convey the essence of what was in the surface code. These are predicate-argument structures, where each argument has a semantic role that relates to the predicate; this is very similar to the first-order representations used in semantic approaches by Mueller [2003] and others. This, in turn, is boiled down to a *situation model* [Zwaan and Radvansky, 1998], which represents the "aboutness" of the text—the states and affairs which are the essence of the story.

As the reader reads, she attempts to integrate each new sentence in the discourse into the situation model using a combination of top-down and bottom-up processes. Psychologists have put forward several theories describing how this is accomplished [Kintsch, 1988; Zwaan *et al.*, 1995]. Most models, though, share a consensus view that a few aspects of a narrative are consistently reified in our cognitive search for meaning: **intentional agents** and the **goals** that they would like to see transpire in the story-world.

The evidence for the primacy of these elements reaches into early development. Even before we can read, we see agency in the most abstract of events; separating intention from action is a basic function of narrative perception [Bundgaard, 2007]. In one classic study, Heider and Simmel [1944] showed subjects short animations of geometric shapes moving around a screen. When asked to describe what they had seen, most subjects told stories about animate agents (typically people), the challenges they faced, the love they defended, the assistance they rendered to the needy, and so on. Though the sample size was small, this effect has been experimentally repeated: Subjects easily attribute mental states, involving goals and intentions, to even highly abstract stimuli [Dik and Aarts, 2007]. Other studies show that infants as young as nine months interpret actions around them as causally related to the goals of their agents [Csibra *et al.*, 1999; Gergely *et al.*, 1995]. One study showed that children as young as six have a better recall of oral stories when characters have well-defined goals [Lynch and van den Broek, 2007]. The children were able to make online inferences as they listened, to connect subgoals to superordinate goals and the actions taken in pursuit of those goals.

These results help explain the large and growing body of evidence in experimental psychology supporting the claim that readers engage in a search for the motivations behind characters' actions, and are better at retaining and retrieving information about the actions of the story when they are backed by clear goals. In other words, readers are actively searching for goals in order to better comprehend a text [Lichtenstein and Brewer, 1980; Hassin *et al.*, 2005; Aarts *et al.*, 2008]. The nature of the types of inferences and connections that are made, and their timing during the reading process, is the matter of longstanding debate. McKoon and Ratcliff [1992], for instance, advocate a "minimalist" hypothesis in which only those inferences which are necessary for local coherence—sentence to sentence consistency—are made online, and global connections (connecting distant elements of the story) are made only when local coherence is violated by new information. Graesser et al. [1994] propose instead a "constructionist" model in which online inferences include causal antecedents (what caused an action), superordinate goals (what motivates an action) and how a character's emotional state is affected by an action. They make the crucial point that these findings relate to narrative discourse rather than expository, persuasive or descriptive texts, because narrative is a "privileged" kind of discourse that is closer to the way we perceive and relate everyday experiences [Kintsch, 1980; Bruner, 1986; Nelson, 2003]. We understand ourselves and others partly in terms of overarching goals and the actions we take to pursue them.

Other work has built on these ideas to explore the amount of identification a reader has with the protagonist of a narrative (which allows the reader to adopt the protagonist's goals as his or her own virtual desires) [Albrecht *et al.*, 1995]. As most stories have multiple agents, often with cross purposes, the goals of different agents must be tracked separately [Magliano *et al.*, 2005]. Suh and Trabasso [1993] find that readers keep both subgoals and superordinate goals in working memory when answering questions about a story, but make only online inferences about the most pressing subgoal during reading, suggesting that they maintain a "hierarchy of goals" in which one must be accomplished in order to achieve the next.

As we have alluded, another consensus view among psychologists is that a basic function of discourse comprehension is to find the **relationships between actions and goals**. Readers tend to want to view each agent's actions as being intended in pursuit of one goal or another, to the point where we infer goals when none have been made explicit. Poynor and Morris [2003], for instance, find evidence that goals are inferred at the time the information is presented, even if the information only implies the goal (rather than stating it outright). This suggests that "readers activate a representation of the protagonist and his or her goals early in the narrative, and that representation is strategically maintained throughout the narrative (or at least until that goals is met) as a vehicle for explaining actions." The import of goals and actions matters as well: Egidi and Gerrig [2006] show that the association between goals and actions is stronger when they are matched in terms of urgency and intensity, respectively.

The constructionist model has also taken on the notion of **goal outcome**, in which events signal that an agent has either succeeded or conclusively failed to achieve a goal. Researchers have found that this information is also an important trigger in the cognitive indexing of goals, subgoals and actions [Stein and Albro, 1996; Richards and Singer, 2001]. For example, Magliano and Radvanksy [2001] show that the success or failure of a goal affects its prominence in working memory as comprehension continues past the point where the outcome is revealed. Stein et al. [2000] describe the relationship between goals and "emotional understanding," in that children link happy memories to stories of goal success, and unhappy memories to stories of goal failure and of threats to valued goals. We empathize with the agents in stories and comprehend goal outcomes in terms of **affectual impact**, identifying agents that are positively impacted by goal success and agents that are negatively impacted by goal failure.

Each of these processes is specific to the reader, and different readers can construct different situation models from the same text. There is no one "true" cognitive representation for a given story. Gerrig and Egidi [2010], for example, point out that not all readers agree on which connections are to be made, and even the same reader can vary in terms of how much inference to bring to a text through periods of reflection. A shallow reader may use reflexive processes and automatic "rules of thumb" to determine the morality of an action or the relative importance of a goal, where a deeper mode of reading triggers alternative connections.

Nonetheless, these results give a basis in cognitive psychology for a set of "narrative primitives" which we use as the basis for our own representation of story logic: intentional agents, goals, subordinate and superordinate goals, outcomes of goals, goal-directed actions, and affectual impacts. A representation that reifies these must also include the basic elements in Bal's definition of *fabula*: **discrete time events** in the story-world and **causal relationships** between events. Taken together, this set of primitives do well against our three criteria for a representation. Just as experimental subjects can indicate which text spans indicate goals and which are intentional actions in a cognitive study, so too can annotators encode a corpus of stories with similar discourse relations in a computational study. A model based on these elements, aligned with our natural reading processes, would be both robust and accessible. Although most of these findings focus on the comprehension of textual discourse in particular, these primitives are invoked in the situation model regardless of the input modality.

In the remaining part of this section, we will consider three descriptive representations for narrative discourse, and consider the advantages and drawbacks of each.[3] In particular, we will examine the way that each system arranges all or some these narrative primitives with a set of useful discourse relations. The three models are:

1. The **GRTN (causal network) model**, championed by Trabasso as a model of cognitive story understanding,

2. Linguistically-rooted **story grammars**, in particular the grammar proposed by Mandler and Johnson, and

3. **Plot units**, an influential model that originated in Lehnert's work in artificial intelligence.

We will continue to use "The Wily Lion" fable as a common point of comparison in considering these three models and setting the scene for our own contribution.

Recursive Transition (Causal) Networks

The question of how these narrative primitives relate to one another is the subject of longstanding debate. There is a predominant view that a conceptual network (a "connectionist"

[3] We exclude from our survey prior work that uses these primitives to describe aspects of narrative logic, but does not provide a set of discourse relations we might apply to the task at hand. For example, van Dijk's application of the philosophy of action to the theory of narrative [van Dijk, 1976] considers the logical relationships between actions, intentions, outcomes, and the discourse in which they are expressed. While many of these insights intersect with those of these three models and our own model, we are only describing in detail those approaches which feature an accessible procedure for discourse annotation and analysis.

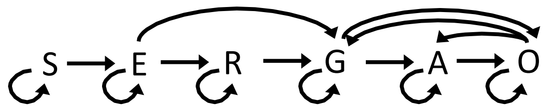


Figure 3.1: The General Recursive Transition Network, redrawn from van den Broek [1988].

model) is constructed in the situation model, but different suggestions for graph topologies have emerged (e.g., [Bearman and Stovel, 2000]). Two suggestions to consider in particular are those of Trabasso and Graesser along with their colleagues.

The "recursive transition" network theory of comprehension proposed by Trabasso [Trabasso and van den Broek, 1985; Trabasso and Sperry, 1985; van den Broek, 1988] organizes the text around the principle of causality. The situation model synthesizes propositions found in the textbase with the reader's world knowledge to arrive at a graph model in which nodes represent story fragments and edges indicate causation. The model assumes that a traditional story features a protagonist who encounters a problem and goes about a strategy for overcoming it.

The story fragments are separated into six functional categories: Settings (S), which establish the protagonist in time and space; Initiating Events (E) which result in Internal Reactions (R), which in turn cause the protagonist to have Goals (G); Actions (A) which are motivated by goals; and the Outcomes (O) of goals. A later articulation of the model [van den Broek, 1988; Trabasso and Wiley, 2005] further distinguishes between successful outcomes (SO) and failed outcomes (FO).

The reader assigns a causal link between any two nodes if there is "necessity in the circumstances," in that the reader's world knowledge tells her that one statement is a necessary consequence of the other. A is necessary for B if it is the case that had A not occurred, B would not have occurred. (Since B must temporally follow A to fit this definition, causal networks are also timelines.) Causal links take on slightly different meanings depending on the classes of the nodes they connect (only a "motivation" arc connects a Goal with an Action). Not all nodes can connect to all other nodes—the General Recursive Transition Network (shown in Figure 3.1) indicates which adjacencies are legal.

The GRTN model also supports goal hierarchies through *goal embedding*, which separates

Span	Category	Content
1	S1.1	A Lion watched a fat Bull
2	E1.1	feeding in a meadow,
3	R1.1	and his mouth watered when he thought of the royal feast he would make,
4	G1.1	but he did not dare to attack him, for he was afraid of his sharp horns. Hunger, however, presently compelled him to do something:
5	G2.1	and as the use of force did not promise success, he determined to resort to artifice.
6	A2.1	Going up to the Bull in friendly fashion, he said to him, "I cannot help saying how much I admire your magnificent figure. What a fine head! What powerful shoulders and thighs! But, my dear friend, what in the world makes you wear those ugly horns? You must find them as awkward as they are unsightly. Believe me, you would do much better without them."
7	O2.1	The Bull was foolish enough to be persuaded by this flattery to have his horns cut off;
8	A1.1	and, having now lost his only means of defense,
9	O1.1	fell an easy prey to the Lion

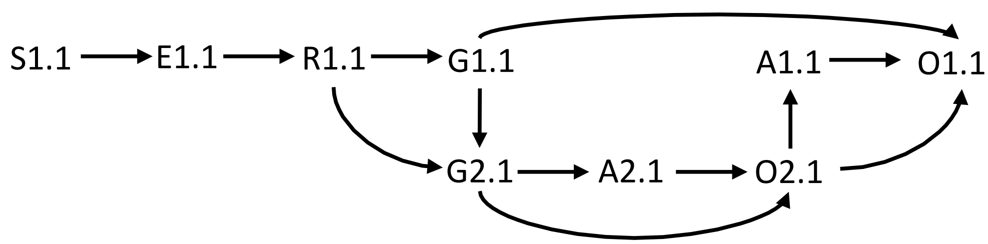


Figure 3.2: Outline of "The Wily Lion" in a causal-network representation.

the network into a tree of connected causal chains. If one attempt at a goal leads to a failed outcome, for example, the outcome can motivate a subgoal which is understood to be a strategy for achieving the main goal. When these networks are visualized graphically, every node is given a subscript (i,j) where i is the level of embedding and j is the node's position in the order of nodes of its type at its level of embedding. For instance, the second action in the causal chain of a subgoal would be A(2,2).

Figure 3.2 shows a GRTN representation of "The Wily Lion". First, the table separates the story into spans and assigns a node of a certain category to each span. Below the table is the graphical layout of the network, including causal connections and goal embedding. (The main goal is on the top row; the lion's subgoal to trick the bull is on the bottom row.) The model brings out many salient aspects of the story using an attractively small number of primitives—six node types and one arc type. It shows that each statement is connected to the main arc of the story, and suggests that some aspects are especially salient to the its structure: those in the critical path from one end of the story to the other, so that they are unavoidable in any trace of the network from S(1,1) to O(1,1). The strong cohesive structure of the GRTN has been shown to predict the accessibility of causally influential statements in memory [Trabasso and van den Broek, 1985; Trabasso and Stein, 1997]. The model is also robust and formal enough to be the basis for a machine implementation. A simulated version of the process of memory retrieval, using GRTNs, can reproduce the results humans give to recall and "think-aloud" tests on simple stories [Langston and Trabasso, 1999; Trabasso and Wiley, 2005].

There are, though, drawbacks to this type of causal network for the purposes of finding similarities and analogies between stories. For one, the topology describes the point of view of a single protagonist. Many stories find thematic meaning in the competing objectives of multiple agents. For the fable example, the model captures that both "The Wily Lion" and "The Fox and the Crow" involve a goal that begets a subgoal, but the goals of the "prey," manipulated as they are by their predators, are not included. More generally, causal networks offer a good model of structural importance within a single story but a vague model of the content similarities between stories, even when the structures are similar.

Other psychologists have extended the expressive range of the conceptual graph to address these shortcomings. Graesser and his colleagues [Graesser and Clark, 1985; Graesser *et al.*, 2001] use nine types of arcs instead of one, including Reason, Manner, Consequence and Implies. They also introduce generic knowledge structures (GKSs), which represent aspects of world knowledge, and show how they integrate into the nodes in the graph. Their schemata captures more implied information than Trabasso's, which is better described as an organization of the textbase around purely causal relationships. We take inspiration from the ability of this network to encode implied information only when it is thematically relevant (as opposed to never, in the case of GRTNs, or as often as possible, as in semantic understanding systems).

Like Trabasso's graph, Graesser and Clark's is designed to model human processes for recall, summarization and question-answering, using the constructionist theory of comprehension. We know of no attempt to apply the model to the process of finding analogical connections between different stories, but it has been a fertile testbed for modeling the way inference and activation occur during question-answering. The QUEST process [Graesser *et al.*, 1991b] is able to traverse the conceptual graph to find the *best* answers to "why" a particular action occurs. (This is often a difficult question to answer, as each action could potentially be explained by any of its causal or motivational antecedents). The graph topology is attractive in its balance of formality and robustness, in that the arcs are carefully selected to cover many stories (those that are totally connected by causal relationships), without being general enough to lump disparate stories into identical graphs. The number of nodes and arcs scales up or down with the number of goals, the complexity of the agents' plans, and the events that they set in motion. The model, though, is not as accessible as Trabasso's. Graphs are constructed from stories with a complex verbal protocol in which the researcher questions the subjects, a method which does not scale easily. Still, the QUEST model has found a recent revival in the domain of collaborative co-authoring, where an assistant is able to procedurally ask "why does this happen?" to help the user write a coherent narrative [Riedl *et al.*, 2008].

3.2.2 Discourse and Literary Theory

The linguistic perspective on discourse intersects with that of cognitive psychology. Both aim to find models that connect spans of text by meaningful, functional relations. Just as a syntactic model shows the relations between words that bind the sentence into a functional whole, discourse relations provide cohesion between phrases and sentences to describe the "point" of the discourse. Various types of relations have been proposed to provide coherence to a discourse. Referential relations, for instance, connect multiple mentions of the same entity as they occur throughout a discourse. Coreference and pronoun resolution, the processes that assign referential links, aid discourse comprehension in many languages by connecting clauses and phrases by the entities that are repeatedly mentioned [Grosz *et al.*, 1995; Grishman *et al.*, 2005].

Other discourse relations that have been proposed deal with the way entire clauses and sentences relate to their neighbors. Of particular interest is the manner in which utterances either introduce, expand upon, or break from a topic [McKeown, 1985; Hobbs, 1985; Polanyi and Scha, 1984; Hirschberg and Nakatani, 1996]. Another set of proposed relations describes the structure of implied *intentions* (what the speaker attempts to accomplish with each utterance) [Grosz and Sidner, 1986; Rambow, 1993]. In this subsection, we focus on those sets of discourse relations which have been proposed to deal with narrative or its close relatives. First, we describe *story grammars* in the context of other hierarchical models of structure. Then, we summarize other linguistic theories of narrative discourse, and transition to structuralism and other relevant ideas from literary theory.

Hierarchical Models

Expository discourse is the focus of hierarchical models such as Rhetorical Structure Theory (RST) [Mann and Thompson, 1988] and the Penn Discourse Treebank [Prasad *et al.*, 2008]. RST defines a set of relations between text spans. Some of the relations overlap with the narrative primitives we established from the cognitive literature (*Volitional Cause*, *Purpose*, *Motivation*, *Sequence*). Other relations are geared more toward argumentative discourse (*Concession*, *Otherwise*). Each span can subsume sub-spans, resulting in a tree-structured description; Mann and Thompson claim it is "typical, but not universal, for texts to be hierarchically structured and functionally organized."

Both models have appealing advantages. They strike the balance between generality and formality necessary to lend themselves to large-scale corpus annotation [Carlson *et al.*, 2003; Prasad *et al.*, 2008], corpus-based studies of text organization [Louis and Nenkova, 2010] and automatic relation extraction from surface text [Marcu, 1997; Lin *et al.*, 2010]. However, there are also significant drawbacks. Neither model aligns well with our set of ideal narrative primitives. The relations are not oriented around agents and are not well suited for hypothetical actions (plans and goals). Expository texts, in contrast, are topic-focused, with a model of cohesion that revolves more around the logical chaining of claims and arguments [McCarthy *et al.*, 2006; Berman and Nir-Sagiv, 2007].

That is not to say that RST-like models have never been proposed for the narrative mode. For several years in the 1970s and early 1980s, a flurry of work attempted to find a context-free grammar that described the structure of a story [Prince, 1973; Rumelhart, 1975;

FABLE	→ STORY AND MORAL
STORY	→ SETTING AND EVENT_STRUCTURE
SETTING	→ {STATE* (AND EVENT*) or EVENT*}
STATE*	→ STATE ((AND STATE)+)
EVENT*	→ EVENT (({AND or THEN or CAUSE} EVENT)+) ((AND STATE)+)
EVENT_STRUCTURE	→ EPISODE ((THEN EPISODE)+)
EPISODE	→ BEGINNING CAUSE DEVELOPMENT CAUSE ENDING
BEGINNING	→ {EVENT* or EPISODE}
DEVELOPMENT	→ {SIMPLE_REACTION CAUSE ACTION or COMPLEX_REACTION CAUSE GOAL_PATH}
SIMPLE_REACTION	→ INTERNAL_EVENT ((CAUSE INTERNAL EVENT)+)
ACTION	→ EVENT
COMPLEX_REACTION	→ SIMPLE_REACTION CAUSE GOAL
GOAL	→ INTERNAL_STATE
GOAL_PATH	→ {ATTEMPT CAUSE OUTCOME or GOAL_PATH (CAUSE GOAL_PATH)+}
ATTEMPT	→ EVENT*
OUTCOME	→ {EVENT* or EPISODE}
ENDING	→ {EVENT* (AND EMPHASIS) or EMPHASIS or EPISODE}
EMPHASIS	→ STATE

Table 3.2: Mandler and Johnson's [1977] story grammar (reformatted).

Mandler and Johnson, 1977]. Inspired by Chomsky, these grammars take the form of rewrite rules that begin with a single "Story" symbol and descend to various types of "atoms" which appear in the form of words and clauses; the act of "parsing" a story assigns each clause to a functional aspect of one of the rules. In all story grammars, the rules combine structural and goal-centric discourse relations—an episode consists of a beginning, a development and an ending; a goal path consists of an attempt and an outcome; and so on. Prince [1973] shows how to parse "Little Red Riding Hood" into a hierarchical structure of nested events and mental states that are joined by rules involving time and intention. His definition of a "minimal story," the smallest tale accepted by his grammar, echoes the Forster distinction we saw earlier: Any story must have at least two events which not only occur at different times but are also causally related.

Mandler and Johnson's [1977] grammar, reproduced in Table 3.2, is typical. Although Mandler and Johnson presented their work in the context of cognitive psychology, showing the connection between the grammar representation and the results of experiments that

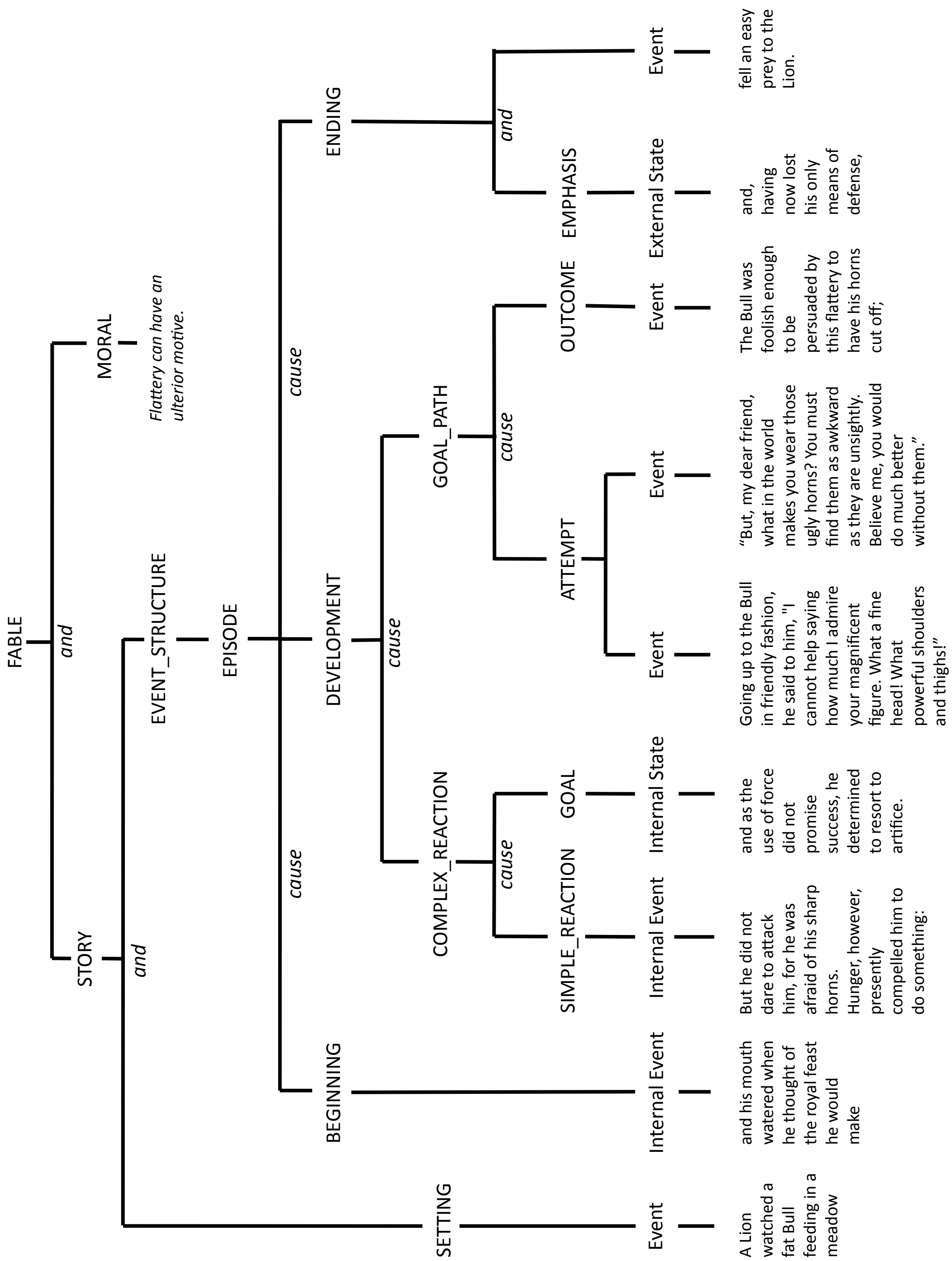


Figure 3.3: Story-grammar parse of "The Wily Lion".

measured reader recall, it has influenced linguistic models of discourse such as RST. The grammar's rewrite rules describe the *fabula* of story, including the temporal connections (`AND` for simultaneous events, `THEN` for sequential), causal connections (`CAUSE`), the internal goals of characters, and goal-directed actions. The terminals are `EVENTS` and `STATES` (the latter being a condition of the world), where both can either be internal (as in thoughts, plans, and perceptions) or external (actions and happenings). A simple, goal-directed story like "The Wily Lion" can be parsed by hand to conform to the grammar (see Figure 3.3)—in fact, multiple parses are possible depending on one's interpretation of causality and other factors that are not made explicit in the text. Note that the terminals in the diagram are sentences from the text, where the grammar calls for more textbase-level terminals (propositions).

Grammars are good for capturing both the local and global coherence of properly structured plots. The quantization of clauses by time states (using the `AND` and `THEN` rewrite rules), causal relationships, and associations between goals and goal-driven actions are laid out more precisely than in the non-hierarchical models we saw earlier. Analogical similarities can be found across the corpus by looking for rules and compositions of rules that recur across the corpus. Moreover, it makes intuitive sense that there would be a story-level equivalent for the context-free grammars suggested for syntax.

There are, though, both theoretical and practical problems with using grammars for stories. A context-free grammar that separates stories from non-stories must, by definition, accept every story and reject every non-story. The designer of the grammar must commit to a precise description of what a story *is* and how it must manifest as surface text. Mandler and Johnson leave some wiggle room—they mention but do not formally describe "transformation rules" that can affect the transition from *fabula* to discourse, with certain atoms rearranged or deleted—but even at a *fabula* level, the method is tilted far toward formality at the expense of robustness and coverage. When challenged about the narrow coverage of story grammars [Black and Wilensky, 1979], both Mandler and Johnson [1980] and fellow grammarian Rumelhart [1980] replied that the domain expressed by their grammars was never meant to cover *all* stories—just those in the oral tradition, or those with a "recursive" structure, respectively. The question remains, though: Can any context-free grammar for stories achieve a wide coverage?

Ryan [1991, 201] argues (before giving her own grammar parse for "The Fox and the Crow") that the idea of a story grammar is fundamentally suspect because there is a seemingly unbounded number of possible story actions that can serve as terminals. The number of lexemes that can serve as terminals in a syntactic model (i.e., the number of words in the vocabulary) is small compared to the number of actions which we might enumerate as possible in a story-world. More importantly, a fundamental aspect to grammars is that elements in one branch of the parse tree can not "cross over" to relate to neighbors in other branches. Individual terminals can also only belong to a single rewrite rule. Unfortunately, both situations regularly happen in even simple stories, when one interprets them to include the inner worlds of character agency. Characters can revisit and alter plans once they have failed. Two characters may have plans at cross purposes who alter each other's executions in turns. An undesirable state (such as poverty) can be activated, deactivated and activated again, though we would not suggest having "in and out of poverty" be a high-level rule with its own structural properties. Well-told stories are contrapuntal documents, as seen in popular storytelling advice that writers have each scene fulfill at least two functions at once (a text and a thematic subtext) [McKee, 1997]. Some of this missing information is key to the analogical connections between stories. Like Trabasso's causal network, which did not itself impose a tree structure on most of its topology, story grammars struggle to see a text from more than one perspective at once. The common experiences of the two victims, the bull and the crow, are enablers for their predators' plans; this is unfortunate, since the *point* of these fables (in the sense of Wilensky's contemporary *story points* model [Wilensky, 1983]) is to have the reader identify with the prey. As cautionary tales, they both warn against taking flattery to heart and taking risks in the name of self-aggrandizement.

Similar concerns about the expressive power of trees were later brought up about RST, which, though not strictly a grammar, also provides that only one type of relation can bind two utterances together. As Moore and Pollack point out [1992], in many cases two utterances can be related in multiple ways, even in expository texts. Wolf and Gibson [2005] similarly posit that mandating a tree structure can forego important discourse relations that cross over. Lee et al. [2008], in discussing the shared arguments which are allowed in the Penn Discourse Treebank, ask whether trees are used in discourse simply based on the historical precedent of trees being used widely in syntax. For the case of narrative, we believe that trees are not the right discourse model for finding analogical connections between stories.

Though story grammars in particular fell out of favor, new attempts at this type of model still come up from time to time [Lang, 1999].

Other Linguistic and Narratological Models

Grammars were neither the first nor the last linguistic proposal for modeling narrative discourse (though they were the most controversial). Of interest are other efforts which take a more corpus-based approach to finding a set of useful discourse relations for characterizing group-wide norms by their thematic content.

When linguists Labov and Waletzky [1967] studied a corpus of urban oral storytelling, they found six recurring units: an abstract, which sets the scene; an orientation, which sets the initial situation or activity; a complicating action (what happened?), an evaluation (so what?), resolution, and a coda. Bell [1999] removes some of Labov and Waletzky's units and adds others to model news articles as narratives, including a premise, a main event, the background that precedes the main event, and so on. Polanyi [1989] considers oral storytelling as a complex social exercise in which the storyteller and her peers take turns and exhibit mutual concern over their images; these pragmatic factors directly influence a story's content, such as when a controversial position is clarified or qualified.

The group norms do not necessarily have to be about the relations that join together clauses. The content of the textbase can also be the subject of a data-driven study. We can ask not only, "how are stories structurally similar?" but also, "what are stories generally *about*?" A typical approach to describing this code is to examine an entire corpus and align it by analogically similar content. These normative areas of overlap can be actions that recur in each story, character stereotypes and other tropes. Then, each story is individually examined in terms of how it follows or deviates from the group norms. One recent annotation scheme for analyzing sets of narratives allows semantic units to "emerge" from clusters of text which convey the same basic content, regardless of the degree of lexical overlap

[Passonneau *et al.*, 2007]. Another study uses this approach in order to build a system that classifies the completeness of children's retellings of simple stories [Halpin *et al.*, 2004].

Certainly the most famous example of this technique is Vladimir Propp's structuralist study of several hundred Russian folk-tales [Propp, 1969]. Propp identified 31 *functions* that recur throughout each story, where a function is "understood as an act of a character, defined from the point of view of its significance for the course of the action." He used his own measure of what we might call "semantic distance" to determine when two events belong in the same function. Each function is given an abstract definition that encompasses all of the scenes it counts as members. Some functions are more abstract than others. The function *The hero is pursued*, for example, is drawn from scenes that span a range of scenarios from the pursuer flying after the hero to the pursuer gnawing at a tree in which the hero takes refuge. Other functions include *The villain causes harm or injury to a member of a family* and *The hero acquires the use of a magical agent.*

Unfortunately, Propp did not give clear guidelines as to *how* he identified each function. That is, he did not tell us exactly how analogous two scenes must be to form a function, so that we can repeat his technique algorithmically. It is also worth noting that not every story supplied an example of every function; some stories contributed to fewer than half. However, Propp's functions do hold the striking property of sequential consistency, in that they tend to appear in the same order in the corpus (albeit with many exceptions). This allowed Propp to draw up a simple grammar of the Russian folk-tale:

$$S \rightarrow ABC{\uparrow}DEFG\frac{HJIK{\downarrow}Pr - Rs^0L}{LMJNK{\downarrow}Pr - Rs}QExTUW*$$

The letters and up/down arrows represent functions. The fraction indicates a branch point, so that each story follows either the top or bottom path. The rigid ordering of Propp's functions lends itself to computational treatment, and indeed Propp has widely influenced the field of narrative generation; several studies have even adapted his scheme outright to make story-writing programs [Díaz-Agudo *et al.*, 2004; Gervás *et al.*, 2005]. For the purposes of our task, we find Propp's method of identifying group norms and their discourse relations appealing, but Propp's functions are not, in and of themselves, a representation we can apply to Aesop's fables or other genres outside of Russian folk-tales. We are more interested in replicating the process of identifying group norms that are abstract enough to have wide coverage, yet precise enough to reveal useful insights. Indeed, this is a summary of our larger objective: Our present search for discourse relations that capture thematic content is a search for a representation that enables a systematic search for corpus-wide group norms that describe interesting narratological tropes. Indeed, Propp's functions are connected only by temporal sequence. We strive to find group norms centered around sets of relations that tie similar events together, including sequence as well as such factors as motivation and enablement that Propp did not model.

Propp was also an inspiration to the mid-century proponents of a structuralist literary theory that sought to find universal system of symbols and relations that organize sentences into coherent narratives. Levi-Strauss [1968] went beyond the story or the function as a unit of analysis, and identified "mythemes" as atomic, irreducible components found across many of the myths of a culture. Like notes forming chords in a musical stave, the mythemes can be joined together ("bundled") when they co-occur in a story, particularly in binary opposition to one another; these bundles are depicted as a dimension of discourse perpendicular to the temporal flow of time. Todorov [1969] found what he called a "narrative syntax" underlying several stories in the *Decameron*, using the grammatical structure of language as an analogy. The clauses are connected by temporal ordering, entailment and spatial relationships, similar to the causal networks proposed in psychology. To Barthes [1975] as well, discourse is a large "sentence," with a grammar that links units together into a combinative scheme.

For each of these approaches, character is little more than an enabler of action. The mental states, explicit or implicit, that motivate agentive characters to plan and act—the fears or hopes for possible futures—are not a part of the equation. Some structuralist thinkers took a character-centric approach, though. Bremond [1980] saw plot as a "network of possibilities" that underlies each action, with characters thinking several moves ahead in a branching model not unlike game theory. In a Propp-style study of French folk-tales [Bremond, 1970], he finds a pattern in which characters cycle between positive and negative affect states.

Structuralism fell out of favor with Derrida and other proponents of deconstruction. Barthes himself turned away from the search for a universal symbolic underpinning to narrative in his later works, advocating instead for a reader-oriented approach to narrative meaning [Barthes, 1978]. Although the symbols we reviewed in our discussion of cognitive psychology are also subjective, in that each reader builds his or her own situation model, deconstructionists are far more cagey about proposing universal symbols and relations that can apply to many texts. In *S/Z* [Barthes, 1974], Barthes proposes a set of "codes" that function as threads in an interwoven search for meaning on the reader's part. The *hermeneutic code*, for instance, gives a sense of suspense to the reader by withholding answers to important questions or delaying an expected event. The five codes apprehend the text as a plurality, a "galaxy of signifiers, not a structure of signifieds," that "has no beginning; it is reversible; we gain access to it by several entrances, none of which can be authoritatively declared to be the main one." The reader is an active participant in constructing threads of narrative meaning, rather than a passive receiver of a single authoritative model of the text.

Although our approach is structuralist in character, we do not make a claim that a structuralist approach to literature is inherently superior to such a receiver-oriented approach for the task of designing a formal representation of story meaning. Rather, we pragmatically find that in the absence of a wide-coverage understanding and inference engine that can read a text and simulate such processes of reading as anticipating future actions and detecting questions needing answers, we cannot systematically compare stories by beginning with a process like the hermeneutic code and working backwards to the textual facets that trigger a response. We must instead start with connections we can find at the discourse, textbase and/or *fabula* level. At that point, if we are studying affect, we can move forward to predicting a normative reader response to a story based on responses to similar stories. However, the lesson that we can draw from post-structuralism is that subjective differences between receivers, as well as plural readings by a single receiver, cannot be marginalized in a symbolic model of narrative discourse. As we saw in Chapter 1, stories are told to communicate information *and* to evoke an affectual response from the receiver. We re-read books and re-watch films not to be reminded of the plot, but to repeat the experience of being in the narrative world. Unlike most annotation tasks, where inter-annotator agreement is unequivocally bad, a narrative interpretation task should elicit, embrace and study the differences between subjective encodings of the same text.

Since the decline of structuralism, literary theorists who claim that there are absolute universals of any kind inherent across the narrative mode of discourse have been few and far between. Even the notion of such a goal is not often conceded by academic theorists (which is why we have turned to psychology for the foundations of a new representation). One recent trend, though, is to apply the **theory of mind** to literature and the semiotics of narrative. This approach takes as its unit of analysis the presence of a conscious entity in the story, capable of identity, self, subjectivity and experientiality [Palmer, 2007]. The receiver must connect to conscious agents within the narrative who are perceiving and transmitting the story-world information (roughly analogous to the narrating agents between the layers of Bal's *fabula*-story distinction). Reading a novel is akin to following the thought processes of these agents. Palmer [2010] applies this mode of analysis to Dickens's *Little Dorrit*, which we included in our study in Chapter 2, and finds a social interplay among the characters as they signal, withhold, and yield information to one another. They derive power and status from this *intermental* (collective) thought, as opposed to individual or private thought.

The foundation for this reading is *attribution theory*, an area of cognitive psychology which (among other purposes) describes how observers attribute states of mind to other conscious entities—not only goals and plans, as we reviewed, but emotions and the entities' own attributions regarding the minds of still other entities [Jones and Nisbett, 1972]. In other words, what is important is not just what is known, and what is known to be unknown, but what is known to each character, what each character knows about what is known to each other character, and so on, in what we will represent as a system of nested **agency frames**. (The outermost or "bottom" frame represents what is known to an agent; the second frame from the bottom represents what an agent knows about what other agents know, and so on. All stacks rest on an objective foundation that we will call "ground truth.") This model is similar to that of the *private state frame*, which Wiebe et al. [2005] devised as a fine-grained annotation scheme to assist in the automatic identification and extraction of opinions, emotions, and sentiments in text. The ability to "mind-read" is also known to be an important milestone in development; autistic children have abnormal inabilities to see the world through the eyes of others [Baron-Cohen, 1989].

As attribution is key to understanding stories as simple as fairy tales, nested agency frames should be expressible in our new narrative representation. In "Little Red Riding Hood", we know that the wolf is masquerading as the girl's grandmother, and we know that the wolf knows that fact, and we know that Little Red Riding Hood does *not* know that her grandmother is really the wolf, and we know that the wolf's goal is to maintain such a belief on the part of the girl [Chen and Fahlman, 2008]. In some tellings, we suspect that the girl is transitioning to a new state of understanding during the dialogue in which she expresses continuous surprise at her supposed grandmother's wolf-like features ("What big, sharp teeth you have!"). In both "The Wily Lion" and "The Fox and the Crow", the predator's plan is to instill a belief in his prey that the predator thinks the prey is almost appealing in some way. In both cases, the deception hinges on the difference between the reality of the predator's thoughts, known to the receiver and to the predator, and the feigned reality purported to the prey. To understand these stories is to constantly separate not only what actions happen before other actions, but also what mental states are occurring and who is associated with them. As Zunshine [2006] argues in her theory-of-mind reading of Woolf's *Mrs. Dalloway* and other modernist works of literature, the narrator's "engagement of our metarepresentational capacity"—its method of leaving clues with which we can read the minds of conscious characters—is a universal that gives the novel its currently familiar shape.

3.2.3 Implemented Understanding: Scripts, Plans and Plot Units

Those who have implemented story understanding systems have typically come from the artificial intelligence tradition, though they sometimes work in concert with cognitive psychologists to arrive at a cross-domain theory of the way the mind represents narratives (e.g., [Charniak, 1972; Schank and Abelson, 1977]). It was one of the first problems considered by what we now call natural language processing: The phrase "story understanding" has often entailed a system that, given a textual story, can answer yes/no and "why" questions that demonstrate inference, retell the story, or generate a summary by choosing key aspects and reformulating them into new sentences [Lehnert *et al.*, 1983;

Reeves, 1991]. Unfortunately, as we mentioned earlier, such a system needs a large and laborious modeling of world knowledge in order to interpret natural language input to the point of inferring causes, consequences, ethics, and key points (that is, comprehensive semantics). The previously discussed work in psychology has informed us that human story comprehension is a fusion between the propositions in the textbase and the world knowledge of the receiver—but models of comprehension such as causal networks only record *that* an inference is made by a human reader, without describing *how*. To build a story understanding system, those methods of inference must be concretized, a problem that has yet to be solved at scale [Lehnert, 1994].

The early researchers attempted to "start small," restricting their input to a very limited set of stories that each invoke only a slice of domain knowledge, and then broaden coverage over time. The broadening never quite arrived, though, and when the tide shifted to corpus-based methods in the 1990s, narrative as a discourse mode of study fell by the wayside. While a few long-term projects are taking steps toward deep understanding with broad coverage [Mueller, 2006; Zarri, 2010], work in the computational modeling of narratives has been performed periodically in different areas—sometimes from a discourse context, sometimes from an agent-centric context, sometimes from a generation context, sometimes from a ludic (interactive gameplay) context. Each context carries with it a particular set of constraints on the narrative representation. Story generation and interactive narrative, for instance, require *prescriptive* models, with rules for story structure and character behavior defined precisely enough to prevent nonsensical output. A *descriptive* model such as ours trades off precision in order to be robust enough for a data-driven (corpus-based) analysis. In this section, we review some of the story representations from both categories that have been devised to build narrative-savvy systems.

The work in understanding at Yale in the 1970s [Schank and Riesbeck, 1981] saw narrative as a way of controlling the combinatorial "explosion" of inference that occurs when reading discourse. This is because the scope of possible logical interpretations of two sentences that have a functional connection in a model of discourse is smaller than the scope of possible interpretations of both sentences as seen individually. "Narrative," then, is a mode of discourse where agents have goals and pursue those same goals in a logically consistent pattern, and a library of narrative facets can guide a system as it interprets the meaning behind a textual story such as a news article.

The common ground between artificial intelligence and the psychology of narrative comprehension was seen early on in work describing the inferences involved in recognizing the goals and plans of agents based on their actions [Schmidt *et al.*, 1978]. The efforts to formalize a system of goals in commonsense psychology continue today, in particular by Hobbs and Gordon [2005; 2010]. They describe logical axioms for the process of identifying goals, developing plans for achieving them, monitoring the outcome of those plans, and modifying those plans as needed. For example, if an agent wants *e1* and believes *e2* causes *e1*, that desire will cause the agent to also want *e2* (as a subgoal). This recalls the models of Trabasso and Graesser, but adds a notion of *importance* in which agents give their goals a partial ordering by preference. A rational agent, then, is one that first pursues those goals that are most important. Many stories hinge on this notion of weighing one goal against another, the essence of the *dilemma*. A protagonist must decide whether to go for the job promotion or fulfill a family obligation, or whether the pursuit of love is worth the alienation of one's tribe.

The *script* was introduced by Schank and Abelson [Schank and Abelson, 1977; Cullingford, 1981] as a frame-like structure [Minsky, 1975] that stores procedural knowledge about some process: what happens, and in what order. In the canonical example, the `RESTAURANT` script gives a sequence of "scenes" (entering, ordering, eating, exiting), each of which containing first-order actions (customer enters restaurant, customer sees tables, customer sits). In the SAM (Script Applier Mechanism) system, when the restaurant script is activated by an action such as "John entered the restaurant," the script triggers expectations for what may happen next. Though scenes are an attractive organizing principle for narratives, such hard-coded scripts fell out of favor due to their rigidity. Contemporary analogies to scripts, such as narrative event chains [Chambers and Jurafsky, 2008a] and certain aspects of the FrameNet project [Johnson and Fillmore, 2000], use corpus-driven statistical models to generate or evaluate scripts.

The *plan* was devised by Schank's students as a knowledge structure that was focused around agents rather than episodes. Plan formalisms describe the motivations and ultimate goals for actions that might appear in the text. They can be combined and modified more flexibly than scripts. Wilensky's PAM (Plan Applier Mechanism) system [Wilensky, 1978a] understands textual stories by inferring the intentions of the story's characters from their actions. Unlike scripts or Trabasso's causal networks, PAM could represent opposing goals as well as chains of *why* questions [Wilensky, 1978b]. It might infer that John asked Mary where he could find a restaurant because John was hungry, but that John's hunger does not need to be explained, because people normally need food. PAM was able to stitch together stories based on chains of intersecting plans, and had a library of both plan architectures and goal transformations. For instance, it understood goal *subsumption* as form of long-term planning in which an action is designed to address a potentially recurring goal rather than one which is clear and present. From this principle, it further asserts that marriage is a plan that prevents recurring episodes of loneliness which may occur later in life. While PAM suffers many of the same issues with rigidity that SAM does, it introduces an attractive architecture for modeling the way in which plans from different agents intersect, whether in competition, cooperation, motivation, subsumption, or another relationship.

The formal treatment of goals, plans and beliefs evolved into an action control architecture called BDI (beliefs, desires and intentions) [Bratman, 1987; Rao and Georgeff, 1995; Busetta *et al.*, 2003; Konolige and Pollack, 1989]. BDI models not only what is known and unknown to a certain agent, but what the agent's goals are and what actions are possible for it to take that might eventually satisfy those goals. In one form, it defines formalisms for the theory-of-mind question of what agents know about the world and about what other agents know [Rapaport, 1986]. BDI has matured over the years to the point where it can approximate the process of rational decision-making and plan-making using limited evidence, emphasizing the subtleties of intention behind each action [Cohen and Levesque, 1990; Pollack, 1990]. Although BDI was not designed as a representation for narrative discourse, its agent-centric approach has inspired similar architectures that allow interactive narrative systems to have plausible, value-driven characters [Peinado *et al.*, 2008; Damiano and Lombardo, 2009]. Similarly, the *partial-order planning* architecture, sometimes combined with features of BDI, is widely used in narrative generation and interactive-narrative systems [Mateas and Stern, 2003; Riedl and Young, 2004; Mott and Lester, 2006;

Winegarden and Young, 2006; Barber and Kudenko, 2008]. Planning is also a useful approach to generating surface-level narrative in both textual discourse and visual media [Callaway and Lester, 2002; Jhala, 2004]. As was the case with PAM and SAM, these systems trade off robustness and expressibility in order to arrive at a level of formal precision sufficient for controlling actions and describing narratives. Their scope is limited to just those narratives consisting of actions and goals that have been modeled by hand (though some BDI systems include a high-level language to allow domain experts to extend the knowledge base as needed [Rao *et al.*, 1992]).

One recent project has aimed for a more abstract, yet computable representation of actions, mental states and agent behavior. SCONE [Chen and Fahlman, 2008] uses a semantic network representation in which nodes represent entities such as physical objects, types of actions, and actual actions that have occurred at some point in time. The system represents actions in a frame-slot representation [Minsky, 1975] but does not understand their consequences in the sense of Mueller's first-order representation [Mueller, 2004], in that SCONE cannot infer implied information or answer general questions about a story. What it does understand deeply is the epistemological component: what characters know, and what they know about what others know. Each action is situated in a *mental context* tied to the time in which the action occurred. The mental context is agent-specific. In their encoding of the Brothers Grimm's telling of "Little Red-Cap" (a.k.a. "Little Red Riding Hood"), the wolf's intention to eat the girl is different from the girl's belief in the wolf's intention, because the girl believes that the wolf is her grandmother. Their system can detect contradictions within mental states and answer simple questions about what characters know. However, it does not currently model goals and plans, so there is no way to understand the crucial point that the girl's belief in the wolf's identity is itself the wolf's intention and part of a larger plan. Our project is similar in approach to SCONE, but it provides a more expressive symbolic vocabulary. Other recent work has adopted the theory-of-mind approach to reading a text, with its emphasis on epistemic differences between agents, in order to model real-life narratives [Löwe and Pacuit, 2008; Löwe *et al.*, 2009; Nissan, 2008]; we see this as a promising approach.

Plot Units

We now conclude our tour of prior models of narrative discourse with a surprisingly versatile formalism called plot units, developed as part of a semantic story understanding system called BORIS [Lehnert *et al.*, 1983]. BORIS was capable of parsing, interpreting and answering questions about simple stories in a small knowledge domain. One of its functions was to summarize the key points of a narrative; to do this in a thematically aware manner, Lehnert [1981; 1984] devised a data structure that represents the affectual state of each character. As BORIS reads the story, it creates a linear map of temporally ordered affect states for each agent. There are three possible states:

- + (Positive Event): Events that please the agent in question
- – (Negative Event): Events that displease the agent in question
- **M** (Mental State): Mental state of the agent in question

The affect states are then connected by instances of four types of directed arcs:

- **Motivation (m)** always points to a mental state M, and from the state which caused M
- **Actualization (a)** always points from a mental state M, and to a + or – state intended by M
- **Termination (t)** points from one mental state or event to another that the first displaces (replacing one goal with another, or having the positive affect of a + displace the negative affect of a –)
- **Equivalence (e)** points from one mental state or event to an identical copy (useful for when two agents perceive the same event in different ways)

In all, there are 15 legal pairwise configurations that function as "lexemes" of plot, as seen from an agent-affect perspective. All arcs and nodes are instantiated in a *domain* associated with an agent. Some arcs may stay in the same domain (belonging to the same agent) or cross domains (describing a particular causal effect that one agent has on another). Each configuration corresponds to a thematically interesting narrative event of some kind. The notion of a "mixed blessing," for instance, has not been modeled by any previous

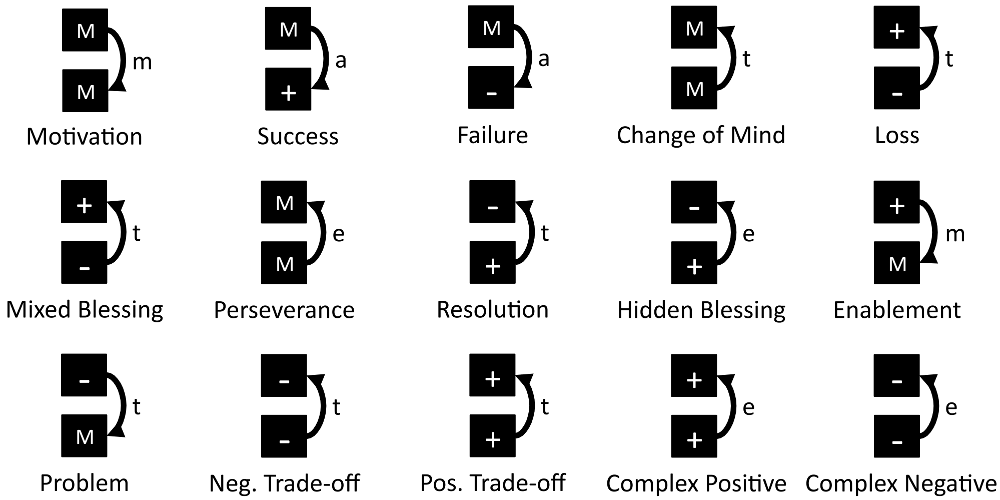


Figure 3.4: Simple plot units, redrawn from Lehnert [1981].

representation we have examined, yet is expressed with a single (e) arc from a negative state to a positive state in the same agent domain (indicating that an event is at once good and bad for the agent). Figure 3.4 depicts the 15 basic units along with their thematic interpretations.

The greatest virtue of Lehnert's model is that like Trabasso's causal arcs, simple plot units can be chained together to form compounded units of arbitrary length and complexity. The expressive range is more powerful than that of the GRTN, though, due to the separation of events and states into agent-oriented domains. One event can have multiple consequences to different agents, occurring in parallel from different perspectives. There no longer needs to be one central protagonist. Lehnert identified some 30 complex plot units describing exchanges between two characters such as the double-cross (where one agent requests an action and instills a mental state in another, only to have the latter agent trigger an event that helps himself and hurts the requester).

"The Wily Lion" can be reduced to a plot-unit representation by virtue of the fact that it hinges on a request that purports to have a positive affectual impact on both characters, but actually has a positive impact for the requester and a negative affect for the requestee. A plot-unit representation of this fable chains five simple plot units to achieve this effect. Figure 3.5 shows the plot-unit mapping in which the fox is *motivated* to eat, *fails*, considers

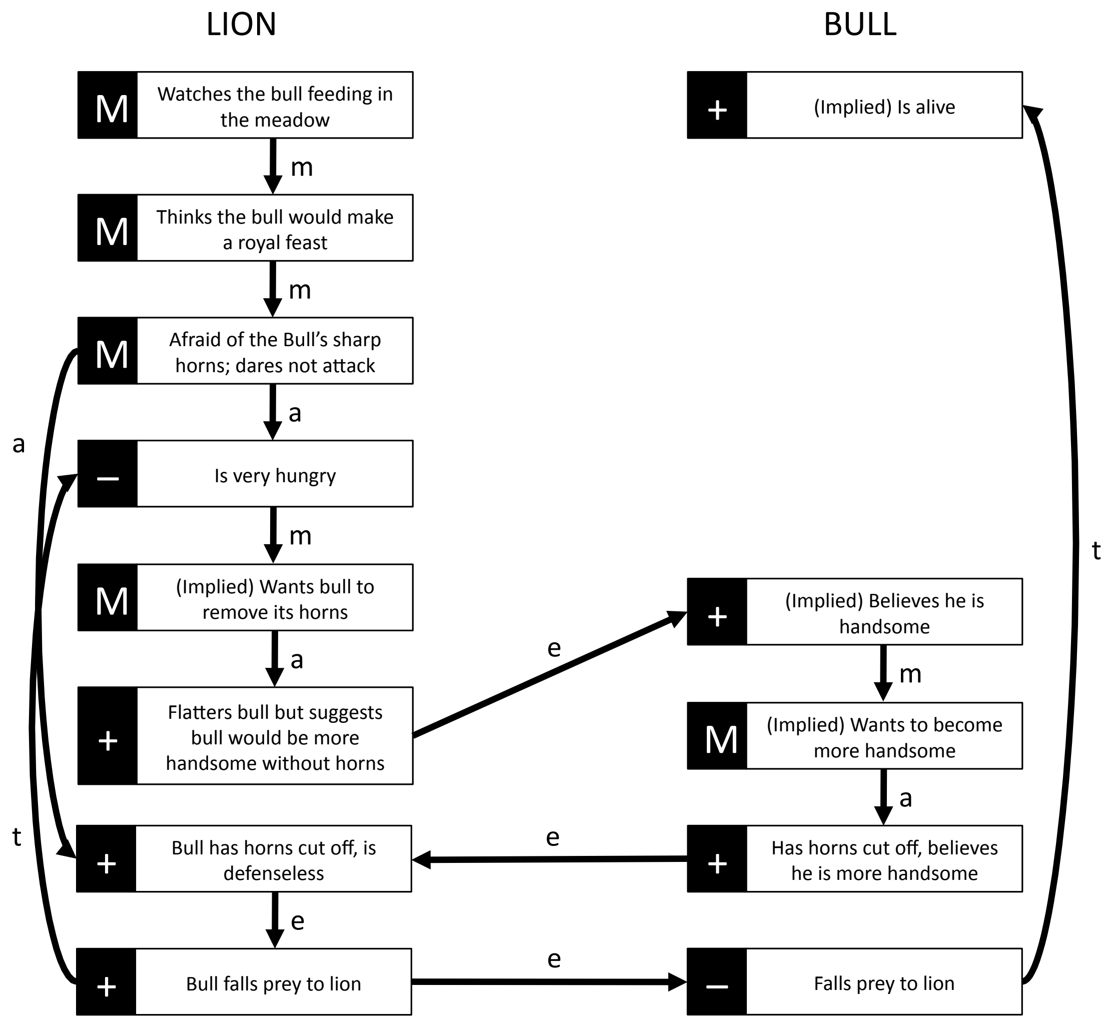


Figure 3.5: "The Wily Lion" in a plot-unit representation.

his *problem*, and has *success* in flattering the bull, who believes he is *enabled* to make himself more handsome. The bull *succeeds* in improving his image, but the same event allows the lion to *resolve* his hunger, which causes a grave *loss* on the bull's part but is a *success* to the lion.

Lehnert's model scores highly against our criteria. It is robust, in that it does not depend on a full semantic understanding of the story *per se* (although it was originally conceived in conjunction with such a system). The chainable nature of plot units makes them a more expressive formalism than a grammar or RST-style representation, allowing parallel causal chains, multiple agent perspectives, and so on. Plot units support implied information, rather than being a rearrangement of the textbase alone. The semantics of the node and edge types are well-chosen to be able to emulate subtle and complex narrative exchanges. Comparisons between stories can be made using complex plot units as a mediator (both "The Wily Lion" and "The Fox and the Crow" might satisfy a *betrayal* plot unit). Our representation strives to replicate each of these advantages.

The plot-unit model, though, does not capture other elements that are useful in a descriptive representation. There are very few time operators, for instance, and it is not obvious that every event in the story that is relevant to the plot must necessarily have a negative or a positive affect to an agent. Further, a ternary +/−/M system is somewhat coarse. + and − can refer to either events or states, and M does not address the epistemic question (what one agent knows or believes about another's knowledge frame). The distinction between hypothetical events (such as goal states) and actual events is unclear; it is impossible, for example, to indicate that a multi-step plan was completely abandoned by a character because another character brought about the ultimate goal on his own accord. The lack of hypotheticals also makes goal hierarchies awkward. In Figure 3.5, it is not quite right to say that "is very hungry" presents a *problem* by motivating the lion to want the bull to remove its horns; it is more of a restatement of the same problem that previously failed when the bull became fearful. These issues are partly due to scope, in that BORIS contained elements that worked orthogonally to plot units and handled aspects such as time. Our new representation, divorced from a semantic inference engine, can extend plot units along these lines.

Ryan [1991; 2007] has proposed a representation called a "recursive graph model" that inherits aspects of Lehnert's architecture (including an open graph structure and character-specific domains), but adds a much richer set of primitives for describing mental states. Each agent has a set of five distinct domains: K-worlds, which are epistemic (beliefs, projections and retrospections), O-worlds, which are private or social obligations, W-worlds, which are desires and fears, G-worlds, which are active goals, and P-worlds, which contain the plans through which characters seek to fulfill their active goals. Physical events, in a timeline, are separated from mental events, which are grouped by agent. To our knowledge, though, Ryan's model has yet to be implemented in an annotation interface or corpus collection project.

Plot units have also been influential in statistical approaches to understanding textual narrative. One recent effort takes a large step toward extracting plot units from unstructured text by classifying the affectual states implied by various clauses with respect to agents that appear as named entities [Goyal *et al.*, 2010a]. Nackoul [2010] develops a natural-language template for describing plot units as well as a system that uses these templates to search for plot units as they appear in narratives as diverse as *Macbeth* and legal case briefs. Similarly, Appling and Riedl [2009] return to Lehnert's original intention for plot units, summarization, with a system that uses conditional random fields to label affectual states, events and relational links as they appear in surface text.

3.2.4 Conclusion

The preceding literature review has featured symbolic models of narrative discourse from a variety of fields and intents. In aggregate, they present a set of tradeoffs: between formality and robustness, between an event-centric and agent-centric view of what a story *is*, between prescriptive and descriptive. The most promising aspects of each model have guided the design of our own contribution. In particular, we are motivated to include a system which can express goals, subgoals, plans, beliefs, attempts to achieve goals, and goal outcomes, as the studies from cognitive psychology strongly suggest that these are hard-wired into the human narrative instinct. In terms of structure, we are attracted to the notion of a small but highly recombinable lexicon of nodes and arcs, as we saw in Trabasso's and Lehnert's models. These are not only abstract and consistent from story to story, enabling contrastive studies; they are also accessible and can be tractably extracted from surface text by automatic taggers. The theory of mind offers a favorable template for modeling complex epistemic interactions in the context of separate agents; these interactions are often behind the thematic crux of a story. Finally, linguistic work along the lines of Propp, Polanyi, Labov and Waletzky and Passonneau tells us that we can find semantic similarities between stories in a corpus without committing to a complete semantic understanding of narrative *fabula*. In the next section, we describe Story Intention Graphs, which attempt to synthesize these insights in a new representation for story annotation, reasoning and comparison.

3.3 Story Intention Graphs

Narrative is an interplay between the minds of agents, the actions they take, the events which befall them, and the perception and transmission of that content in a communicative artifact. In this section, we propose a representation of a story that reifies these as nodes and arcs (relations) in a semantic network. We call the schemata itself the "Story Intention Graph" (SIG), and each instance of story annotation using this model a "SIG encoding." The SIG is a constructionist model, in that it brings out coherence at both the local and global levels: what events happen, when, why, and to whom. Like the previous models we examined, in a SIG the entire discourse is modeled in a single, integrated data structure. It is descriptive, rather than prescriptive.

A SIG consists of three interconnected subgraphs called *layers*:

Textual layer. Analogous to the *text* layer in the van Dijk and Kintsch [1983] model, or the *discours* to Todorov, this is a linear vector of nodes that contain the **utterances of the original discourse** that is being modeled. While we only deal with textual discourse in the present study, nodes in the textual layer can also represent snippets of other kinds of media, such as oral storytelling. Each node contains anywhere from one proposition's worth of text to a paragraph or a passage, depending on its role in the overall structure of connected relations. Collectively, all these nodes represent the story as it is told from the telling's start to the telling's finish.

Timeline layer. Nodes in the timeline layer formally encode **story-world happenings** that have been expressed in the textual layer, such as events and statives. These nodes are arranged in a timeline that represents the sequence of story-world time. This layer is analogous to Todorov's *histoire*, van Dijk and Kintsch's *textbase* and the Formalist *fabula*. It represents the stated story content from the beginning of the story's chronology to the end of the story's chronology. Each node is annotated with the identity of the agent, if any, that is responsible for the narrated happenings (e.g., the perpetrator of an event). A more complete knowledge representation of the content in question, such as a predicate-argument structure, may also be attached to each node, though none is required.

Interpretative layer. The interpretative layer is analogous to the cognitive situation model. Here, nodes represent **goals, plans, beliefs, affectual impacts, and the underlying intentions of characters (agents) as interpreted by the story's receiver**. This includes both content that is directly stated (duplicating timeline-layer content) and content that is implied, but never stated outright, in the story as it is narrated. Its purpose is to **relate timeline-layer and textual-layer content by their motivational, intentional and affectual connections**, as opposed to their temporal connections as in the other two layers. For example, five actions in the story can all be intentional attempts to reach the same implied goal, which is represented as a node in the interpretative layer even though the narrator never explicates it. Collectively, the interpretative layer represents a receiver's agent-oriented interpretation of the narrative, with connections back to the stated content (in the textual and timeline layers) that justifies it.

This section introduces the set of node types and relations (arc types) that constitute the three layers of the SIG. Table 3.3 gives a summary of the node and arc types we will describe. To illustrate the instantiation of the schemata for encoding a particular story, we will apply the SIG to "The Wily Lion" and compare the result to those of previous models.

This approach differs from prior work in two important areas:

1. The discourse order of the surface text is preserved alongside the chronological order of the narrated content. That is, the SIG includes two temporal orderings of the stated story content, rather than one: an encoding of the discourse fragments in which they appeared in the narrated discourse (**telling time**, in the textual layer) and an encoding of the same content in the chronological order of the story-world being described (**story time**, in the timeline layer). The previous models we examined disregarded either the telling time (GRTNs, plot units) or the story time (grammars). In the SIG, both orderings are present and cross-referenced. In the next section, we will see how this is useful for modeling narrative discourse.

2. Previous models conflate what we call timeline and thematic content. For instance, plot units are built on a +/–/M system, with the + and – indicating both an event and

Symbol	Name	Usage/Signified Element
TEXTUAL LAYER		
TE	Text	Continuous span of surface discourse
f	Follows	Ordering of text spans in a discourse
TIMELINE LAYER		
S	State	An instant of story-world time
P	Proposition	A unit of discrete story-world content, such as an occurring action or event, typically pertaining to an agent. Can include a more complete knowledge representation of the narrated happening
T	Timeline	A continuum of time states in the story-world in a single modality
f	Follows	Ordering of States in a Timeline
ia	Interpreted as	Equivalence between TE and P nodes
in	In	Connects a State to its Timeline
ba	Begins at	Connects a Proposition to its temporal initiation State
ea	Ends at	Connects a Proposition to its temporal termination State
r	Referenced by	Connects a Timeline to a P or I node that incorporates it modally
e	Equivalent	Connects State nodes referring to the same moment in two Timelines
INTERPRETATIVE LAYER		
I	Interpretative Proposition	A unit of story content, equivalent to a P node in the Interpretative space. Either Hypothetical (H), Actualized (A) or Prevented/Ceased (PC) with respect to each State of the main Timeline
G	Goal	Indicates that certain I, G or B nodes are the goal of an agent
B	Belief	Indicates that certain I, G or B nodes are the belief of an agent
A	Affect	The baseline affectual state of an agent
in	In	Connects an I, G or B node to the G or B frame in which it is situated
ia	Interpreted as	Equivalence between a P node and an I, G or B node
im	Implies	Implication by a P node of an I, G or B node
a	Actualizes	Links a P node to an I, G or B node when the reader infers that the latter becomes actualized because of the former
c	Ceases	Links a P node to an I, G or B node when the reader infers that the latter becomes prevented/ceased because of the former
wc	Would cause	Link between one I, G or B node and another that is sufficient for its actualization
wp	Would prevent	Link between one I, G or B node and another that is sufficient for its prevention/cessation
pf	Precondition for	Link between one I, G or B node and another that is necessary for its actualization
pa	Precondition against	Link between one I, G or B node and another that is necessary for its prevention/cessation
ac	Attempt to cause	Indicates intention by the agent of a P node to actualize an I, G or B node
ap	Attempt to prevent	Indicates intention by the agent of a P node to prevent/cease an I, G or B node
p	Provides for	A positive affectual impact of an I, G or B node (traversing to A)
d	Damages	A negative affectual impact of an I, G or B node (traversing to A)

Table 3.3: Summary of the types of nodes and arcs that constitute Story Intention Graphs. Node types have capitalized symbols; arc types have lowercase symbols.

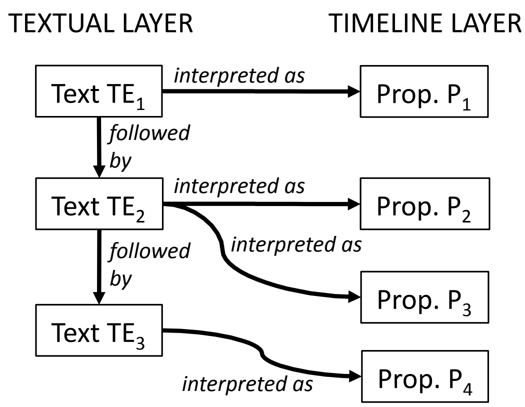
its affectual impact on an agent. GRTNs categorize story actions among a set of mutually exclusive, goal-oriented labels (Goal, Action, Outcome and so on). Story grammars include both temporal and goal-oriented rewrite rules (`EPISODE` and `GOAL_PATH`, respectively). These conflations make certain narrative scenarios difficult or impossible to describe, such as the hidden agenda, where one action serves distinct purposes in two separate plans. In the SIG, the timeline layer encodes only the temporal organization of the textbase, with its discrete nodes of narrated story-world content. Goal-oriented labels appear in the interpretative layer, which, while separate, is cross-referenced to the timeline and textual layers. We demonstrate in Appendix B that this modular approach enables the schemata to have a wide expressive range for describing many types of narrative situations, including hidden agendas.

3.3.1 Textual and Timeline Layers

The textual and timeline layers of the SIG include nodes for the surface form of the discourse and for the textbase form of the narrated story-world. The node and arc types are chosen to organize the textbase into a semantic network based around time.

In the textual layer, the discourse is divided up into fragments (continuous spans). Each fragment is represented by a Text node. Text nodes are chained together by *followed by* arcs so that the order of nodes in the chain reflects the order in which the fragments appear in the original discourse. The textual layer, then, encodes the "telling time" of the story. Each Text node is linked to a node in the timeline layer that represents an equivalent textbase happening. Figure 3.6 illustrates the beginnings of a SIG encoding for "The Wily Lion", including three text fragments in the textual layer and their equivalents in the timeline layer. The *interpreted as* arc indicates equivalence. For instance, the first Text node in the chain of Text nodes, TE_1, represents the first sentence of the discourse, "A Lion watched a fat Bull feeding in a meadow." This node is attached with *interpreted as* to a Proposition node containing an equivalent textbase unit. In this case, the unit is labeled with a propositional equivalent, `watch(lion, feed(bull, meadow))`.

The size of each fragment of surface text is set so that the resulting Text node can be connected to a uniformly equivalent P node in the timeline layer, which in turn is connected



Node	Example
TE_1	A Lion watched a fat Bull feeding in a meadow
TE_2	Going up to the Bull in a friendly fashion
TE_3	The Bull fell an easy prey to the Lion
P_1	`watch(lion, feed(bull, meadow))`
P_2	`approach(lion, bull)`
P_3	`friendly()`
P_4	`identity(bull, prey(lion))`

Figure 3.6: Fragment of a SIG encoding showing textual-layer nodes, as well as Proposition nodes in the timeline layer. A non-contiguous subset of "The Wily Lion" is encoded.

to an expression of agentive intent in the interpretative layer. In a concise discourse such as a fable, this is typically of clause or sentence length; in other cases, a longer passage may be reducible to a single functional unit with respect to an agent-oriented reading of the narrative.

Figure 3.6 shows the temporal structure in the textual layer with *followed by* arcs. In a more complete SIG encoding, the Proposition nodes are also temporally ordered (hence the name of the layer); crucially, though, they are arranged in a timeline that corresponds to "story time," the chronology of the story-world. The question of how to structure this arrangement is non-trivial. Time is the most fundamental discourse relation in a story, and most thematic content depends on an ordering of events (for instance, actions are followed by their consequences). A depiction of temporal ordering is common to all the representations we examined in Section 3.2, but each model simplified to some degree the many temporal relationships that can be found within a story. The relationship between story time and telling time can be quite complicated (see Mani *et al.* [2005] for a comprehensive review). Events in a story can occur over long or short periods of time, overlap, terminate one another, refer back or forward to other points in time, and cross over into hypothetical or imagined modalities. Even given a representation of time, the process of parsing the tense and aspect of narrative rhetoric (whether in English or another language) into a formal understanding of time is quite complex, subject to decades of work in linguistics (e.g., [Comrie, 1976; Comrie, 1985; Halliday, 1976; Nerbonne, 1986; Hornstein, 1990; Vlach, 1993]), natural language processing [Hinrichs, 1987; Webber, 1987; Passonneau, 1988; Mani and Pustejovsky, 2004], and artificial intelligence/database theory [Allen, 1991; Özsoyoglu and Snodgrass, 1995; Terenziani and Snodgrass, 2004]. Temporal and modal relations have also been singled out as bases for a discourse annotation scheme, TimeML [Pustejovsky *et al.*, 2003a], and associated annotated corpus, TimeBank [Pustejovsky *et al.*, 2003b]. While a detailed inquiry into the process of understanding or representing time is beyond the scope of this thesis, we will later revisit the relationship between a formal representation of time and English tense and aspect (Section 4.4).

We propose here a representation of time for the SIG that is robust, computable and amenable to manual annotation: The structure of the timeline is based on event *intervals* in the tradition of Allen's classic work on temporal reasoning [Allen and Ferguson, 1994]. Each P node takes place over an interval, which is a pair of states (points on a linear timeline). The states have a complete ordering which we can express with an enumeration, $t_1..t_n$.

In sum, there are four types of nodes in the textual and timeline layers:

Text node (TE) Represents a continuous span of surface discourse corresponding to a textbase happening (P) node and agentive interpretation in the interpretative layer.

State node (S). Represents an instant in time in the story-world. Each State has an associated time index t, a natural number:

$$t(S) \in \mathbf{N}$$

Timeline node (T). Represent a continuum of states in the story-world. There must be at least one main Timeline node, dubbed the **Reality timeline**, in a SIG encoding. As we will see, modal situations such as imagined past events are expressible with additional Timeline nodes.

Proposition node (P). An encoding of story-world content (a textbase happening, such as an event) that corresponds to a span of surface discourse. The happening occurs at a single State or an interval between States. (The interval can be unbounded in the case of events that never end or whose ending points are unimportant.) Epistemically, P node can belong to any **modality**: The content can depict an occurrence in the story-world's reality, an imagined concept such as a fear, an opinion or a metatextual comment. (The modality is set by the Timeline node associated with the Proposition node through the arcs we describe below.) If the story content involves an intentional agent, that agent is associated with the node as metadata.

Throughout this chapter, we illustrate P nodes with propositional (predicate-argument) encodings; however, despite the node's name, the *type* of encoding used to represent textbase content within a P node is unimportant to the SIG schemata. A P node may, for instance, have no encoding whatsoever—in this case, it only marks that a certain span of story text occurs at a certain story-world time, in a certain modality, featuring a certain discrete agent. In Chapter 5, we construct some SIG encodings using this "placeholder" technique, and others where each P node features a constructed propositional equivalent of the text span associated with the corresponding Text node.

There are seven relations in these layers, the first five of which are:

Followed by (f). Placed between one Text node and the Text node that immediately follows the first node in the original discourse. Also traverses between a State node and State node of the same timeline that immediately follows it. Logically, f is transitive, although the implied arcs are not drawn in the SIG:

$$f(S_1, S_2) \wedge f(S_2, S_3) \quad \Rightarrow \quad f(S_1, S_3) \tag{3.1}$$

$$f(S_1, S_2) \quad \Leftrightarrow \quad t(S_1) < t(S_2) \tag{3.2}$$

Interpreted as (ia). Traverses between a Text node and a Proposition node which represents the textbase equivalent of the discourse fragment associated with the Text node. There is a many-to-many relationship permitted with *ia*: a Text node can invoke several Proposition nodes, and each Proposition node can be justified by several discourse spans.

Begins at (ba). Traverses between a P node and the State at which the proposition first takes effect in the story-world. Such a relationship is not necessary for every P node. Propositions that do not link to a State with *ba* do not have a start time that can be inferred from the content of the corresponding Text node.

Ends at (ea). Traverses between a P node and the State at which the proposition culminates, stops or ends. Such a relationship is also not necessary for every P node. Propositions that do not link to a State with *ea* do not have an ending time that can be inferred from the content of the corresponding Text node.

If both *ba* and *ea* are given for a Proposition node, the beginning state must be followed by the ending state:

$$ba(P, S_1) \wedge ea(P, S_2) \Rightarrow f(S_1, S_2)$$

In (in). Traverses between a State node and the Timeline node representing the scope of story-world time in which it exists. A Timeline is said to "contain" all of the States that link to it with *in*, as well as all of the Proposition nodes that link to those States with *ba* or *ea*. We use the $\in$ notation as shorthand for *contained by*:

$$in(S, T) \quad \Leftrightarrow S \in T \tag{3.3}$$

$$(ba(P, S) \vee ea(P, S)) \wedge in(S, T) \quad \Leftrightarrow P \in T \tag{3.4}$$

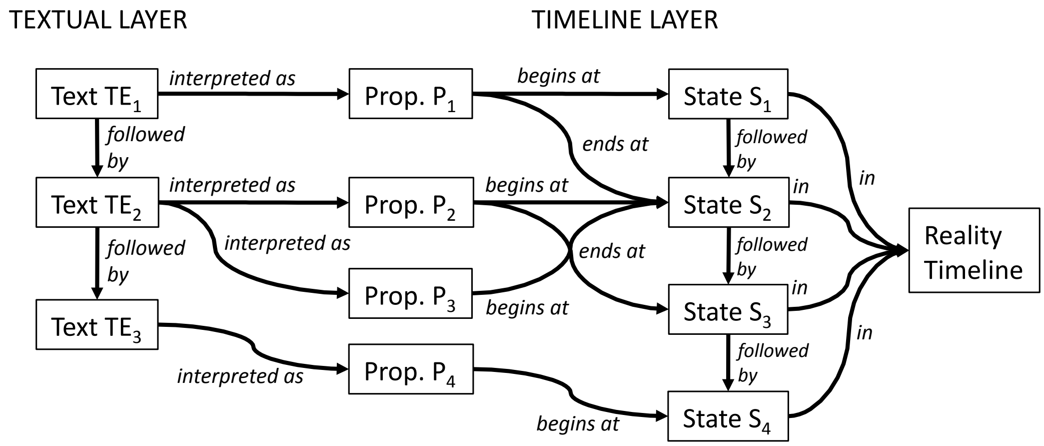
Note that a Proposition can only belong to one timeline:

$$ba(P, S_1) \wedge ea(P, S_2) \wedge in(S_1, T) \quad \Rightarrow \quad in(S_2, T) \tag{3.5}$$

$$ba(P, S_1) \wedge ea(P, S_2) \wedge in(S_2, T) \quad \Rightarrow \quad in(S_1, T) \tag{3.6}$$

The union of a Timeline node and all of the State nodes and Proposition nodes that it contains through *in*, *ba* and *ea* is collectively known as a **timeline**.

Figure 3.7 shows a representative example of these two layers of the SIG. Three fragments of "The Wily Lion" are mapped onto four propositions, including one modifier (one Text node has two outgoing *interpreted as* arcs). The temporal relationship between the first two events is what Allan terms *meets*: one's start time is the other's end time. At State S_4, a stative representing the bull's identity as the lion's prey begins; this is in essence a



Node	Example
TE_1	A Lion watched a fat Bull feeding in a meadow
TE_2	Going up to the Bull in a friendly fashion
TE_3	The Bull fell an easy prey to the Lion
P_1	`watch(lion, feed(bull, meadow))`
P_2	`approach(lion, bull)`
P_3	`friendly()`
P_4	`identity(bull, prey(lion))`

Figure 3.7: Example SIG encoding (textual and timeline layers only) for a non-contiguous subset of "The Wily Lion".

"become" action (the bull becomes the lion's prey). In all, the four propositions involve four time states, each of which is seen to be in the Reality timeline.

A complete textual- and timeline-layer encoding of "The Wily Lion" is shown in Table 3.4. The first two columns are the vector of Text nodes in the textual layer; the *followed by* arcs that join adjacent State and Text nodes are not shown. Also implied but not shown are the *interpreted as* arcs traversing from each Text node to the associated Proposition node(s) on its respective row. The outgoing arcs incident to each Proposition node are shown in the rightmost column—*begins at* and *ends at* arcs traversing to State nodes.

Story Time vs. Telling Time

Since the SIG features a mapping between the discourse ordering of events and the story-world ordering of events, it allows us to study a discourse in terms of the ordering and pacing of its fragments with respect to the story-world being described. From the nodes and arcs we have introduced, we can draw a "plot" of the discourse in which the horizontal axis is

Node	Text Node content	Node	Proposition node content	Arcs
TE_1	A Lion watched a fat Bull feeding in a meadow	P_1	watch(lion, feed(bull, meadow))	ba(S_1); ea(S_5)
TE_2	and his mouth watered when he thought of the royal feast he would make	P_2	thought(lion(potentialFuture(identity(bull, feast))))	ba(S_2); ea(S_{13})
		P_3	watered(mouth(lion))	ba(S_2); ea(S_3)
TE_3	but he did not dare to attack him,	P_4	¬attack(lion, bull)	ba(S_3); ea(S_{13})
TE_4	for he was afraid of his sharp horns.	P_5	afraid(lion, horns(bull))	ba(S_3); ea(S_{13})
TE_5	Hunger, however, presently compelled him to do something	P_6	compelled(lion, act)	ba(S_4); ea(S_5)
		P_7	hungry(lion)	ba(S_1); ea(S_{15})
TE_6	and as the use of force did not promise success	P_8	believe(lion, ¬promise(force, success))	ba(S_4); ea(S_{13})
TE_7	he determined to resort to artifice	P_9	plan(lion, artifice)	ba(S_4); ea(S_{13})
TE_8	Going up to the Bull in friendly fashion	P_{10}	approach(lion, bull)	ba(S_5); ea(S_6)
		P_{11}	friendly(lion)	ba(S_5); ea(S_{13})
TE_9	he said to him, "I cannot help saying how much I admire your magnificent figure."	P_{12}	say(lion, bull, admire(lion, figure(bull)))	ba(S_5); ea(S_6)
		P_{13}	say(lion, bull, magnificent(figure(bull)))	ba(S_5), ea(S_6)
TE_{10}	"What a fine head!"	P_{14}	say(lion, bull, fine(head(bull)))	ba(S_6); ea(S_7)
TE_{11}	"What powerful shoulders and thighs!"	P_{15}	say(lion, bull, powerful(shoulders(bull)))	ba(S_7); ea(S_8)
		P_{16}	say(lion, bull, powerful(thighs(bull)))	ba(S_8); ea(S_9)
TE_{12}	"But, my dear friend, what in the world makes you wear those ugly horns?"	P_{17}	ask(lion, bull, reason(wear(bull, horns(bull))))	ba(S_9); ea(S_{10})
		P_{18}	say(lion, bull, ugly(horns(bull)))	ba(S_{10}); ea(S_{11})
TE_{13}	"Believe me, you would do much better without them."	P_{19}	say(lion, bull, ifThen(¬have(bull, horns(bull)), succeed(bull)))	ba(S_{11}); ea(S_{12})
TE_{14}	The Bull was foolish enough to be persuaded by this flattery to have his horns cut off	P_{20}	foolish(bull)	ba(S_{12})
		P_{21}	persuade(lion, bull, allow(bull, cutOff(helper, horns(bull))))	ba(S_{12}); ea(S_{13})
TE_{15}	and, having now lost his only means of defense,	P_{22}	lose(bull, ability(bull, defend(bull)))	ba(S_{13}); ea(S_{15})
TE_{16}	fell an easy prey to the Lion.	P_{23}	identity(bull, prey(lion))	ba(S_{14}); ea(S_{15})

Table 3.4: A textual- and timeline-layer encoding of "The Wily Lion".

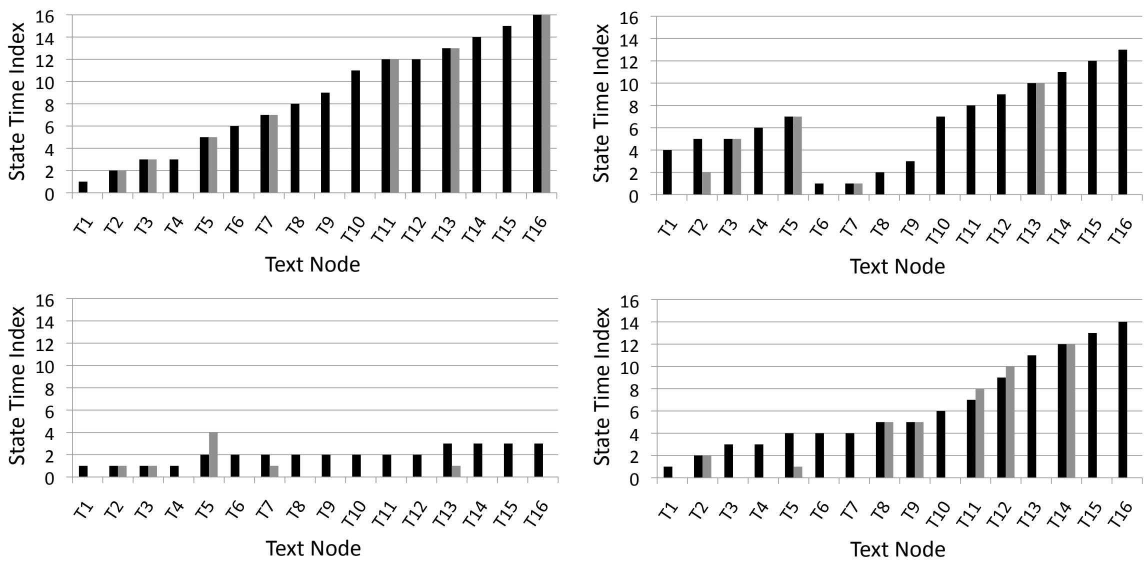


Figure 3.8: Telling time vs. story time. Clockwise from bottom left: a "slow" story, a "fast" story, a flashback, and "The Wily Lion" as modeled in Table 3.4.

telling time and the vertical axis is story time [Eco, 1995]. For each Proposition node on the timeline, we plot a point at (x, y) where x is the ordinal position of the first Text node linking to the P node with *ia*, and y is the time index of the State node linked by the P node with *ba*. If the curve increases monotonically, each new span in the discourse advances the story time (a linear telling). A flat curve indicates that multiple discourse spans describe a single point in story time (a suspension of time). Certain other curves indicate that the narrator is "flashing back" or forward, or moving quickly or slowly through a period of story time (the so-called "tempo" of a story) [Mani, 2010].

Figure 3.8 gives four such plots, three for hypothetical stories and one for "The Wily Lion". For every Text node along the horizontal axis, there are one or two bars for linked Proposition nodes. (There is usually one bar, but if a single span maps to two P nodes, the second is plotted in an adjacent grey bar.) Clockwise from the bottom left are: A "slow" story, in which the narrator describes several moments in detail; a "fast" story, where nearly every span invokes a new time state; a "flashback" story, where the narrator begins in the middle of the story and interrupts the flow of time to give a scene of background information; and on the bottom right, "The Wily Lion" as modeled in Table 3.4. In the latter case, temporal modeling alone (divorced from information about goals and plans) suggests thematic content. The first half of the fable moves slowly through time, detailing a moment with a set of statives relating to the lion. He is established as the protagonist; his mental states in a particular moment dominate the attention of the storyteller. In the second half of the fable, time moves swiftly until the story's conclusion. The lion begins to act externally until the bull performs his only actions as agent in TE_{14}-TE_{15}. The story is one of thinking, then doing.

Alternate Timelines

We have seen the "Reality timeline" used as a model context for textbase happenings in the timeline layer. Additional modalities are represented by separate Timeline and Proposition nodes. These **alternate timelines** indicate hypothetical and imagined modalities. Through the use of *referenced by* arcs, they take functional roles in Reality-timeline P nodes. For example, at a particular moment an agent may express speculation that some action will happen in the future. In our timeline encoding of "The Wily Lion", node P_2 depicts a thought process by the lion about the royal feast the bull "would make" in a potential future. The feasting on the bull does, in the end, happen, but P_2 does not itself jump forward in story time because it concerns the lion's mental state in the "present." We used a predicate `potentialFuture` for this scenario. With an alternate timeline we can instead create a "scope of time" that represents speculation, and use an arc to refer to it in P_2. This feature adds formality to the model's representation of imagined, desired or feared events, which are common in thematically rich stories.

Hypothetical events and states are particularly important to character agency, since goals and plans are themselves potential futures which may or may not come to pass. Hypothetical *pasts* are also thematically important, as agents sometimes try to reason about the history of the story-world. In a mystery story, the detective has a goal for the future (to solve the case), but that goal is to develop hypotheses about *prior* events based on a collection of evidence.

To reiterate, the SIG allows additional Timeline nodes that have their own sets of State and Proposition nodes. We integrate these into the lerger graph with two additional arcs:

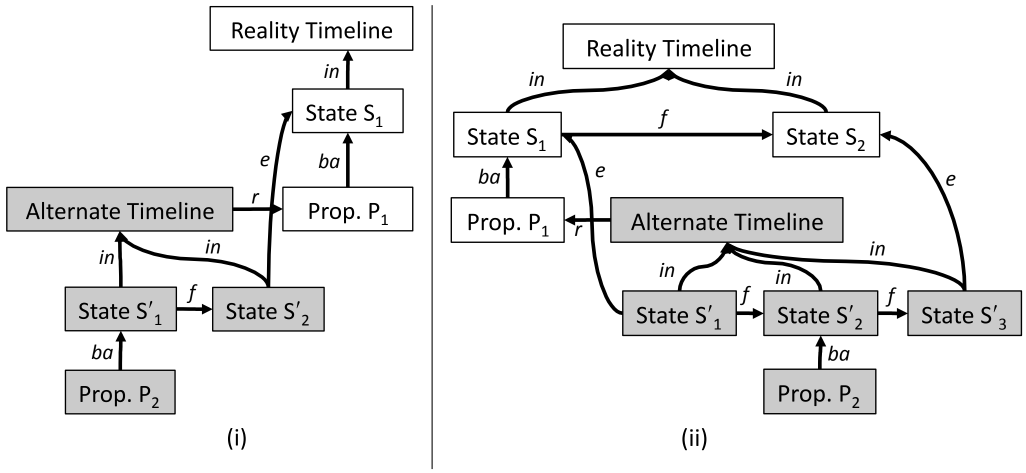
Referenced by (r). Traverses from a Timeline node T to a Proposition node P that incorporates the timeline in a modal context. The timeline containing P is said to be a "parent" of the timeline represented by T. An alternate timeline is, in a sense, a "inner scope" of narrative reality, existing wholly and exclusively in the context of the outer scope that references it. Timeline relationships are therefore tree-like, in that a timeline can have multiple children (inner scopes) but only one parent (outer scope). The Reality timeline is the root of the tree, and the "ancestor" of all alternate timelines:

$$
\begin{aligned}
(P_1 \in T_1) \wedge r(T_2, P_1) &\Rightarrow T_1 \neq T_2 \\
(P_1 \in T_1) \wedge r(T_2, P_1) &\Rightarrow parent(T_1, T_2) \\
parent(T_1, T_2) \vee (parent(T_1, T_3) \wedge ancestor(T_3, T_2)) &\Rightarrow ancestor(T_1, T_2) \\
ancestor(T_1, T_2) &\Rightarrow \neg ancestor(T_2, T_1)
\end{aligned}
$$

Equivalent (e). If timeline T_1 is a parent of timeline T_2, a State node $S_1 \in T_1$ is equivalent to a State node $S_1 \in T_2$ if S_1 and S'_1 are two modal contexts of the same functional State. That is, *equivalent* indicates that the same time slice is manifest as two State nodes in different timelines, and joins the nodes together as a common point of reference. The e arc can only join two states in different timelines that have an ancestral relationship. Multiple e arcs are permitted between the same two timelines, with the logical constraint that the relative ordering must be preserved:

$$e(S'_1, S_1) \wedge e(S'_2, S_2) \wedge f(S_1, S_2) \Rightarrow f(S'_1, S'_2)$$

Figure 3.9 shows fragments of two example SIG encodings that invoke alternate timelines. The Reality timelines are depicted in white node boxes; the alternate timelines are drawn in grey. In 3.9(i), an action references a timeline in which a modal state S'_2 is *equivalent* to the action's state S_1 in the Reality timeline. The modal state S'_2 is then preceded by another modal state S'_1 containing an imagined action P_2. Because the *equivalent* arc establishes a common point of reference between the two timelines, any action at S'_1 occurs "some time previous" to both S'_2 in the modal timeline and S_1 in Reality. This topology might represent that a character is thinking about an action that occurred at some point in



Node/Figure	Example
(i)	**Henry thinks that Orson graduated from Cornell.**
State S_1	A moment of time.
Prop. P_1	Henry believes in the events of a separate scope of time.
Alternate Timeline	The scope of time believed by Henry in P_1.
State S_2'	Within Henry's belief in P_1, the present moment of Henry's belief.
State S_1'	Within Henry's belief in P_1, a moment prior to S_2' (Henry's believing).
Prop. P_2	Orson graduates from Cornell.
(ii)	**Clarissa will have bought the flowers by 6 P.M.**
State S_1	A moment of time prior to 6 P.M.
State S_2	6 P.M.
Prop. P_1	The events in a separate scope of time occur.
Alternate Timeline	A scope of time.
State S_1'	Within the separate scope of time, the present moment equivalent to S_1.
State S_2'	Within the separate scope of time, a moment between S_1' and S_3'.
State S_3'	Within the separate scope of time, a moment corresponding to 6 P.M.
Prop. P_2	Clarissa buys the flowers.

Figure 3.9: Two configurations of alternate timelines in the timeline layer of a SIG.

the past (previous to the moment of thinking): "Henry thinks that Orson graduated from Cornell" would be one example. 3.9(ii) similarly depicts a modal context for a possible future action P_2, because P_2 is attached to a modal state S_2' which follows the common point of reference $e(S_1', S_1)$. This figure, however, adds another temporal constraint by employing a second *equivalent* arc between S_3', which follows S_2', and S_2, which follows S_1. Because P_2 occurs between S_1' and S_3', in Reality it is imagined to occur between S_1 (the moment of imagining) and some future time S_2. A character may, in this case, be promising that some event happen will happen by a future deadline represented by S_2, e.g., "Clarissa will have bought the flowers by 6 P.M."

Both of these examples use an e arc to attach the modal timeline to the Reality timeline at a common point of reference. In the absence of any e arcs, a modal timeline represents an entirely separate narrative scope with no points of attachment to Reality. This occurs when a story embeds a fictional inner story as told by a character (a "frame narrative"), such as *One Thousand and One Nights* and its serial storyteller Scheherazade. A nested story can be modeled as an embedded SIG encoding, with the storyteller-character taking the role of "focalizer" (narrating agent) [Bal, 1981; Bronzwaer, 1981; Genette, 1983]. The speech actions of Scheherazade become the discourse utterances of her own story; in essence, the timeline layer of the framing SIG encoding becomes the textual layer of the nested encoding.

Linguistically, alternate timelines allow us to model tenses and aspects in the discourse that refer to an ambiguous span of time in the Reality timeline. Consider the sentence: "John started to make breakfast but went to the store because he ran out of eggs." It is semantically unclear whether John used his last egg during his breakfast preparation, perhaps dropping it, or if he used the last egg in some prior episode. A modal timeline such as Figure 3.9(i) preserves this ambiguity by asserting that an event (running out of eggs) takes place at some time in the past—the exact past time is unknown because there are no additional *equivalence* arcs to provide bounds. The past participle tense, "he had run out of eggs," would allow us to draw such a bounding *equivalence* arc. It tells us that the "running out" event occurs prior to the "making breakfast" event.

Such a use of the *equivalence* arc is analogous to the notion of the *reference time* in Reichenbach's [1947] study of tense and aspect. In Reichenbach's approach, the temporal interpretation of a sentence is governed by the relative ordering of three important time points: the speech time S, the event time E, and a temporal point of reference R. This system maps onto the present model of alternate timelines. S is the point of attachment in the parent timeline (the state associated with the incoming *referenced by* arc), E is an event in the alternate timeline, and R is a time state in the parent timeline with an incoming *equivalence* arc. In 3.9(ii), S would be S_1, the speech time; E would be P_2, the event time, and R would be S_2, the reference time. This figure can be read in the future perfect tense given by the ordering S-E-R: "Clarissa will have bought the flowers" (by time S_2). This mapping assumes that a primary *equivalence* arc establishes a modal time state equivalent to the speech time, S_1' in this case. In the absence of a secondary *equivalence* arc, R is set to be the same as S, resulting in a simple future tense with no separate reference time ("Clarissa will buy the flowers"). We will further investigate the relationship between alternate timelines and a model of tense and aspect in Section 4.4.

Discussion of the Representation of Time

We believe this approach to representing time is robust, in the sense that it is tolerant of partial encodings of the *fabula* timeline. Unlike temporal databases, we do not tie each state to a particular UTC timestamp or formally represent the relative lengths of time intervals signified by *followed by* (e.g., 9 hours 4 minutes passed between S_1 and S_2). In addition, if complete interval information cannot be inferred from the discourse, the timeline can be instantiated with *begins at* arcs alone (reducing the timeline to a set of points rather than intervals). The essential aspects of the SIG timeline are the relative orderings of TE and P nodes (via States and *begins at* relations).

This is not to say that reductionism is appropriate in all cases. Situations in which more precise information about time is relevant to the thematic content of the story are modeled <u>in terms of that relevance</u>. For example, the drama of the Puccini opera *Madama Butterfly* (with libretto by Luigi Illica and Giuseppe Giacosa) hinges on the long absence of Pinkerton, a U.S. Naval Officer, from Japan. Pinkerton has married a local girl, Cio-Cio San, who endures loneliness and financial hardship while waiting for Pinkerton's ship to return. She gives up opportunities to remarry even though she does not hear from him for years. The drama derives not from the mere length of time that passes between the opera's acts, as Cio-Cio-San waits, but from the decisions she takes to maximize her happiness based on a trust of Pinkerton's intentions. When Pinkerton finally returns with an American wife in tow, the tragedy is not that an exact number of years has elapsed, but rather that a tremendous opportunity cost has been exacted from the heroine—Cio-Cio San's plan has backfired and precluded her from finding happiness by any means. The large passage of time was the <u>enabler</u> of the affectual harm to Cio-Cio-San, but not the harm itself. That thematic idea, which is the purpose of the interpretative layer, can be encoded even though time is itself only represented as an ordering function.

In the next section, we transition away from the textual and timeline layers, and introduce the many aspects of the interpretative layer that allow us to model the tragedy of *Madama Butterfly* and other stories.

3.3.2 Interpretative Layer

The interpretative layer of the SIG depicts a situation model of the story-world. Like the timeline layer, it contains a set of nodes that represent story-world happenings. The major difference from the timeline layer is the manner in which these nodes are organized: Rather than by time, the interpretative layer takes a theory-of-mind (agent-centric) approach, structuring content by its motivational, intentional and affectual connections. We call the layer "interpretative" because the situation model is a subjective artifact that reflects a particular receiver's interpretation of the story's agents and their motivations. While we have developed annotation guidelines, the process of arriving at such an interpretation is not itself a part of the schemata.

Let us first define the final node types: Interpretative Proposition (I), Belief (B), Goal (G), and Affect (A). We call these the **interpretative nodes**.

Interpretative Proposition (I). The equivalent of P nodes in the interpretative context, these nodes represent story-world content such as events and statives that may or may not have been expressed in the surface discourse.

Belief (B). A belief node acts as a frame, inside of which the content of other nodes is understood to be a state of the story-world in the mind of a discrete agent. This agent is an inherent and immutable attribute of the node (so that every Belief node that is instantiated is associated with an agent). This agent can be a single intentional entity or a set of entities who share the same beliefs. We use the notation $B{:}X()$ to describe a Belief frame, with X referring to the agent in question, and the content of the frame appearing as a set of arguments. An unlimited number of interpretative nodes can be placed inside the frame. A belief frame can itself be negated to assert a *lack* of belief in its content (note the distinction between believing a statement N is false, $B{:}X(\neg N)$, and not having the belief that the statement is true, $\neg B{:}X(N)$).

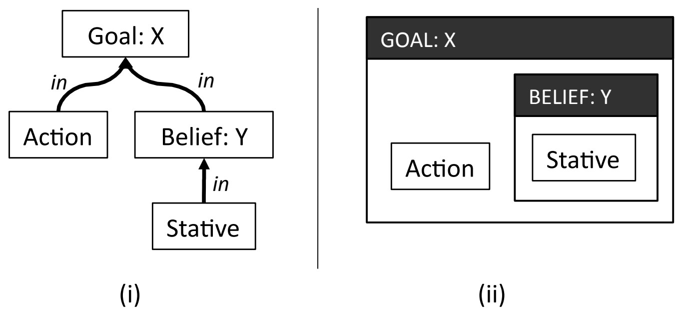


Figure 3.10: Nested agency frames, in two forms of graphical notation.

Goal (G). A goal node acts as a frame for other interpretative content, similar to a Belief. The difference is that the content of a Goal frame is understood to the state of the story-world as *desired* by the discrete agent. We notate Goals as $G{:}X()$.

Affect (A). An Affect node represents a baseline affectual state with respect to a discrete agent. As in Belief and Goal nodes, the agent can be a single intentional entity or a set of entities.

An *in* arc appears in the interpretative layer with a semantic meaning distinct from that of the *in* arc of the timeline layer:

In (in), additional. Traverses between an interpretative node and a Belief or Goal node representing the frame in which the interpretative node is situated. Each interpretative node must have 0 or 1 outgoing *in* arcs (each node can only belong to one frame), but a Goal or Belief node can have an unlimited number of incoming *in* arcs.

For the clarity of this discussion, we will draw Goal and Belief frames as graphical boxes that contain the nodes connected with *in* arcs (their content). Figure 3.10(i) depicts an example SIG encoding fragment that represents a two-part goal of agent X: for some action to happen, and for another agent Y to believe that some stative is true. Such a situation could be: "Larry wanted (to win the chess game against Debra, and have Debra believe (that he is a skilled player))." 3.10(ii) shows the same graph fragment, but is drawn using the box notation. Note that "Action" and "Stative" nodes are, logically, both Interpretative Proposition (I) nodes; we label them more specifically for clarity.

Agency frames—goals and beliefs—can be nested indefinitely to model theory-of-mind interpretations of narrative meaning, as we saw in Section 3.2.2. This allows us to represent not only what agents believe about the world, but also what they believe each other's beliefs, about each other's beliefs about others' beliefs, *ad nauseam*. When an interpretative proposition (I) node is not placed in any agency frame, it is in what we call the **ground truth** of the SIG: that which the narrating agent of the story asserts to be true in the scope of the story-world.[4]

The rest of this section is divided as follows: We first describe **actualization**, which allows the SIG to express changes in interpretative content over story time (such as when an outstanding goal is resolved through an outcome). We then introduce arcs to connect interpretative nodes into **plans** and **attempts**. Finally, we describe the manner in which **Affect** nodes may be used to express the affectual impact of interpretative content on certain agents.

3.3.2.1 Actualization

The interpretative layer, in and of itself, is timeless. As we noted in the introduction to this section, previous cognitive situation models conflate temporal and interpretative connections. The present model instead separates these two types of discourse relations, with temporal connections in the textual and timeline layers, and thematic content (goals, plans, and so on) in the interpretative layer.

However, time is still crucial to the interpretative layer. Goal outcomes, for example, must temporally follow attempts. The SIG assigns a temporal dimension to interpretative content by relying on its connections to the timeline layer to determine what interpretative nodes "happen," and in what order. For each State node in the Reality timeline, a set of logical entailments determines which nodes in the interpretative layer are occurring at

[4]The narrator may, of course, be unreliable [Booth, 1961]. Frame narratives are a particular risk, since the storyteller is itself a character who might be interested in distorting the facts. In a sense, all story-worlds are constructions of artificial realities, even when they purport to be non-fiction, because of the editing process inherent in the intentional act of storytelling [Genette, 1972]. For our purposes, all content placed in ground truth represents the objective reality of the story-world.

the corresponding point in story time, and which are not. This computation is called **actualization**.

For example, consider again the interpretative goal in Figure 3.10, in which Larry wants to win a chess game and have his opponent Debra believe that he is a skilled player. This figure has no sense of time—it is only a goal in isolation. But a timeline can actualize certain pieces of it in sequence, and draw a story out of it:

- At state S_1, the story is beginning.

  No one has any goals. The entirety of Figure 3.10 is **hypothetical** (immaterial) at this point in time, because it does not yet exist in the story-world.

- At state S_2, Larry develops a goal to win a chess game against Debra and have her think he is a skilled player.

  Larry goes from having no goal to having this particular goal. The goal frame becomes **actualized**. The goal content inside the frame (winning the game and having Debra believe he is skilled) is hypothetical, rather than actualized—it is a possible future.

- At state S_3, Larry wins the chess game against Debra.

  The "winning" action inside the goal frame transitions from being hypothetical to being actual. The goal frame is still actual. Since Larry has a goal to win the chess game at S_2, and he does in fact win the chess game in S_3, he has achieved a positive outcome on this aspect of his goal. In general, this is the mechanism by which we express goal outcomes.

- Also at state S_3, Debra comes to believe that Larry is an <u>unskilled</u> chess player (perhaps she believes he has won the game unfairly).

  The belief frame nested within Larry's goal frame transitions from being hypothetical to being demonstrably false (what we call **prevented/ceased**). Debra's opinion of Larry's skills goes from undefined to "not skilled." Larry has reached a negative (failed) outcome on this aspect of his goal.

This example demonstrates the logical property of the SIG whereby interpretative content has a certain **actualization status** relative to every state in the Reality timeline. The actualization status of the goal frame was hypothetical until it became "actualized" at S_2; the node representing the goal action itself (for Larry to win) was hypothetical until it was actualized at S_3; the frame representing the other aspect of the goal (Debra's belief) was hypothetical until it was "prevented/ceased" in S_3.

In general, a node's actualization status relative to some point in story time is always one of three conditions that describe the truth (within the story-world) of the node's content at that time:

1. **Hypothetical (H).** The node's content is in a hypothetical state, existing as a concept rather than as an assertion of a story-world happening. The present truth of a Hypothetical node is indeterminate; no assertion is made about whether the content is true within the story-world at the moment in question, or not.

2. **Actualized (A).** The node's content is true (in effect; currently occurring in the story-world). Successful goal content is given A status at the point when it becomes successfully true.

3. **Prevented/Ceased (PC).** The node's content is false (not in effect; decisively incompatible with the story-world). Nodes that are prevented/ceased not only are untrue at the present time, but given the current state of affairs, have been prevented from happening in the foreseeable future. In the language of prior models, a goal with a failed outcome has PC status.

Formally, we let the actualization status s of an interpretative node I at some time index n be one of {H, A, PC}, and let every node's status at time 0 be Hypothetical:

$$\forall I \in \mathbf{I} : \forall n \in \mathbf{N} : s(I, n) \quad \in \quad \{H, A, PC\} \tag{3.7}$$

$$\forall I \in \mathbf{I} : s(I, 0) \quad := \quad H \tag{3.8}$$

Every interpretative node logically carries one actualization status for each State in the Reality timeline. In the example above, there are three states, S_1 to S_3, and four nodes of interpretative content. There are therefore 12 actualization statuses defined for this example, one for every node-state combination.

Actualization status transitions are **triggered** by particular arcs that traverse from the timeline layer to the interpretative layer. Using these arcs, the Reality timeline acts as an instruction set and clock for determining the actualization status of each node for each state. This is accomplished by virtue of the fact that Proposition nodes are totally ordered

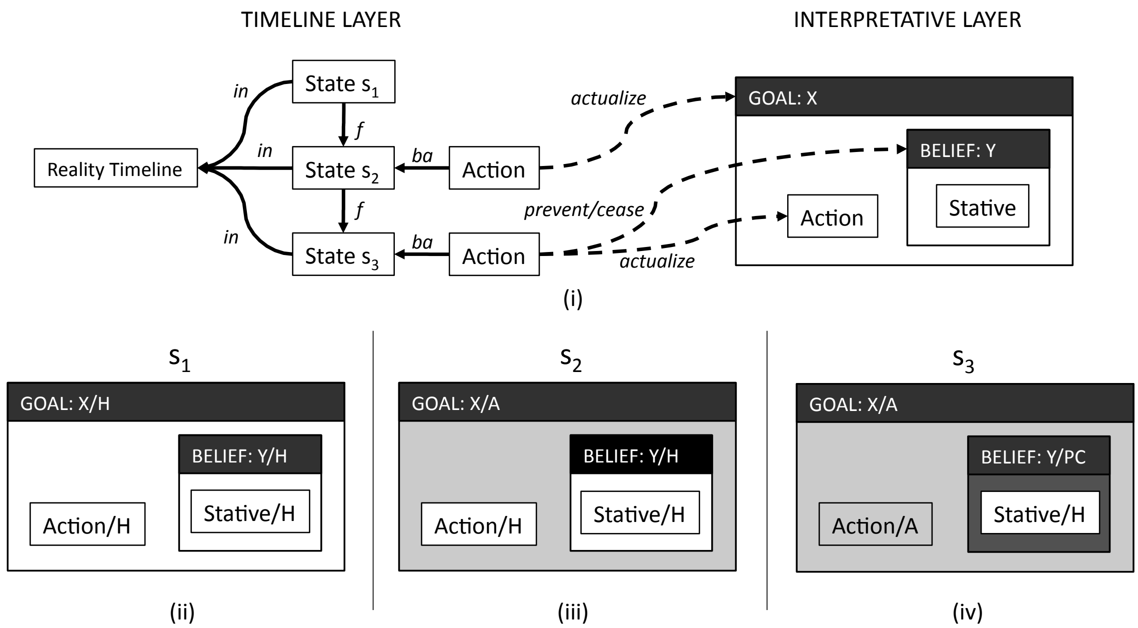


Figure 3.11: SIG encoding fragment showing timeline and interpretative layers, as well as the actualization status of an interpretative goal at three discrete time states.

in story time, from the first P node in the earliest state (attached to the State node with no incoming *f* arcs) to the last node in the latest state. (For purposes of ordering propositions, only *ba* arcs are considered; *ea* arcs do not trigger changes in interpretative actualization.)

All interpretative nodes begin in Hypothetical (H) status. Then, each successive Proposition node associated with the Reality timeline has the opportunity to trigger a change in the actualization status of one or more interpretative nodes. There are two types of triggers: those that *actualize* (*act*) and those that *prevent/cease* (*pc*). These are not SIG arc types, but useful shorthands, as we will soon introduce multiple arc types that logically entail either *act* or *pc*. To apply this to our chess example (see Figure 3.11(i)):

- At state S_1, the story is beginning.

  There are no triggers.

- At state S_2, Larry develops a goal to win a chess game against Debra and have her think he is a skilled player.

  There is an arc that triggers *actualize* traversing from the motivating action at state S_2 to the node representing Larry's goal frame.

- At state S_3, Larry wins the chess game against Debra.

Prior Actualization Status	Incoming Trigger	New Actualization Status
Hypothetical	*Actualize*	Actualized
Hypothetical	*Prevent/Cease*	Prevented/Ceased
Actualized	*Actualize*	Actualized
Actualized	*Prevent/Cease*	Prevented/Ceased
Prevented/Ceased	*Actualize*	Actualized
Prevented/Ceased	*Prevent/Cease*	Prevented/Ceased

Table 3.5: Transition of interpretative node actualization status upon receiving a trigger from a new time state.

There is an arc that triggers *actualize* traversing from the "win" action at state S_3 to the "win" action inside the Larry's goal frame.

- Also at state S_3, Debra comes to believe that Larry is an unskilled chess player.

  There is an arc that triggers *prevent/cease* traversing from the "win" action at state S_3 to the "believe skilled" frame inside the Larry's goal frame.

The actualization status of a node at a particular time state is a function of two factors: the node's previous status (that is, its status with respect to the preceding P node), and the presence of any incoming trigger arcs. The effects of the two types of triggers are summarized in Table 3.5. *Actualize* triggers always cause the interpretative node to become Actualized, no matter the prior status; *prevent/cease* triggers always cause the interpretative node to become Prevented/Ceased. In other words, the truth-value of a node may alternate between Actualized and Prevented/Ceased, or remain Hypothetical.

Formally, we let $a(P, I)$ and $pc(P, I)$ indicate that proposition node P triggers *actualize* and *prevent/cease* (respectively) on some interpretative node I, such that both entail an actualization status for I with respect to the time index associated with P:

$$a(P, I) \land ba(P, S) \land t(S) = n \quad \Rightarrow \quad s(I, n) := A \tag{3.9}$$

$$pc(P, I) \land ba(P, S) \land t(S) = n \quad \Rightarrow \quad s(I, n) := PC \tag{3.10}$$

In the absence of any incoming arcs that entail such triggers, the actualization status of an interpretative node I at some time index n is unchanged from the previous time index $n - 1$:

$$\forall n \in \mathbf{N} : \forall I \in \mathbf{I} : (\neg \exists P : ((a(P, I) \lor pc(P, I)) \land ba(P, S) \land t(S) = n) \\ \Rightarrow \quad s(I, n) := s(I, n - 1))$$

As we have seen, Figure 3.11 illustrates the triggering of actualization transitions by timeline P nodes. 3.11(i) shows a partial SIG encoding. The timeline layer includes two actions and the initial state S_1. The interpretative layer includes the same multi-part goal seen in Figure 3.10: Agent X (Larry) wants both for an action to happen (to win the game) and for Agent Y (Debra) to believe that some stative is true (that Larry is skilled). The actualization status of the goal at each state is shown in 3.11(ii-iv) by means of a suffix associated with each node: /H, /A, or /PC. Actualization statuses are also drawn graphically, with light shading for Actualized status (/A), dark shading for Prevented/Ceased status (/PC) and no shading for Hypothetical status (/H). The two timeline actions trigger three actualization status changes: there are two *actualize* triggers and one *prevent/cease* trigger. As these are not themselves SIG arcs, they are drawn with dashed arrows.

With respect to nesting, actualizations must proceed from the "outside in." No node can be actualized or ceased if it is in a frame that still has Hypothetical status.

3.3.2.2 Actualizing Arcs

Let us now introduce the first four of the 13 SIG arc types which relate to the interpretative layer. These relations always connect timeline-layer nodes to interpretative-layer nodes, and are the only arc types that trigger actualization status changes. The first three of these are *actualizing* triggers; the last (c) is a *preventing/ceasing* trigger.

Interpreted as (ia), additional. Traverses between a timeline P node and an interpretative frame or I node when there is a direct equivalence (that is, the content of the interpretative node is a paraphrase of the content of the timeline node); the same arc is similarly used to connect equivalent nodes between the textual and timeline layers.

Implies (im). Traverses between a timeline P node and an interpretative frame or I node when the interpretative content can be inferred from the timeline content, but there is not a direct equivalence. This "weaker" form of *ia* connects a timeline happening with a node of interpretative content that it entails without stating outright. It should not be used if *ia* is possible.

Actualizes (a). Traverses between a timeline P node and any interpretative node (frame, I node or Affect node) when the interpretative content is actualized as a causal result. This "weaker" form of *im* and *ia* connects a timeline P node with a node of interpretative content that it *causes*, but does not either state or entail directly. It should not be used if *ia* or *im* are possible, but rather, when a timeline happening can be inferred to indirectly trigger an actualization.

Ceases (c). Traverses between a timeline P node and any interpretative node (frame, I node or Affect node) when the interpretative content is prevented/ceased as a causal result. Like *a*, it signifies that a timeline event implies a consequence in the interpretative layer.

Through I nodes and *ia*, *im* and *a* arcs, the model supports the representation of any fact that might be implied or stated by a timeline P node. This does imply that a proper encoding is one that takes every opportunity to encode a consequence of an event, whether stated or unstated in the original discourse. That is, we wish to avoid the "frame problem" in artificial intelligence, in which an event has a prohibitively large number of possible consequences throughout the story-world to formally model [McCarthy and Hayes, 1969]. In a SIG, the task is not to model interpretative propositions for *every* consequence of an event, but for only those consequences that significantly impact the thematic content of the story. The test for "impact significance" has to do with the Affect nodes we will soon introduce: Any consequence that does not impact the affectual state of at least one agent in a manner that the storyteller explicated or implied in the story should not be encoded as a node.

Figure 3.12 shows a possible interpretative encoding for a small section of the "Wily Lion" timeline, as outlined in Table 3.4. The action at S_1, in which the lion watches the bull feed from the meadow, actualizes two nodes: the frame indicating that the bull wants to feed from the meadow, and the interpretative action that indicates his successful feeding. The initial story state at S_1 is one in which there is one goal, and it is already being successfully fulfilled. The action at S_2 implies that the lion has conceived of a goal to eat the bull; S_{14} triggers both an actualization of the lion's goal content (a successful outcome for the lion) and a cessation of the bull's goal content (a loss for the bull). The overall dramatic arc in

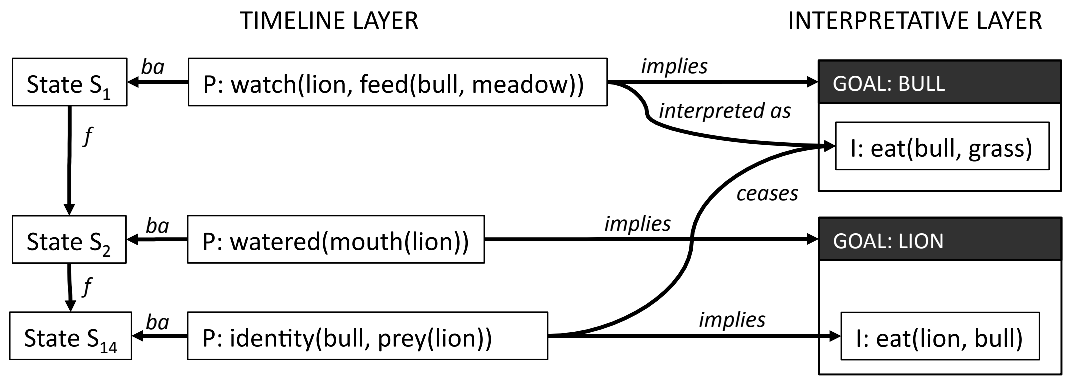


Figure 3.12: SIG encoding fragment showing a possible interpretative-layer encoding for three timeline propositions in “The Wily Lion”.

this encoding of the story is one of a tradeoff—the lion’s goal is satisfied by the same action that ceases the satisfaction of the bull’s goal.

3.3.2.3 Plans

We have seen actualization triggers indicate goal outcomes. However, we earlier saw that such outcomes are only part of a thematic narrative. Our schemata also needs a representation for a strategy toward fulfilling a goal—a plan, with subgoals that make progress toward actualizing the larger goal. In the case of “The Wily Lion”, the plan is a multi-stage scheme on the part of the lion to take advantage of the bull’s vanity, so that the bull takes action which removes (ceases) the horns which serve as an obstacle in the way of a hot lunch. The plan is never stated in the text; we are told that the lion decides to use artifice, but the details of the intended artifice are never made explicit. It is up to the reader to infer what the plan is. Our schemata allows a receiver to encode not only his or her inference of the lion’s plan, but the gradual reveal of that plan by the narrator as the discourse unfolds. The next set of relations provide a mechanism for describing the strategies and possible futures of each agent, whether implied or explicit.

A plan is modeled in the interpretative layer as a chain of connected nodes inside a Goal frame. Each node is a “subgoal” that leads to the ultimate goal at the end of the chain. The chain is connected with directed arcs that indicate **expected causality**: the agent believes that the actualization of one subgoal would lead to the actualization of the superordinate goal that lies next on the chain, and so on, leading to the ultimate goal. Crucially, these expectations are themselves beliefs of the agent. These beliefs may be mistaken. For instance, an agent may devise a plan to bring about rain by praying to rain gods, even though in the ground truth of the story-world, no causal connection exists between the acts of praying and raining.

As we mentioned, each interpretative node begins with Hypothetical (H) status. This is true for each of the subgoal steps of a plan as well. Just as a single goal is understood to have a successful outcome when it is actualized, a plan is a multi-stage goal where each step can be individually actualized when (and if) it is achieved. Similarly, a plan that fails can be ceased at the point of failure.

The relations that define expected and/or intended futures are:

Would Cause (wc). Traverses from one interpretative node to another interpretative node. Signifies that in the belief context of the originating node, an actualization of the originating node would causally lead to (is sufficient for) an actualization of the destination node.

Would Prevent (wp). Traverses from one interpretative node to another interpretative node. Signifies that in the belief context of the originating node, an actualization of the originating node would causally lead to (is sufficient for) a prevention/cessation of the destination node.

Precondition for (pf). Only differs from *would cause* in that it signifies that an actualization of the originating node is necessary for an actualization of the destination node (but not necessarily sufficient).

Precondition against (pa). Only differs from *would prevent* in that it signifies that an actualization of the originating node is necessary for a prevention/cessation of the destination node (but not necessarily sufficient).

As an illustrative example, Figure 3.13 shows an interpretation of the lion's plan. The actualization statuses are drawn with respect to state S_5 in the timeline laid out in Table 3.4, when the lion approaches the bull. At this point in the story, the lion has actualized a

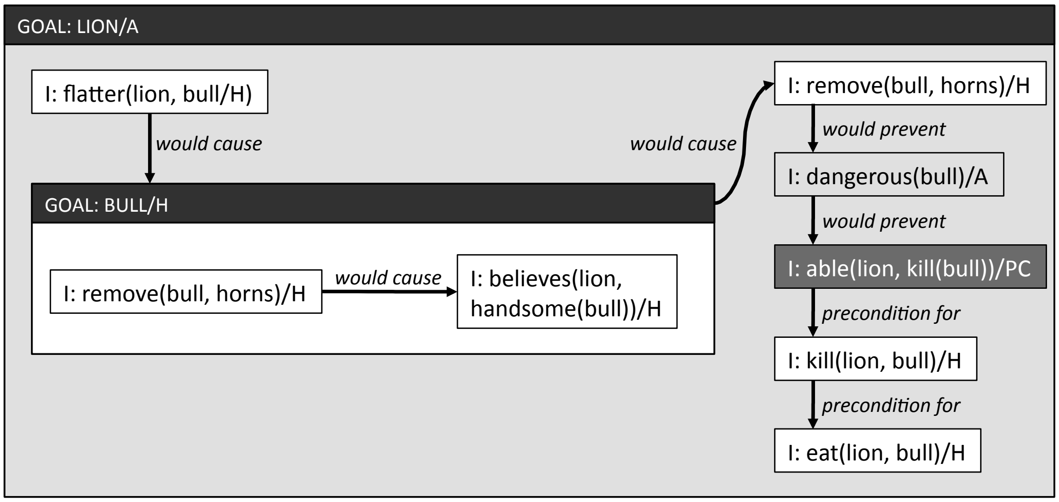


Figure 3.13: SIG encoding fragment showing a multi-step plan in "The Wily Lion".

mental state in which he lays out a plan consisting of a seven-step causal chain. The lion plans to instill a goal on the part of the bull, which (the lion believes) would cause the bull to have his horns removed. The removal of the horns would allow the lion to kill and eat the bull.

As we mentioned earlier, applying theory of mind to literature suggests that much planning involves the management of the goals, plans and beliefs of others. In this example, a nested goal frame acts as a step in a plan, a subgoal that must be actualized in the same manner as if it were an I node. The lion's plan calls for the bull to construct his own plan in which removing his horns is the first step. When the bull takes this action, it implies that the bull has indeed decided to embark on such a strategy (i.e., he has actualized the inner goal frame), and the lion can proceed with the next step of his larger plan.

Note in particular the duplication of `remove(bull, horns)` in two contexts. It exists in the bull's plan as a means to making the bull handsome in the lion's view, but in the lion's plan as a means for killing the bull. When the bull does have his horns removed, it actualizes both nodes and furthers both plans. As such, this is a graph topology that depicts an ulterior motive or a hidden agenda.

In general, it is not necessary for all elements of a plan to be within the same structural frame (that is, all connecting to the same Goal node with *in* arcs). The meaning would be the same if the last four elements of the chain (from `dangerous(bull)` to `eat(lion, bull)`) were in a separate goal frame of the lion's. In this case, the *would prevent* arc traversing to `dangerous(bull)` would cross from one goal frame to another. Plan chains may also involve segments in ground truth; only the beginnings and ends of plans must be inside goal frames.

In general, a plan can include not only the sequential actualization of I nodes, but the deliberate *cessation* of a node which is blocking the route to a goal via *would prevent*. Triggering *prevent/cease* on an actualized node that, through *would prevent*, blocks a desired state is a form of "double negation" that is equivalent to actualizing a node that would in turn actualize a desired state. Briefly put, a plan step can either be about ceasing an undesired state that is actualized, or actualizing a desired state that is hypothetical or ceased. In this particular case, the lion's problem is that the bull holds a certain attribute, that it is well defended by its horns, and the actualized nature of that fact is preventing the lion from being able to kill and eat the bull. Only by ceasing the attribute, thereby cutting off the triggering of *prevent/cease* on his `able()` stative, does the lion gain the power to pursue his ultimate goal.

Would cause and *would prevent* arcs carry no logical constraints regarding the actualization statuses of either their source or their destination nodes at any state in the timeline. They do, however, imply that the agent *expects* the actualization status of the destination node to change once the actualization status of the source node changes. This expectation may be violated, and that violation may be a crucial dramatic turning point. Aristotle [1961] defined this as *peripeteia*, the point in a tragedy when the hero suffers a reversal of fortune after his expectations are violated. Peripeteia often goes hand-in-hand with *anagnorisis*, when the hero undergoes a revelation about himself and his situation. In Figure 3.13, the lion's plan is predicated on the expectation that actualizing `remove(bull, horns)` would cease `dangerous(bull)`. The lion may have found that, contrary to his expectation, the bull continued to be a formidable opponent without horns—and therefore he was the one who had been tricked while attempting to be the trickster. Such an outcome would be an example of both anagnorisis and peripeteia. We further explore the capability of the SIG to represent these concepts in Appendix B.

The distinctions between *would cause/prevent* and *precondition for/against*, respectively, are that satisfying a precondition does not cause the agent to expect the actualization status of the destination node to change—the agent believes that the preconditions are necessary but not necessarily sufficient. We use *precondition for* for the last two of the lion's plan steps because they are about enablement; whether he chooses to exploit this ability once it is actualized is up to him. In general, the precondition arcs are useful for when a goal requires multiple parallel plans. In "Little Red Riding Hood," the wolf seeks to fool the girl into believing that he is her grandmother by succeeding in two parallel tasks: disguising himself as the grandmother in appearance, and feigning the grandmother's voice. Both are preconditions but neither is sufficient for the girl to lower her guard.

As we have mentioned, a plan may include steps which are inferred by the reader, in that there are no equivalent textual-layer or timeline-layer nodes. In this case, it is never stated in the original story that the lion's plan is to trigger a plan on the part of the bull. This inference is enabled by world-knowledge and mind-reading processes that are not themselves a part of the descriptive SIG schemata.

It is technically possible that the lion has an altogether different plan at S_5 than the one depicted in Figure 3.13. The illustrated plan assumes that all the events following S_5 transpire more or less as envisioned by the lion at S_5. It is possible, however unlikely to us, that the lion sincerely wishes for the bull to become handsome because that would satisfy his hunger by other means (such as by increasing tourism and economic activity in their corner of the plains), and that when the bull removes his horns, the lion unfortunately succumbs to the baser instincts he has been repressing in a bid to remain acceptable to the civilized world. Such a reading is more about the lion's internal conflict with his moral compass than it is about the bull's foolishness. In a larger sense, stories rarely explicate the mental states of all their agents at all times; most, like this one, explicate some mental states and strongly imply others through action. Some narratives deliberately leave intentions and beliefs ambiguous. Our approach allows an annotator to encode his or her own reading of the entire story, including both explicit and implicit thematic content. Both types of content can then become data for automatic processing such as the identification of analogies. (We attempt this in Chapter 5. While multiple encodings can also represent plural readings by

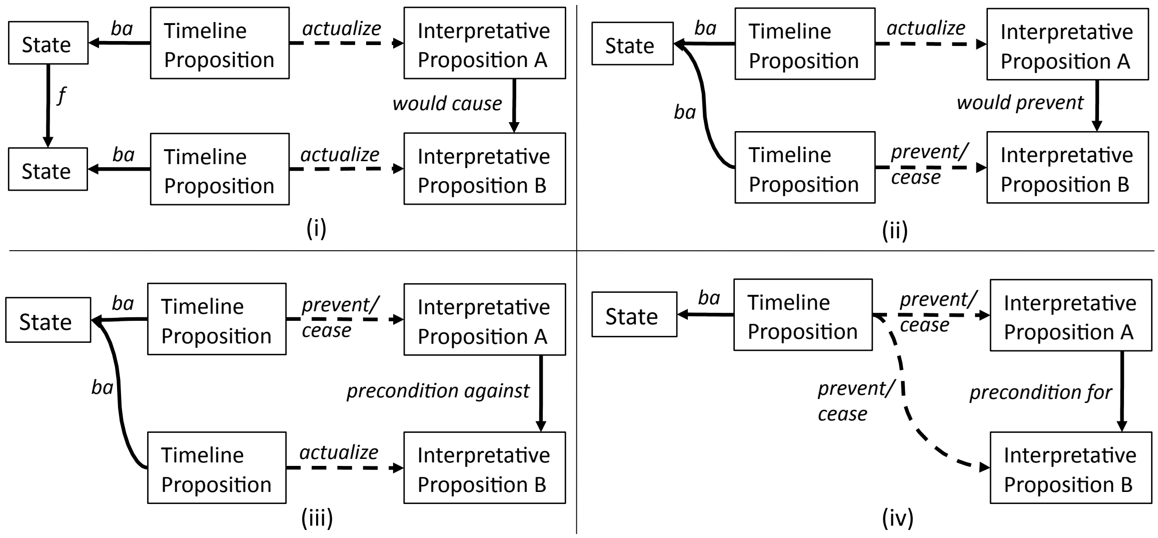


Figure 3.14: Causality in the SIG: The four graphical relationships between two interpretative propositions, A and B, from which we infer from the SIG that A (or its prevention/cessation) causes B (or its prevention/cessation). See also Appendix C.2.

the same annotator, each with alternative inferences, we asked annotators to settle on a single, preferred reading for this experiment.)

3.3.2.4 The Inference of Causality

The SIG can represent **actualized causality** between two I nodes A and B (and, by extension, the timeline P nodes that actualize or prevent/cease them) in four ways. These follow intuitively from the definitions of *precondition for/against* as "necessity" relationships and *would cause/prevent* as "sufficiency" relationships:

1. A is newly actualized, B is newly actualized at the same or a following time state, and A *would cause* B (Figure 3.14(i)). A caused B, in whole or in part.

   Example: "Going to the loud concert would give Thomas tinnitus. Thomas went to the loud concert, so he got tinnitus."

2. A is newly actualized, B is newly prevented/ceased at the same or a following time state, and A *would prevent* B (Figure 3.14(ii)). A caused the prevention/cessation of B, in whole or in part.

   Example: "Going to the concert would prevent Adam from getting to work on time the next day. Adam went to the concert, so he failed to go to work on time the next day."

3. A is newly prevented/ceased, B is newly actualized at the same or a following time state, and A *precondition against* B (Figure 3.14(iii)). The prevention/cessation of A allowed B to happen, in whole or in part.

   Example: "Nathan's excellent social skills, among his other abilities, kept him from losing his job. When he became extremely antisocial, his company decided to let him go."

4. A is newly prevented/ceased, B is newly prevented/ceased at the same or a following time state, and A *precondition for* B (Figure 3.14(iv)). The prevention/cessation of A allowed the prevention/cessation of B to happen, in whole or in part.

   Example: "The financier's support was an integral part of the art gallery's operational budget. The financier pulled his support, so the art gallery folded."

See Appendix C.2 for formal descriptions of these four scenarios. Note that they are symmetric: We may take 3.14(i), invert B and its incoming arcs, and arrive at 3.14(ii) as an alternate formulation of the same underlying relationship (e.g., "Going to the loud concert would end Thomas's run of <u>not</u> getting tinnitus."). Also note that for illustrative purposes, Figure 3.14 varies the temporal relationship between A and B. In 3.14(i) and 3.14(ii), A and B are linked to separate timeline P nodes in sequential time states. In 3.14(iii), A and B are linked to separate nodes in the same time state. In 3.14(iv), A and B are linked to the same P node. All three temporal scenarios allow the inference of causality from A to B. However, the flow of time cannot be reversed; no causality can be inferred if, in any of the five scenarios, B begins at a state preceding that of A's onset state.

These examples demonstrate the representation of "ground truth" with respect to the causal relationships between events. In addition, as we mentioned above, the causality arcs can be used inside certain belief contexts (that is, agency frames). In this case, they represent an agent's belief about the causal relationship between two hypothetical or actual events—a belief which may or may not be true with respect to the ground truth. For example, an agent may be mistaken or ignorant about what would happen if it tries to execute a plan, or it may draw a false conclusion about the causal antecedent of an event. We give example encodings of such scenarios in Appendix B.

3.3.2.5 Epistemology of Belief Frames

In the interpretative layer, both nodes and frames can have incoming arcs that trigger *actualize* or *prevent/cease*. The actualization of a frame refers to the mental state of the agent associated with the frame, but not necessarily to the content found within the frame. For instance, the actualization of a belief frame that proposition A is true says nothing about whether A is indeed true—it only asserts that the agent believes A. In the chess example from Figure 3.11, we triggered *prevent/cease* on Debra's belief that Larry is a skilled player. This denotes that Debra does not believe Larry is skilled, but makes no logical assertion about whether Larry is skilled or unskilled. The I node regarding Larry's attribute as a skilled player is left with Hypothetical (indeterminate) status at the end of the timeline.

There are three ways to logically indicate that the agent is correct or incorrect in its belief in an assertion A:

1. Actualize or prevent/cease (respectively) the node containing A itself, inside the belief frame;
2. Actualize or prevent/cease (respectively) an interpretative node with identical content (A) in ground truth; or
3. Prevent/cease or actualize (respectively) an interpretative node with negated content ($\neg$A) in ground truth.

Once actualized, the semantic meaning of the belief frame depends on the structure of the subgraph found within the frame. If an I node inside a belief frame has an outgoing arc, the representation means that the agent <u>believes the relation</u> rather than the assertion in the node, and the actualization of the frame makes no commitment to whether the assertion is true or whether the agent believes the assertion to be true. Figure 3.15 illustrates an I node in a belief frame with and without an outgoing arc. In Figure 3.15(i), the stative E_1 is believed by Agent X; in Figure 3.15(ii), agent X believes that there is a causal relationship between hypothetical statives E_1 and E_2 but is not depicted to believe that either is true or false; finally, in Figure 3.15(iii), the agent believes *both* that E_1 is true and that E_1 would cause E_2. Figure 3.15(iii) depicts what we call an *expectation* of X that E_2 is actualized by

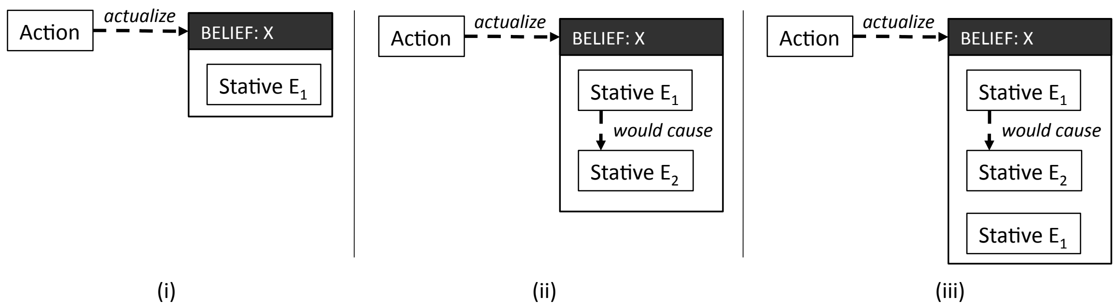


Diagram	Example	Example
(i)	Andy thought that April was in Toronto.	Caroline thought it would rain within the hour.
(ii)	Andy thought that if April was in Toronto, she could not come to his birthday party.	Caroline thought that if it rained within the hour, she would have to reschedule her arboretum tour.
(iii)	Andy thought that April was in Toronto, and therefore could not come to his birthday party.	Caroline thought it would rain within the hour, in which case she would have to reschedule her arboretum tour.

Figure 3.15: Belief frames in a SIG can refer to (i) an agent's belief in a proposition such as a stative, (ii) an agent's belief in the hypothetical relationship between two propositions, or (iii) the combination of (i) and (ii) with respect to a single proposition.

a subsequent timeline happening. None of these three examples commit to whether E_1 is ever true, whether E_2 is ever true, or whether E_1 would truly cause E_2.

3.3.2.6 Attempts

We have seen four types of SIG arcs that connect the timeline and interpretative layers (*interpreted as*, *implies*, *actualizes* and *ceases*). There are two others. These deal with the **intentionality** of an agent with respect to a goal or a plan:

Attempt to cause (ac). Traverses between a timeline P node with an agent and an interpretative frame or node when it is to be understood that the timeline happening is performed by its agent as an intentional attempt to bring about the actualization of the interpretative content, such that any actualizations/cessations by the timeline node are unintended by the agent unless they are also connected to the node with an *attempt* arc.

Attempt to prevent (ap). Traverses between a timeline P node with an agent and an

> interpretative frame or node when it is to be understood that the timeline happening is performed by its agent as an intentional attempt to bring about the prevention/cessation of the interpretative node, such that any actualizations/cessations of the timeline node are unintended by the agent unless they are also connected to the node with an *attempt* arc.

Neither of these arcs triggers a change in the actualization status of the interpretative node to which it connects. They do, however, signal that the agent in a timeline node is knowingly and willfully acting to try to bring about such a change in actualization status. These arcs are analogous to the Attempt (A) nodes in Trabasso's cognitive model; in a larger sense, they are motivated by the experiments we saw in Section 3.2.1 that actions executed in pursuit of goals are a key part of memory retention for goals themselves and for stories overall.

In much of the present fable, many of the actions are understood as being attempts to actualize a part of a plan. The timeline nodes P_{10} through P_{19} in Table 3.4 show the lion approaching the bull and speaking to him. With respect to the lion's plan in Figure 3.13, these actions do not themselves actualize or cease any interpretative nodes. They are, however, related to the plan, in that they are all an *attempt to cause* the hypothetical first step (`flatter(lion, bull)`).

3.3.2.7 Affect

The final aspect of the interpretative layer, and our overall schemata, is the **Affect node**. This node is designed to represent an ultimate answer to the question of *why* an agent acts.

In a representation without Affect nodes, "why" can be answered with plans that generalize to larger and larger purposes. We say that the lion tricks the bull in order to be able to kill it, so that it can eat it; unfortunately, this does not intrinsically represent the meaning of `eat()` to either the predator or the prey. We are concerned with how an action fits into an overall plan, but in the case of `eat()`, the "plan" is simply commonsense biology (the lion wants to eat in order to digest the bull, which gives the lion a source of protein, which is metabolized into energy, which is expended to sustain basic life functions, and so on). None of these nominally superordinate goals are necessary for understanding the story. At some point, a goal's rationale must be general enough that it is recognizable as a constant aspect of the human condition. Eating is necessary for health, and health "just is" as a rationale for action. We simply wish to encode that the lion's eating of the bull is ultimately good for the lion, in terms of affectual impact, and bad for the bull.

Affect nodes fulfill this purpose by acting as affectively-charged termini for plans. They represent our premise that thematic content is ultimately about the relationships between agents and their basic needs. For each interpretative node, they answer the question: Why is this node ultimately relevant to an agent? Why is this aspect of the story interesting as an aspect of a tellable narrative? In short, Affect nodes are the way the we represent the affectual impact of each interpretative node with respect to each agent.

As a SIG node, Affect (A) can be instantiated in the interpretative layer an unlimited number of times, but each instance must be connected to the larger graph structure. Each instance includes two features, an agent and a type; we will describe types in a moment. Affect nodes are connected to interpretative nodes and frames by one of the two following arcs:

Provides for (p). Traverses from an interpretative node to an Affect node. Signifies that in the belief context of the interpretative node, the actualization of that node implies a positive affectual impact on the Affect node's agent in a manner consistent with its type, and the prevention/cessation of the interpretative node has a corresponding deleterious impact.

Damages (d). Traverses from an interpretative node to an Affect node. Signifies that in the belief context of the interpretative node, the actualization of that node implies a deleterious affectual impact on the Affect node's agent in a manner consistent with its type, and the actualization of the interpretative node has a corresponding positive impact.

Semantically, Affect nodes represent the basic needs of agents as conscious entities. To provide for an Affect node is to positively affect the agent in question; to damage an Affect node is to negatively affect the agent. Table 3.6 considers an interpretative node that relates to an Affect node and lists the effects of changing the actualization status of the

Prior Status	New Status	Arc	Meaning
Hypothetical	Actualized	Provides For	The agent achieves something positive
Hypothetical	Prevented/Ceased	Provides For	The agent is hit by something negative
Actualized	Prevented/Ceased	Provides For	The agent loses something positive
Prevented/Ceased	Actualized	Provides For	The agent is freed of something negative
Hypothetical	Actualized	Damages	The agent is hit by something negative
Hypothetical	Prevented/Ceased	Damages	The agent avoids something negative
Actualized	Prevented/Ceased	Damages	The agent is freed of something negative
Prevented/Ceased	Actualized	Damages	The agent is hit by something negative

Table 3.6: Interactions between actualization status transitions and arcs relating to Affect nodes.

interpretative node. In essence, actualizing a node which provides for an Affect node helps the agent in question, and preventing/ceasing the node hurts the agent in question.

Logically, an encoding can be read “backward” from Affect nodes to understand the affective context of nodes that are not directly connected. For instance, in Figure 3.16(i), Agent X has a goal in which E *would cause* F, which itself *provides for* an Affect node. E thus has an indirect but positive affectual impact on X. (Our drawings depict Affect nodes as non-frame labels with white text over a black fill.) When X attempts to cause E, X is also attempting to cause F and, in turn, actualize the Affect node. B, E and F are all oriented as being ultimately about a positive affect for X. We call this type of inference **goal closure**. In Appendix C.1, we give a set of formal rules for inferring the affective meaning of a node based on an arbitrary number of “hops” to an Affect node.

Not everything that transpires in a story deserves to be oriented toward a particular agent’s affectual state. A description of a space, for instance, might serve no other purpose than to set the scene for the reader. Accordingly, our schemata does not call for Affect nodes to be attached to every interpretative node. To illustrate this, Figure 3.16(i) includes two interpretative propositions that exist in ground truth: C, which has no affectual impact, and D, which has a deleterious affect on an agent by linking to an Affect node with *damages*.

However, in a proper SIG *all* content inside goal frames must relate to one or more Affect nodes, either directly or through closure. Closure is possible for a node if there is a path in the graph leading from the node to an Affect node. The path must only follow some combination of *in*, *would cause*, *would prevent*, *precondition for*, *precondition against*, *provides for* and *damages*. This rule has a graphical interpretation which stipulates that

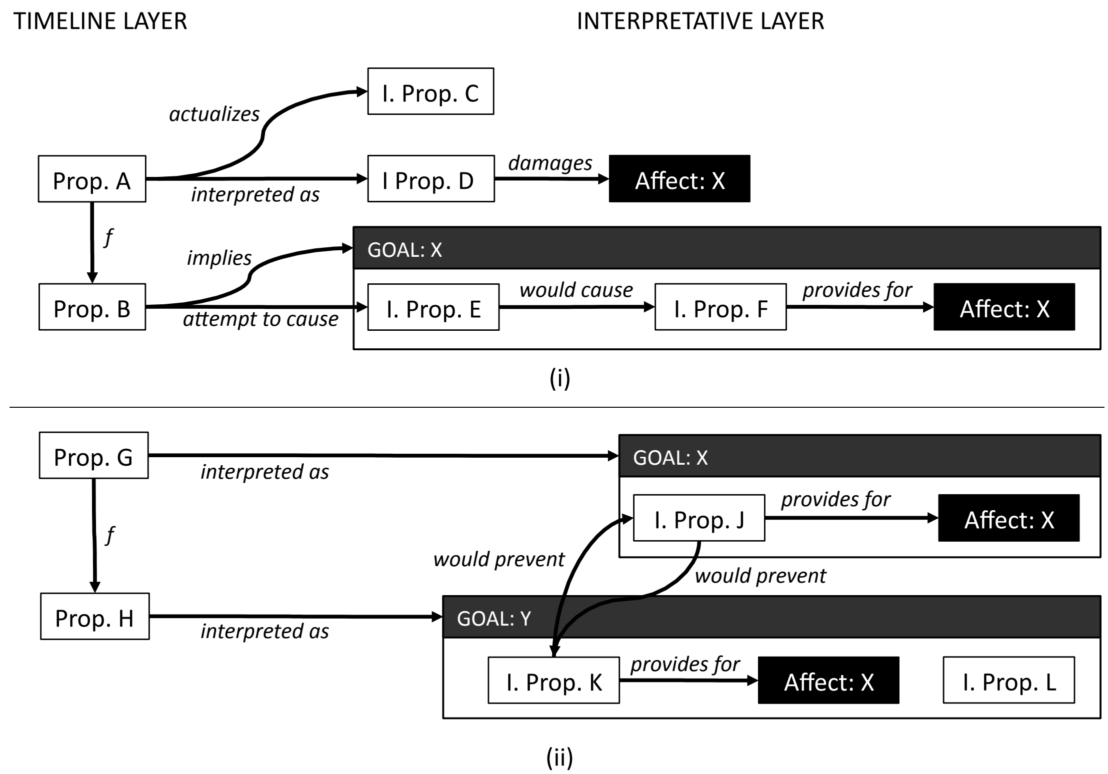


Figure 3.16: Legal SIG encoding (top) and one that violates Affect node usage.

all content inside a goal frame must **drain** to an Affect node by following arcs of these seven types until it reaches one—that is, for each goal node, one should be able to trace a path from the timeline through that goal node to an Affect node while only following forward arcs. A node inside a goal frame for which there is no path to an Affect node violates the schemata. The purpose of this rule is to ensure that every goal is annotated with the affectual impact that motivates it. Figure 3.16(ii) illustrates an illegal encoding: Interpretative Proposition L, in the goal frame for Agent Y, does not have any outgoing arcs through which it could drain, and so no possible affectual impact can be ascribed to it.

Another restriction on goal closure is that **no cycles** are allowed. That is, the subgraph of a SIG encoding that only includes these seven arc types and the nodes that are incident to them must be a directed acyclic graph (but not necessarily a connected one). This is because a goal closure arc implies both temporal and causal ordering. The causal path of a hypothetical plan must always propagate forward to an Affect node. 3.16(ii) is also an illegal encoding because nodes K and J *would prevent* each other, forming a causal cycle. One might suggest such an encoding in order to convey the concept of two mutually exclusive goals, such as in a conflict between two agents. The proper encoding for this scenario, demonstrated in Section B.5, uses two additional plan steps (one for each frame) to convey the same relationship without causing a cycle.

Affect nodes always represent the "ground truth" of affectual impact on an agent. They cannot be placed inside goal or belief frames with *in* (though for graphical convenience, we sometimes draw them inside frame boxes). Strictly speaking, Affect nodes are not the endpoints of agent-intended plans, but metadata provided about each plan. This does not limit the expressibility of the schemata in terms of agent beliefs about affect. One may still encode a scenario where an agent expects an event to cause a positive affectual impact, only to have it cause a negative impact (Figure B.5).

3.3.2.8 Affect Typing

We mentioned earlier that P nodes in the timeline and I nodes in the interpretative layer can be cross-indexed with other annotation schemes applied to the same discourse. For example, a propositional encoding of a span of text, with semantic role labeling, can be associated with a Proposition node (hence its name). Only the identity of the agent (if any) is strictly necessary metadata for each node. The same is true of Affect nodes. Each instantiation of an Affect node can be cross-indexed with a knowledge representation for the "type" of affectual impact represented by the node. This can be useful when multiple Affect nodes are used for different purposes according to the semantics of the narrative. For instance, one may use two Affect nodes in a scenario where a single event is good in one manner and bad in another manner (a trade-off—see Figure B.2).

We make no claim in this thesis as to what knowledge representation is best for Affect nodes with respect to any narrative corpus, including Aesop's. We only claim that such typing can increase the expressive range of the schemata. However, for purposes of demonstration, we have devised and implemented a set of types based on prior investigations into the psychology of human motivation. The most well-known set of types is a hierarchy of needs devised by Maslow [1943]. To Maslow, "practically all organismic states are to be understood as motivated and as motivating," a sentiment which dovetails with our notion of goal closure. He identifies five broad categories: physiological needs (those needed to maintain bodily homeostasis, such as food and sleep), safety needs (protection from wild animals, extremes of temperature, criminals, etc.), and the needs for love (affection and belongingness), esteem (self-confidence and the respect of others), and self-actualization (the fulfillment of one's potential). These categories are hierarchical in that one tends to not be a concern unless the previous ones are satisfied. In another classification, Max-Neef [1992] devises an ontology of needs along two interacting dimensions: existential (being, having, doing and interacting) and axiological (subsistence, protection, affection, understanding, participation, creation, leisure, identity and freedom). Max-Neef argues that these basic needs are not only few, finite, and classifiable, but consistent across cultures and through historical periods. We have adapted these typings into the following twelve Affect types found in Table 3.7.

Although these categories are distinct, they are non-exclusive. One action may simultaneously cover multiple types. Such an action would be connected to multiple Affect nodes. For example, a father caring for a sick son is acting both for his son's health and for his own love. We presented these types to annotators in the experiments we will describe in Chapter 5. We notate Affect nodes with a period between the agent and the type (e.g., `X.FREEDOM`).

Figure 3.17 augments Figure 3.13, the plan diagram for "The Wily Lion", by adding typed Affect nodes. The plan now represents the notion that the lion's overall purpose is to help himself by providing for `LION.HEALTH`. His plan for doing this involves instilling a goal on the bull's part to act toward his own positive ends. More specifically, the lion prompts the bull to act in such a way that would favor the bull's ego. The same action, `remove(bull, horns)`, advances toward three affect states when it is actualized: The bull believes it is helping the bull's ego, but the lion knows it is enabling him to eat the bull—an action that the lion seeks for purposes of his health, but that has the side effect of ending the bull's life.

Type	Description	Motivation
Life	Continuation of basic life functions; existence vs. non-existence.	Subsistence [Max-Neef, 1992]
Health	Freedom from pain, disease, malnutrition, and other physical/mental ailments. (If a loss permits the character to live in greater pain, it is a Health matter; if life and death are immediately at stake, it is a Life matter.)	Safety [Maslow, 1943], Protection [Max-Neef, 1992]
Ego	A positive perception of one's qualities by one's self and by others.	Esteem [Maslow, 1943]
Wealth	Material possessions or currency, above that needed for basic sustenance (those for Health).	Esteem + Leisure [Max-Neef, 1992]
Love	Feelings of fondness, warmth, and romance for and from another person; familial companionship; compassion or a desire to heal the world.	Affection [Max-Neef, 1992]
Leisure	Entertainment and enjoyment, whether from peaceful solitude, active socializing, or another form of recreation.	Leisure [Max-Neef, 1992]
Membership	Feeling of belonging to a group; acceptance by its other members and holding one's self positively by the norms and customs of that group. The group can be ethnic, social, economic, or in the micro sense, about cliques and clubs.	Identity [Max-Neef, 1992]
Actualization	Fulfillment of one's artistic, athletic, spiritual, professional or other aspirational potential in an elective endeavor.	Self-actualization [Maslow, 1943], Creation [Max-Neef, 1992]
Freedom	The state of being unrestricted in movement, action and behavior, whether the restricting force is other characters, natural forces, or an internal struggle.	Freedom [Max-Neef, 1992]
Justice	The perception that one's code of ethics is being executed fairly; the desire to see good outcomes come to those whose actions one believes are moral.	Safety [Maslow, 1943]
Enlightenment	A more full and accurate view of the world, whether through education, spirituality or other means.	Understanding [Max-Neef, 1992]
Honor	The perception that one is fulfilling one's own code of ethics, and that of the law and moral code of the community to which one belongs.	Participation [Max-Neef, 1992]

Table 3.7: Affect typing used for the present study.

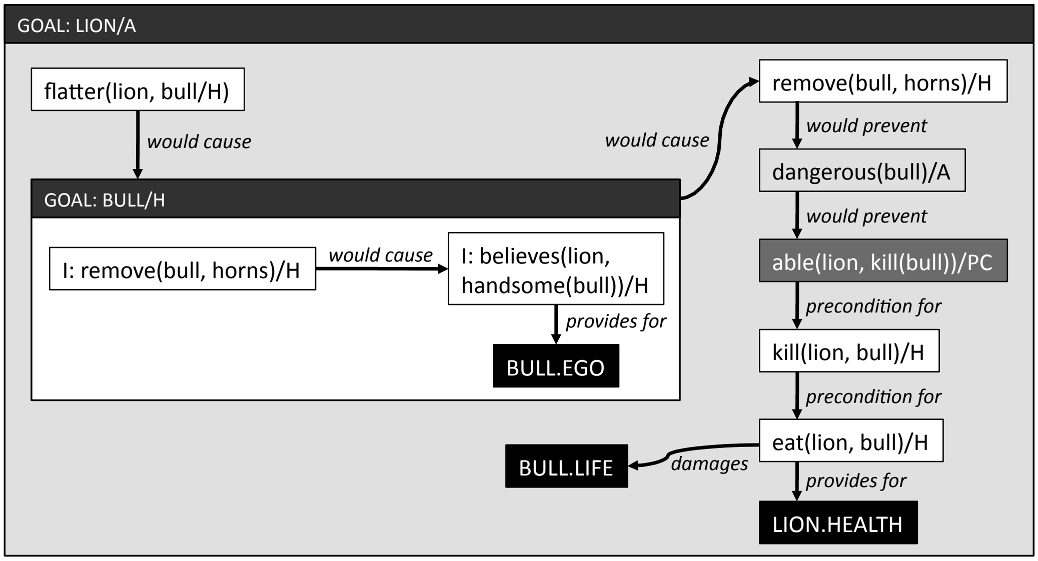


Figure 3.17: Encoding showing a multi-step plan with Affect nodes in "The Wily Lion".

3.3.2.9 Synthesis

We have now described all the types of nodes and relations that constitute a SIG. Figure 3.18 shows an overall SIG for "The Wily Lion" (except for the textual layer, which is given in Table 3.4, and P_8, which is omitted for brevity). State and Timeline nodes, as well as their related arcs, are not shown; as a notational convenience, we instead draw *followed by* arcs directly between P nodes when the nodes are attached to subsequent State nodes. The interpretative layer here includes nodes representing many of the features we set out to model: the lion's motivation (to eat the bull for purposes of enhancing his health), the problem blocking his goal (that the bull is dangerous due to his horns), his plan for overcoming the problem (to flatter the bull into forming a plan that involves removing his horns), his attempt at actualizing the plan (flattering compliments and pointed suggestions), and the successful outcome of the plan. The graph models the notion that the bull is the net loser in the transaction, having lost its life in an attempt to provide for its ego. A series of arcs connect textbase propositions to these meaning structures, either because they explicitly state the content of the structures, imply (entail) the content, or can be otherwise inferred to mean that the content is actualized or ceased. The textbase propositions, in turn, are mapped to discourse utterances in Table 3.4; though these *interpreted as* arcs are not drawn

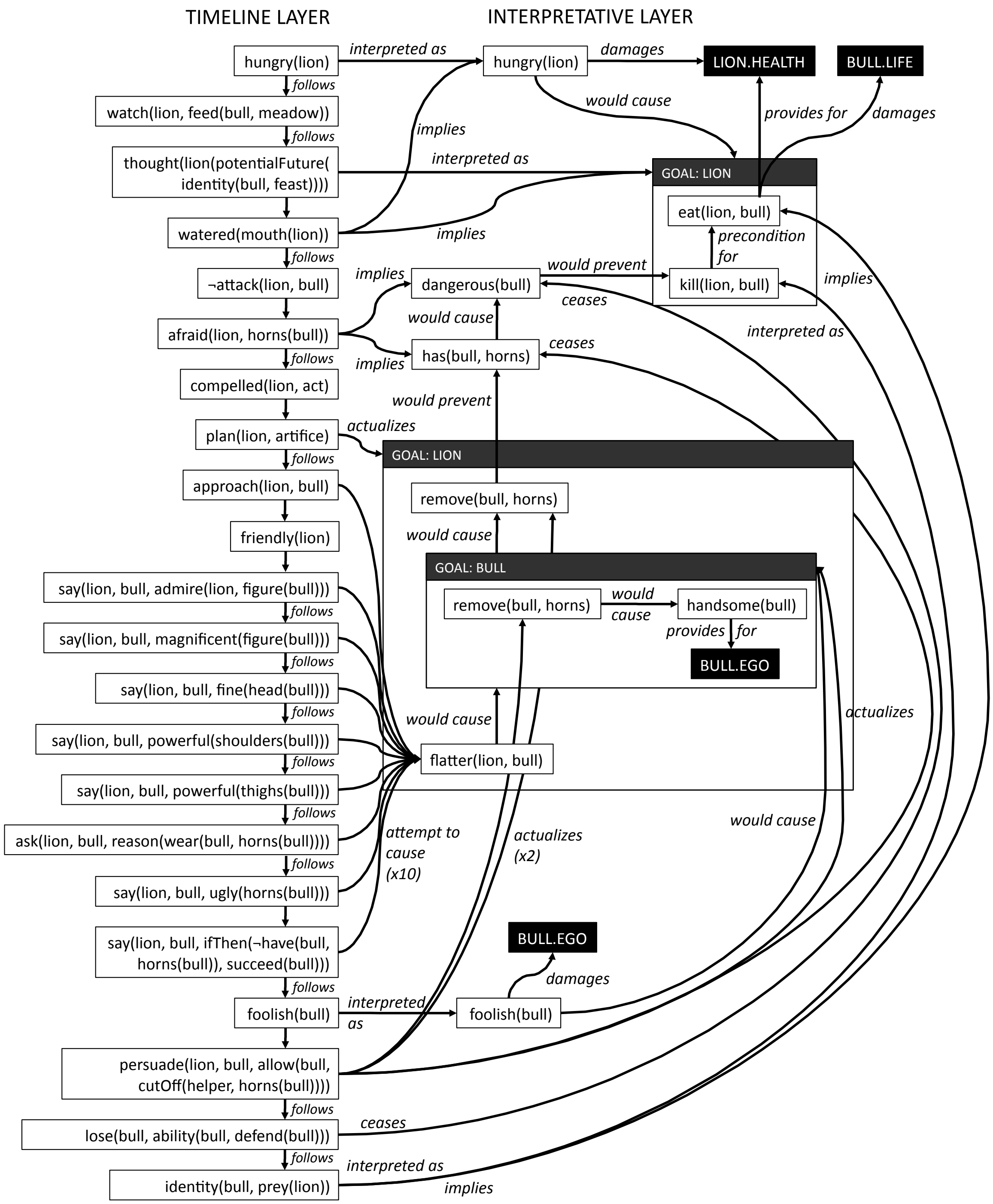


Figure 3.18: Overall encoding for "The Wily Lion" (textual layer shown in Table 3.4).

	Text (TE)	State (S)	Timeline (T)	Prop. (P)	Interpretative Prop. (I)	Belief (B)	Goal (G)	Affect (A)
TE	f			ia^+				
S		f, e^+	in					
T				r^+				
P		ba, ea			$\overrightarrow{a}$	$\overrightarrow{a}$	$\overrightarrow{a}$	
I					$\overrightarrow{b}$	in, $\overrightarrow{b}$	in, $\overrightarrow{b}$	p^+, d^+
B					$\overrightarrow{b}$	in, $\overrightarrow{b}$	in, $\overrightarrow{b}$	p^+, d^+
G					$\overrightarrow{b}$	in, $\overrightarrow{b}$	in, $\overrightarrow{b}$	p^+, d^+
A								

$$\overrightarrow{a} = \{ac^+, ap^+, ia^+, im^+, a^+, c^+\}$$
$$\overrightarrow{b} = \{wc^+, wp^+, pf^+, pa^+\}$$

Table 3.8: Valid relations between nodes in a SIG. For each adjacency between two node types, the set of legal arc types for that adjacency. See Table 3.3 for a key to the arc types and node types abbreviated here.

in Figure 3.18 due to space constraints, this mapping explicates the relationship between story time and telling time. The result is an encoding of a theory-of-mind interpretation of the fable that integrates several aspects of narrative, including agency and time, without relying on a prescriptive model of discourse structure such as a grammar.

3.3.3 Summary and Comparison to Prior Work

In this section we have described the 8 types of nodes and 18 relations that constitute the schemata of a Semantic Intention Graph. Table 3.8 summarizes the model in terms of the arc adjacencies that are defined for each possible pair of node types, with rows representing originating nodes and columns representing destination nodes. For instance, a Goal frame can relate to a Belief frame with *in*, *would cause*, *would prevent*, *precondition for* and *precondition against*. A + after an arc type indicates that more than one outgoing arc for the type is legal; otherwise, at most one outgoing arc is permitted.

We see the SIG as a next step in the evolution of narrative discourse models. For comparison, we have given four diagrams of “The Wily Lion” using different representations: Trabasso’s causal network formalism (Figure 3.2), Mandler and Johnson’s grammar (Figure 3.3), Lehnert’s plot units (Figure 3.5) and finally the SIG (Table 3.4 and Figure 3.18). The features of the SIG overlap with those of each model, but as a whole, it is a novel approach to diagramming narratives.

	GRTN	M/J Grammar	Plot Units	SIG
Purpose	Cognitive modeling	Cognitive modeling, Discourse	AI understanding, story summarization	Discourse (corpus analysis; NL understanding)
Structure	Semantic network	Parse tree	Semantic network	Semantic network
Approach	Descriptive, but inflexible	Prescriptive	Descriptive	Descriptive
Input	Textbase propositions	Discourse units	Event model	Discourse units
Implied content	None (organizes textbase)	None	Yes	Yes
Time	Propositions organized into temporal chains	No semantic model of time	Events organized into temporal chains	Interval-based timelines mapped to discourse units
SRL/WSD compatible	Yes (textbase propositions)	No	No	Yes (propositions, discourse input)
Goals	Explicit (G nodes)	Explicit (GOAL)	Explicit & Implicit (M nodes)	Explicit & Implicit (G frames)
Plans	Subgoals (G→G)	Subgoals (nested GOAL_PATHs)	“Motivation” units (M→m→M)	Network of subgoals (*wc*, *wp*, *pf*, *pa*)
Beliefs	Explicit (R nodes)	Explicit (Internal Event)	Explicit & Implicit (M nodes)	Explicit & Implicit (B frames)
Attempts	Explicit (A nodes)	Explicit (ATTEMPT)	None	Explicit & Implicit (*ac*, *ap*)
Outcomes	Explicit (O nodes)	Explicit (OUTCOME)	Explicit & Implicit ($a \rightarrow +$, $a \rightarrow -$)	Explicit & Implicit (actualization status)
Affect	Explicit (SO vs. AO outcomes)	None	Complete (+ and –)	Complete (Affect nodes; goal closure)
Theory of Mind	Single-character POV	None	Multiple broad domains	Nestable agency frames
Implementation	Machine simulation; cognitive experiments	Cognitive experiments	AI system, cognitive experiments	Software platform & annotation UI; collection project

Table 3.9: Comparison between Trabasso's GRTN model, Mandler and Johnson's story grammar model, Lehnert's plot-unit model, and the SIG model.

Table 3.9 outlines the way the four models differ in terms of their purpose and design. The interpretative layer of the SIG resembles plot units: In both cases, one can enumerate a set of small, canonical "subgraphs" that represent thematic elements (such as loss) and chain them together to form arbitrarily large and complex structures. However, SIGs address many of the shortcomings of plot units by including a representation of time, a connection to the original discourse, and a more expressive representation of outcomes and mental states. Like GRTNs and grammars, SIGs show the connections between the functional components of a discourse—diagrams such as Figure 3.18 can connect any two timeline propositions, and by extension discourse clauses, that both relate to the same interpretative node (either directly or through closure rules). However, the SIG is more expressive than either of these models, as each imposes a rigid structure that excludes what we would consider to be thematically rich stories. For instance, it is difficult in either case to model two-agent interactions where mutual beliefs are important; agency frames allow us to represent such details. Overall, the SIG captures more thematic content in a narrative discourse than any of these four models, with respect to a theory-of-mind reading of a text.

3.4 Conclusion

This chapter has introduced the Story Intention Graph (SIG) as a closed set of discourse relations that collectively represent aspects of an agentive reading of narrative discourse. We reviewed four prior approaches and found them to have a limited expressive range, especially in representing elements of agency that research in cognition has shown to be key to narrative comprehension (agentive goals, plans, beliefs and attempts). Building on this prior work, we described the SIG schemata as emphasizing these and other facets that differentiate a story from an expository text or a set of disassociated facts. We do not claim that every discourse must be interpretable as a story, or that every story must feature easily discernible goals; indeed, one can find selections of modernist fiction that eschew both of these conventions. Rather, we claim that the SIG is an expressive, yet formal model for representing thematic content in the volumes of narrative discourse which employ the devices of time, mode and agency.

Let us conclude this chapter by revisiting once more the citation from E. M. Forster about the difference between a non-story and a story:

1. The king died. Then the queen died.
2. The king died. Then the queen died of grief.

Forster's point is that a set of timeline-ordered events is not necessarily a narrative. There must be other inter-sentential relations that bind the discourse together. Causality and motivation are the underpinnings of our approach, and in the case of (2), they relate the second sentence to the first.

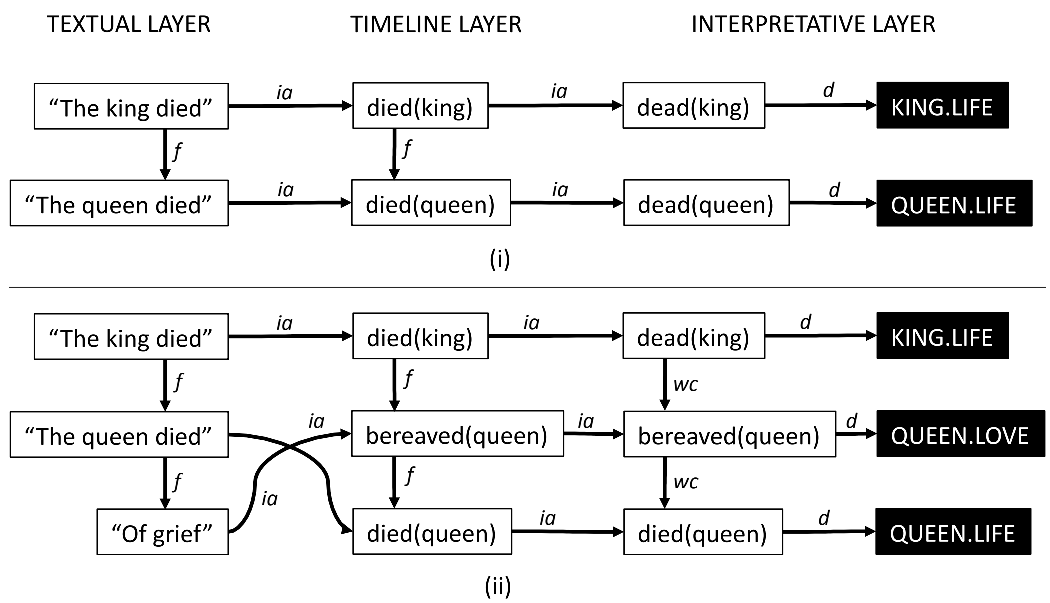


Figure 3.19: SIG encoding of Forster's distinction between a non-story (top) and a story.

Figure 3.19 illustrates this effect graphically. SIG encodings are drawn for both (1) and (2) in 3.19(i) and 3.19(ii), respectively. In both cases, the textual layer contains the original discourse clauses, and the timeline layer dutifully constructs a timeline of sequential events. But only the second version provides a rationale for relating the two deaths as a causally coherent whole, and the interpretative layer provides the means to represent the difference.

On a broader scale, we may also consider how SIGs rate according to the criteria for a new representation we defined at the beginning of this chapter: expressiveness, robustness, computability and accessibility.

First, is the SIG schemata expressive enough to have wide coverage over the range of what a reasonable reader would consider to be thematically rich narratives? We believe that SIGs are highly expressive due to the "open-ended" nature of the graph architecture. We have defined a set of relations that can be instantiated and combined in patterns to reflect an extensible range of thematic content, just as a closed set of words in a lexicon can be combined in syntactic patterns to form a far larger set of possible sentences. The relations serve as common building blocks for a range of narrative situations congruent with a theory-of-mind reading of a text (such as revenge, deception, success, failure, and regret) as well as formal storytelling devices (flashbacks, point of view, mystery, and so on). Because new stories are constantly being told, recombined as they may be from previously told stories, we cannot prove by exhaustion that the SIG is sufficiently expressive to cover every possible discourse that a reasonable reader would consider narrative in nature. However, we can show that the SIG is more expressive than previous descriptive models we have considered, and by enumerating a set of SIG fragments that model a wide-ranging set of narrative scenarios, we can demonstrate the framework of a wide expressive range by example. We take up this task at length in Appendix B.

Second, is a SIG robust, so that it gracefully handles varying degrees of semantic precision? Yes—as we have designed it in such a way that partial encodings of a story are permissible. Being a descriptive formalism, the SIG allows multiple levels of abstraction. There is no prescribed number of nodes that must be instantiated, though we will discuss guidelines for human annotation in Chapter 4. One can, for instance, forego a detailed discussion of a multi-step plan and label all of the actions by an agent as an "attempt to cause" a positive or negative affect state. (The wily lion, in such a flat reading, did everything in an attempt to fulfill a one-step plan to improve his health.) The formalism is also agnostic to the type of representation associated with each Proposition (P), Interpretative Proposition (I) and Affect (A) node—one can combine the SIG relations with any type of sentential knowledge representation (such as propositions) using I and P nodes as containers, or assign nothing to them except for agent metadata. In the latter case, the relations still describe the thematic aspects of a highly abstract story, such as one about overcoming adversity:

> "Agent X wanted to do A because he thought it would help him do B. Agent Y tried to prevent X from doing A. In the end, Agent Y prevented Agent X from doing A. But Agent X did B through other means."

This aspect of the SIG also gives it a domain independence, in that no particular predicate vocabulary is defined to be part of the model; although we focus on Aesop's fables for their brevity, we will soon apply the model to other genres and longer stories. We take up the question of whether the schemata is *computable* in Chapter 5.

The final questions is that of the *accessibility* of our approach with respect to trained annotators. We will explore this in the following chapter, in which we implement a software platform and annotation interface for creating a DramaBank of story encodings.

Chapter 4

Scheherazade

A set of proposed discourse relations such as the SIG is more useful when implemented as a machine-readable markup scheme and applied to a corpus with automatic or manual annotation. In Chapter 3, we developed a novel set of relations for representing the temporal, causal, affectual and goal-oriented features of a narrative discourse. In this chapter, we describe the implementation of a software package that facilitates story annotation, representation and management, using the SIG formalism as the basis of its data structure.

The system we have built, Scheherazade, meets six design goals:

1. Well-formed SIG encodings: Scheherazade allows a user to interactively build and construct encodings so that the semantics are enforced (for example, that *provides* and *damages* arcs can only point to Affect nodes).

2. Computability: The system is able to perform inference on SIG encodings according to the logical entailment rules that we outline in Chapter 3 and Appendix C, such as tracing whether an interpretative state is Actualized, Prevented/Ceased or Hypothetical, and inferring indirect causes based on causal chains.

3. Scalable precision: Scheherazade allows annotators to build propositional equivalents of story clauses and sentences. Specifically, it provides a process for encoding predicate-argument structures that leverage the taxonomies of nouns and verb frames found in external linguistic resources.

4. Domain independence: We strove to avoid over-committing the knowledge base (or the discourse model) to a particular set of narratives or a particular narrative genre.

5. Extensibility: We built the system as a platform for story management, so that other discourse relations could be applied to the same text by means of an API. The API also allows external learning tools to extract features from encodings.

6. Accessibility: We built an interactive, graphical user interface so that trained annotators can construct encodings from source texts. This includes a feedback text generator that serializes the encodings back into surface text, for purposes of allowing annotators to check whether the system has correctly captured the intended meaning of the story.

The following sections provide details regarding the design and implementation of the system. We describe the system architecture in Section 4.1 and the core logic in Section 4.2. Section 4.3 delves into the graphical interface we have developed for community story annotation. We then describe in Section 4.4 the textual generation module which "reverses" the annotation process by synthesizing a discourse from an encoding. We conclude in Section 4.5, leaving the experimental collection project to Chapter 5.

4.1 Data Structure and Architecture

The data structure for a narrative in SCHEHERAZADE mirrors the formal description of a SIG we gave in Chapter 3. Specifically, we use a graph structure where nodes represent elements such as spans of surface discourse, states in time, goals of characters, actions that occur in the timeline, and so on. The content of P and I nodes can either be placeholder content, where only the agent is indicated, or a more complete **propositional modeling** consisting of predicate-argument structures tied to a series of external resources. For instance, instead of placing "John walked to the store" in a node, we place `walk(person1, store1)` where `walk(<agent>, <destination>)` is a verb frame from a formal taxonomy of such frames, and `person1` and `store1` invoke instances of a `man` and `store` respectively (which are noun types from another formal taxonomy).

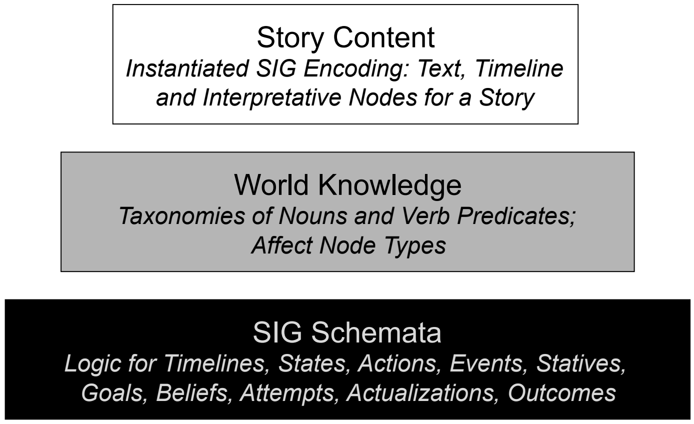


Figure 4.1: Three classes of data are distinguished by SCHEHERAZADE, each of which applies the one that appears beneath.

The "optional" nature of propositional modeling allows us to encode precise information about events and statives when possible, but still have a well-formed SIG encoding otherwise. On one hand, state-of-the-art semantic parsing tools are currently unable to automatically convert a text into a sequence of predicate-argument structures with high accuracy, and in our formative evaluations, even trained human users can find the same task challenging. On the other hand, when propositional modeling is present in an encoding, it is a rich source of structured data with which we might algorithmically find similarities and analogies between stories. SCHEHERAZADE supports either approach, and in Chapter 5 we explore the nature of the trade-off in detail by collecting corpora of story encodings under both conditions and comparing the results.

The system distinguishes between three classes of data (Figure 4.1):

Narrative semantics. This class includes the definitions and logical rules of the SIG schemata as we defined it in Chapter 3. It is a "hard" constraint in the data structure, in that it is immutable over all stories and all domains. SCHEHERAZADE enforces the rules of the SIG and returns an error to the user if an illegal change is requested during the annotation process. For example, there must always be a Reality timeline and zero or more alternate timelines, and circular plans are illegal.

World knowledge refers to the particular verb frames (predicates), noun types, and other facets of linguistic knowledge which are available for propositional modeling during the annotation process. It also includes a list of legal Affect types, such as `FREEDOM` and `EGO`, as we discussed in Section 3.3.2.8. For instance, there might be a `say` predicate encoded in this layer that, like a frame [Minsky, 1975], is known to take two type-restricted arguments with particular thematic roles (a conscious entity as a speaker in the Agent role, and an object as an intended hearer in the Experiencer role). This is a "soft" constraint, in that the system can be configured prior to the annotation process with a supply of **predicate frames**, noun types, selectional restrictions and Affect types that are to exist in the story-world. In the case of `say`, it would then disallow a non-organism from serving as an Agent. Similarly, a configuration of world knowledge geared toward Aesop's fables need not include out-of-domain knowledge such as `American`. We will soon describe a default knowledge base that we have compiled for our experiments involving some 200,000 noun and verb elements.

Story content includes the content of a particular encoding of a source text. Propositions, alternate timelines, goals, plans, beliefs and Affect nodes, once instantiated in an iterative annotation process, are linked back to nodes that represent spans of source text (the textual layer). The content must follow the constraints of both narrative semantics and world knowledge.

The result is a data structure that includes the textual layer, timeline layer and interpretative layer of a complete SIG encoding. Figure 4.2 illustrates the way that the three classes of knowledge interact. Narrative semantics, world knowledge and story content are visualized as black, grey, and white shapes, respectively. Nouns and predicate frames are first stored as world knowledge (in grey), then instantiated (in white) and linked to particular clauses in the source text.

As an implementation of a descriptive model, rather than a prescriptive model, the system does not try to understand whether the content of any action or stative makes logical "sense" in the context of the story. For example, it will allow a character to act even if the "die" predicate is applied to the same character at a previous story state. However, as we explore in Chapter 5, it will leverage the structure found in the external taxonomies to find semantic similarities between story propositions.

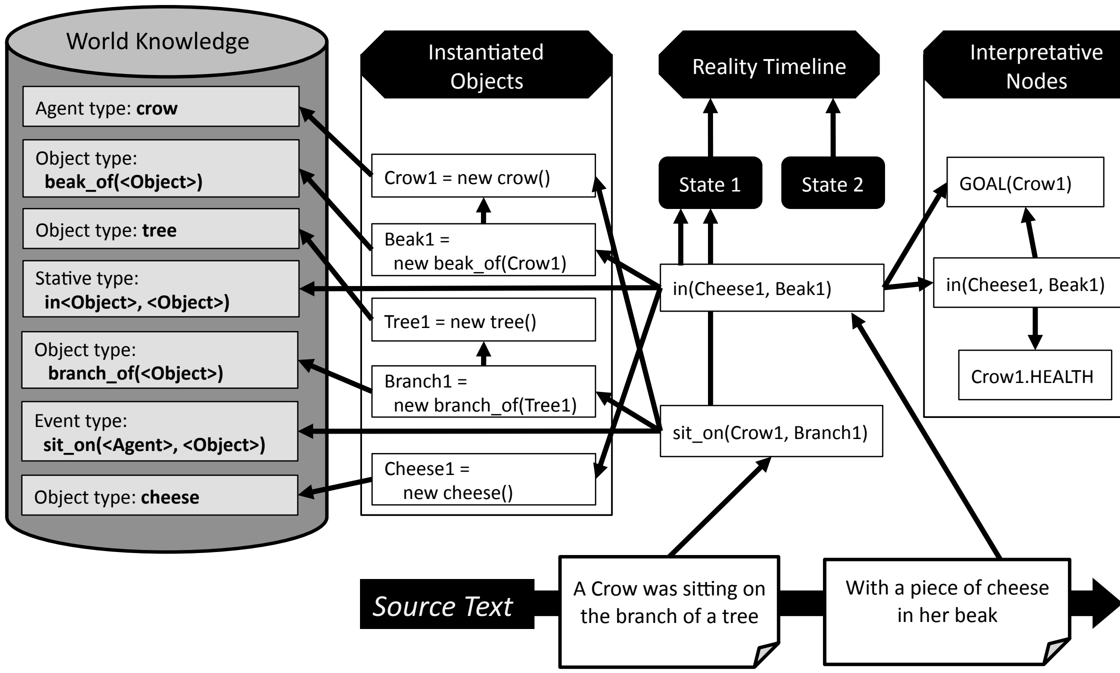


Figure 4.2: The SCHEHERAZADE data structure as applied to “The Fox and the Crow”.

System Architecture

The process of symbolically encoding a story in SCHEHERAZADE begins with the loading of a text file containing the source text. A user then repeatedly sends various commands to a command interpreter to build up the data structure—instructions to establish a new alternate timeline, add a new State node, instantiate a new proposition with a certain predicate frame and set of arguments, link a Proposition node to a plan with an *actualizes* arc, and so on. The user receives acknowledgment that each command has been carried out, and that the encoding remains valid on both the structural and content levels.

An architectural overview of SCHEHERAZADE in shown in Figure 4.3. At the bottom of the stack is a general-purpose engine for managing semantic networks. This engine can be configured to accept arbitrary types of nodes, arcs and attachment/inference rules. It also includes a serializer for saving networks to disk and a parser for reading them back into memory. A separate module, the Story Logic Manager (SLM), applies the particular logical form of the SIG network structure, including node types, arc types and the rules governing

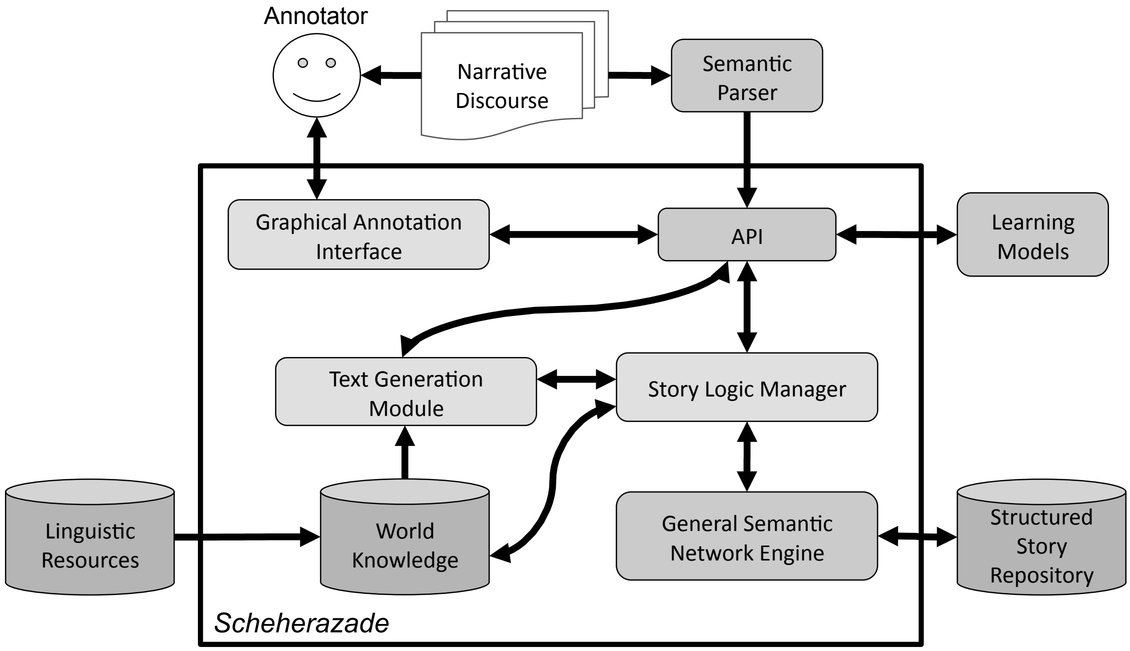


Figure 4.3: Scheherazade architecture.

their inter-relationships. The SLM also incorporates the default knowledge base we have compiled from external linguistic resources.

The Story Logic Manager is exposed to plug-ins and third-party tools by means of an API. Our intention is for future work to use Scheherazade as a foundational platform for work in narrative analysis. In Chapter 5, we use the API to extract features of SIG encodings for purposes of finding similarities and analogies between story encodings.

Finally, we have built a graphical annotation interface on top of the API, iterating the design over the course of several formative evaluations. This involves a separate text generation module. As we have mentioned, this module "reverses" the encoding process, serializing a story encoding into text in order to provide annotators with helpful feedback.

The following sections go into more detail about each of these components.

4.2 Semantic Network Engine and Story Logic Manager

At the core of Scheherazade is a general-purpose knowledge representation engine that allows for the iterative construction of semantic networks, including constraint satisfaction

Parameterized Rule Name	Meaning
circularLinksAllowed	$f(a, b) \wedge f(b, c) \wedge f(c, a)$ is legal
reflexiveLinksAllowed	$f(a, a)$ is legal
cannotRemoveIfLinkedTo	If $f(a, b)$, b cannot be deleted
cannotRemoveIfLinks	If $f(b, a)$, a cannot be deleted
fromNodesInheritFromSubtypes	$f(a, b) \wedge isa(a, c) \vdash f(c, b)$
fromNodesInheritFromSupertypes	$f(a, b) \wedge isa(c, a) \vdash f(c, b)$
toNodesInheritFromSubtypes	$f(a, b) \wedge isa(b, c) \vdash f(a, c)$
toNodesInheritFromSupertypes	$f(a, b) \wedge isa(c, b) \vdash f(a, c)$
multipleBackwardLinks	$f(a, b) \wedge f(c, b)$ is legal
multipleForwardLinks	$f(a, b) \wedge f(a, c)$ is legal
validLeftType	$f(a, b) \wedge isType(a, t)$ is legal
validRightType	$f(a, b) \wedge isType(b, t)$ is legal
mustHaveSameType	$f(a, b) \wedge isType(a, t) \wedge \neg isType(b, t)$ is legal

Table 4.1: Rules for entailment, deletion, and typing parameterized by the semantic network engine for each arc type (function *f(a, b)*).

and first-order logic entailments. Each node contains a frame type or frame instance, and each arc represents a first-order relation. The engine functions both as a tool for expressing logical relations and as a robust database for storing world knowledge and story content on the order of several hundred thousand connected nodes [Elson and McKeown, 2009].

We developed this engine to be separate from the Story Logic Manager; it has customizable data types and inference rules so that one can use it for purposes other than story logic. For instance, an API allows users to customize rules for entailment, deletion and typing, which it will then enforce as the network is built, modified, saved to disk and loaded from disk (Table 4.1). We have released the engine, along with the rest of the Scheherazade library, as a public resource.[1]

The Story Logic Manager (SLM) rests above the semantic network engine and imbues it with the particular logical constraints of the SIG: the types of nodes found in the three layers, how timelines must be structured, the rules for determining whether an interpretative-layer node is actualized at some point in the Reality timeline, and so on. The SLM provides an API for higher-level tools, using Scheherazade as a software library, to construct, store and load encodings. A subset of the commands offered by the API is shown in Table

[1] `http://www.cs.columbia.edu/~delson`

Construction	Destruction	Retrieval & Analysis
assignEvent	modifyEventTime	findStoryIntersections
assignModifier	modifyAssociatedText	getEventsBeginningAt
assignStative	modifyEvent	getEventsEndingAt
assignInterpretativeNode	modifyModifier	getEvents
defineEventFrame	modifyStative	getStatives
defineModifierFrame	removeEvent	getInterpretativeCausalChains
defineStativeFrame	removeModifier	getInterpretativeNodes
linkInterpretativeNodes	removeStativearc	getLinkedInterpElements
defineNoun	redo/undo	getDefinedNouns
newAlternateTimeline	revert	queryForPattern

Figure 4.4: A subset of the commands offered by the Story Logic Manager's API.

4.4. For each command, the SLM raises an error if the request is invalid (e.g., a user tries to delete a non-existent node); there are particular errors if the request violates the SIG schemata (such as when one attempts to create a cyclical plan). It then interprets each command into a sequence of operations for the semantic network engine to carry out.

The commands listed in Table 4.4 involve the definition and assignment of four types of world knowledge. "Definition" commands augment the world knowledge structure with a new frame or noun type, while "assignment" commands instantiate a frame or type into an instance node and add the result to the network as story content. For instance, we may first define `person` as an organism and `store` as a location, then `walk(<agent>, <destination>)` as a verb frame. We may then assign `john` as a particular person, `countryStore` as a particular store, and `walk(john, countryStore)` as a timeline event in which John walks to a country store. The SLM enforces these selectional restrictions, finding that an organism is a satisfactory agent and a location is a satisfactory destination.

In general, the world knowledge structure involves four taxonomies, one for each facet of linguistic knowledge available for composing propositions:

1. **Nouns.** We identify five classes of nouns:
    - *character*, an animate being capable of agency
    - *location*, a relative or absolute spatial placement
    - *prop*, a non-agentive physical object
    - *activity*, a behavior such as a gathering or performance
    - *quality*, an attribute such as "handsomeness" or "height"

There is a hierarchical taxonomy for each class. For instance: `monkey` IS-A `primate`, a type of character.

2. **Statives.** Statives are predicate frames that represent *durative* properties of nouns, which apply over a span of time but do not transform the story-world from one state to another. Adjectival descriptions (such as `happy(<character>)`) are statives, as are abilities, amounts, comparisons, beliefs, fears, goals, plans, hopes, identities ("John was the masked assailant"), obligations, possessions, and positional relationships ("the book was on the table").

3. **Events.** Events are predicate frames that represent state-changing actions and happenings. The frame slots can be filled with nouns, as in `walk` taking an agent and a destination, or with nested propositions. An example of the latter would be `says(<character>, <stative>)`, which represents the statement of some stative by a character ("John said that he was hungry").

4. **Modifiers.** Modifier frames reference other propositions. `slowly(<action>)`, for example, fills its slot with a reference to an action that is happening slowly.

For purposes of our experiments in Chapter 5, we turned to external linguistic resources to populate these taxonomies. Nouns, statives and modifiers adapt WordNet [Fellbaum, 1998], a well-established lexicon that features thousands of words organized into *synsets* with the same meaning. One synset, for example, includes the nouns "meadow" and "hayfield" in their typical senses. Synsets are organized into hypernym trees, with each synset related to more and less specific synsets. As our knowledge model also involves hierarchical taxonomies of nouns, we needed only to decide which subtrees of the root noun synset (*entity*) were to be adapted for each SCHEHERAZADE noun type. For example, to populate our list of available character types we adapted the *organism* subtree, allowing users to model stories concerning thousands of animal species or roles such as `traveler`. Table 4.2 lists the roots of the subtrees adapted for each noun class. Overall, we adapted 29 WordNet subtrees, including approximately 125,000 total noun lexemes (including multiple instances of nouns that appear in more than one adapted synset).

Class	WordNet Synsets Adapted	Lexemes Imported
Character	"organism"/1, "imaginary being"/1, "spiritual being"/1, "organization"/1, "social group"/1 EXCEPT "social gathering"	40,877
Location	"geographical area"/1, "area"/5, "body of water"/1, "structure"/1, "geological formation"/1, "location"/1, "land"/1-2, "land"/4, "position"/1	15,389
Prop	"artifact"/1, "plant"/2, "substance"/1, "substance"/7, "body substance"/1, "plant part"/1, "body part"/1-3, "currency"/1	48,087
Activity	"activity"/1, "social gathering"/1	5,659
Quality	"attribute"/2, "ability"/2	14,676
Total Nouns	**29**	**124,688**
Statives	**All adjectives**	**29,753**
Modifiers	**All adverbs**	**3,046**

Table 4.2: WordNet synsets (as *sense key/sense numbers*) which served as the roots of the subtrees of the WordNet hyponymy-based lexical hierarchy that we used for each of the five SCHEHERAZADE noun type taxonomies; we also adapted adjectives as statives, and adverbs as modifiers.

WordNet's adjectives, meanwhile, serve as adjectival statives that describe people or things ("the king was mighty"); its adverbs provide the basis for modifiers ("the king apologized graciously"). Other modifiers were hand-authored to serve as connectives between the modified proposition and a "third party" proposition: "A because/despite/in order to B." We implemented a routine that allows the user to either model a new proposition for the connected B clause, or invoke (plug in) a proposition that occurred elsewhere on the Reality timeline. In other words, a "nested proposition" that serves as an argument can be a reference to an actual event that occurred elsewhere in the story—"The king died. The queen died because the king had died."

While WordNet provides a hypernym tree for verbs as well, there is limited information about the manner in which each verb can be used as a predicate. For thematic roles, selectional restrictions, and syntactic descriptions, we turned to VerbNet [Kipper *et al.*, 2006], the largest online verb lexicon currently available for English. Each verb is annotated with thematic roles (arguments) and their selectional restrictions, as well as syntactic frames for the various sentence constructions in which the verb and its roles might appear. Figure

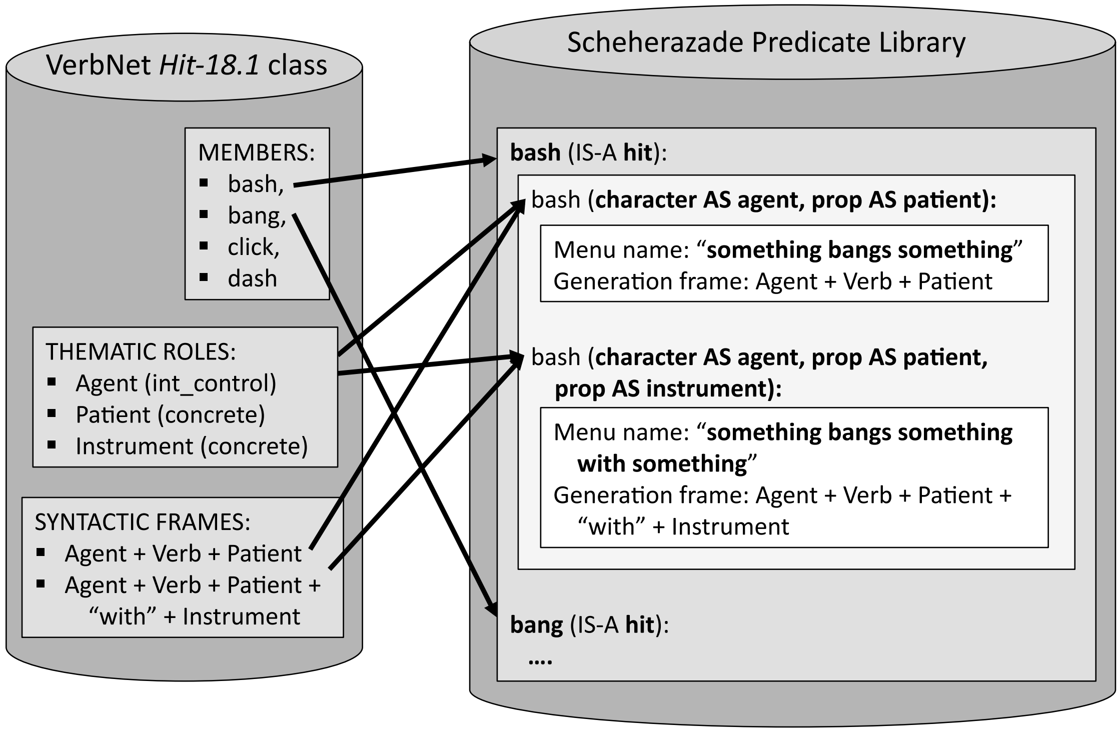


Figure 4.5: An example of the procedure by which VerbNet records are adapted to serve as predicate frames, including thematic roles with selectional restrictions and syntactic constructions.

4.5 illustrates through example the adaption of a VerbNet record into SCHEHERAZADE world knowledge. For each lexeme that serves as a member of the VerbNet class, we:

1. Find the corresponding WordNet synset for the member, which allows us to arrange predicates into an IS-A hierarchy (e.g., `bash` IS-A `hit`);
2. Map each VerbNet thematic role into a SCHEHERAZADE thematic role (such as a restriction to a Patient becoming a restriction to a prop noun);
3. Generate an easily readable phrase to represent the predicate in the user interface (such as "something bashes something with something"); and
4. Map each VerbNet syntactic frame into a distinct predicate frame. A syntactic frame may include only some, not all, of the thematic roles defined by the WordNet record. The syntactic construction that VerbNet gives is adapted into a plan for our textual generation module (Section 4.4).

VerbNet Thematic Role <u>or</u> Selectional Restriction	VN Syntax Restriction	SCHEHERAZADE Slot Restriction	Example
Agent, Experiencer, Recipient, Product <u>or</u> Animal, Comestible, Animate, Human, Int_control	NOT adv_loc	Character	*John* went walking.
Theme, Stimulus, Destination, Location, Cause	Any	Character or Prop	I walked to *the visitor*.
Location, Destination, Source, Stimulus <u>or</u> Concrete, Location	Any	Location	I set out from *home*.
Material, Patient, Product, Instrument, Destination, Asset <u>or</u> Currency	Any	Prop	He put the book on *the shelf*.
Organization	Any	Behavior	She volunteered at *the county fair*.
Patient, Stimulus, Theme, Product, Topic	NOT concrete	Quality	He spoke of *her beauty*.

Table 4.3: Mappings from VerbNet thematic roles, selectional restrictions and syntactic restrictions to SCHEHERAZADE slot restrictions. Only the mappings for noun slots are shown.

We implemented SCHEHERAZADE so that a frame's thematic role can carry a "slot restriction" of either a noun, a nested proposition (an inner event, stative or modifier) or a reference to an alternate timeline. The exact mappings with which we populated slot restrictions with the information from VerbNet records are given in Tables 4.3 and 4.4. For each table, a set of VerbNet thematic roles and selectional restrictions are given in the first column, and a set of syntactic restrictions are given in the second column. If a VerbNet slot featured at least one of the thematic roles <u>or</u> one of the selectional restrictions, <u>and</u> satisfied the syntactic restrictions (if any), it was mapped into the SCHEHERAZADE slot with the restriction given in the third column. Table 4.3 gives the mappings for noun restrictions, while Table 4.4 gives those for nested-proposition and timeline restrictions. In the latter case, slot restrictions carry a grammatical component: A SCHEHERAZADE frame may call for a nested stative in the *assertive* mode ("she told him *that he was gracious*") or in the *imperative* mode ("she told him *to be gracious"*), among others.

In the cases of *control verbs*, an argument argument of the nested proposition (that is, the subordinate clause) is controlled by the encapsulating frame (the main clause). The system recognizes these cases from its external lexicons, and when such a verb is instantiated,

VerbNet Thematic Role <u>or</u> Selectional Restriction	VN Syntax Restriction	Scheherazade Slot Restriction	Example
Theme, Topic, Cause, Stimulus <u>or</u> Communication	that_comp	Stative (assertive)	She told him *that he was gracious.*
Proposition	that_comp	Stative (imperative) or Timeline (imperative)	She ordered *that he be gracious.*
Proposition	NOT that_comp	Stative (imperative, control) or Timeline (imperative, control)	She urged him *to be gracious.*
Theme, Stimulus <u>or</u> Communication	that_comp	Timeline (assertive)	She told him *that the dog had eaten.*
Theme, Topic	np_tobe, np_to_inf	Event (infinitive), Stative (infinitive)	She had a desire *to swim the English Channel.*
Predicate	oc_ing	Stative (gerund, control)	She characterized him as *being studious.*
Topic	ac_ing	Stative (gerund)	She lectured about *being studious.*
Theme, Predicate, Topic	sc_to_inf, for_comp	Event (infinitive)	I needed *for her to arrive.*
Theme, Source, Topic, Proposition	be_sc_ing, oc_ing	Event (gerund, control)	He were forced into *using his savings.*
Theme, Topic, Proposition	oc_to_inf	Event (control) or Stative (control)	Lack of money forced him *to get a job.*
Proposition	oc_to_inf	Timeline (imperative)	The accident forced his wife *to get a job before April 1.*
Theme, Source, Topic, Proposition	sc_ing, np_ing, ac_ing	Event (gerund)	He relied on *her arriving before midnight.*
Topic	wh_comp	Timeline (whether)	He asked her *whether she had broken the lamp.*
Any	how_extract	Event (how, instructional), Stative (how, instructional) or Timeline (how, instructional)	I discovered *how to do it.*
Any	how_extract	Timeline (how, factual)	I discovered *how he had done it.*

Table 4.4: Mappings from VerbNet thematic roles, selectional restrictions and syntactic restrictions to Scheherazade slot restrictions. Only the mappings for slots restricting to nested propositions and references to alternate timelines are shown.

automatically populates the nested proposition with whatever argument the user indicates in the appropriate slot of the controlling verb (and forwards any subsequent changes to that argument). For instance, the verb *ordered* is a control verb in that the Agent of the subordinate clause is bound to the Experiencer of the main clause; as such, the system allows the propositional equivalent of "she ordered him *to sit down*," but not "she ordered him *for his sister to sit down*."

Using this approach, we have adapted VerbNet into a set of 20,530 predicate frames. However, the mapping from VerbNet to our own slot restrictions is incomplete, in that our system features a coarser and somewhat smaller range of selectional restrictions than what VerbNet offers. We disregard, for instance, VerbNet syntactic frames that deal with values ("the dress cost ten dollars") or amounts ("he was six feet tall"). This is an implementation detail rather than a design choice. In the future, Scheherazade can be expanded to more closely adapt the VerbNet system of selectional restrictions, adding support for values, amounts and other concepts, which would increase the expressive range of its approach to propositional modeling.

VerbNet has also been mapped to the large-scale annotation project PropBank [Kingsbury and Palmer, 2002]. There are three key differences between PropBank and this project. First, PropBank focuses on the sentence level, where a SIG encoding is a single structure with many interconnected propositions that bind the entire discourse. Second, due to the factors we have discussed, we do not impose a direct mapping from predicates in a textual discourse to propositions in the encoding. In our collection experiments, annotators either altered or consolidated text to focus on underlying story events rather than rhetoric, or skipped over propositional modeling altogether (upon request) to focus on the thematic content found in the SIG's interpretative layer. Third, PropBank fills its arguments with spans of text, such as assigning a clause to Arg0 or Arg1. Our representation disallows this use; an annotator chooses an element from a formal taxonomy to serve as an argument. For example, rather than highlight the text "the window" to serve as an argument, the annotator would select a noun instantiated from the appropriate WordNet *window* synset. If no formal symbol can be found for an argument, the annotator "rephrases" the proposition as best as he or she can, and still relates it to the equivalent span of source text.

4.3 Graphical Annotation Interface

To evaluate the accessibility of the SIG model, as well as to collect a corpus of SIG encodings, we ran several collection projects in which we charged subjects with the task of taking a source text and constructing a SIG encoding using SCHEHERAZADE. These subjects were undergraduate or graduate students at Columbia who were not experts in either computer science or linguistics (although they included some literature experts). This necessitated the development of a user-friendly graphical interface to make the process of constructing a SIG encoding, including propositional modeling, convenient and enjoyable to such users. This requirement became clear after a formative evaluation we conducted with a baseline approach in which a small set of users drew SIG encodings for a source text using a free-form vector graphics tool and a set of written guidelines. We found that the annotators sometimes drew graphs that were "malformed" with respect to logical constraints of the SIG (such as arc type/node type compatibilities). This motivated us to build a graphical user interface (GUI) that rests atop the Story Logic Module and API and makes the annotation process amenable to a interdisciplinary user community.

The design of such a user interface, one that bridges users to discourse annotation, temporal modeling, theory-of-mind interpretation and propositional modeling, presents challenges. Users must be able to access and instantiate the hundreds of thousands of frames of world knowledge, arrange them on multiple timelines (one for Reality and one for each modal context), be constrained by the logical rules governing interpretative-layer graph topology, and grasp the logical rules of interpretative node actualization. They must not only be able to construct an encoding using a point-and-click interface, but at each step, get *feedback* that confirms whether or not the system has correctly "understood" the aspect of the story being annotated or encoded. The interface must not only allow users to model stories symbolically, but guide them and provide advice.

In designing the GUI, it also became clear that users would find propositional modeling easier if the workflow featured natural language alone. Propositional form (with predicates, parentheses and lists of arguments) is not easily readable to users who are not highly trained, especially when the propositions are further imbued with temporal and intentional metadata. Through formative evaluations, we found that the approach with the highest usability was one which hides propositions from the user by means of feedback text—automatically generated textual renderings of the every aspect of the SIG data structure. We built a text generation module to meet this task. Feedback text has allowed us to implement a "natural language in, natural language out" workflow. For example:

1. The user sees an NL phrase in the source text: "A Crow was sitting on a branch of a tree."

2. The user searches through a list of frames in NL form, until she finds an appropriate frame: "Something sits on something."

3. The system asks the user to fill in the slots with NL prompts: "What sits?" "On what?" The user answers by selecting from among the instantiated nouns she has set up (the crow, the branch of the tree).

4. The successfully modeled proposition, `sits_on(crow1, branch(tree1))`, is never revealed to the user. Rather, an NL equivalent is given: "The crow sits on a branch of the tree." This may or may not be the surface form of the original clause or sentence.

5. The user evaluates the feedback text to check that the concept has been successfully encoded. If the generated feedback text is wrong in some way, the user reselects either the frame or its arguments.

6. The user situates the encoded proposition in a certain state and timeline. The feedback text adjusts the tense and aspect of the feedback text to reflect the temporal positioning of the proposition (Section 4.4): "A crow was sitting on a branch of a tree."

7. The user repeats the process for other spans of source text. The system juxtaposes proposition-level feedback text to create a continuous discourse called the **reconstructed story.** The user is able to compare the NL source story to the NL reconstructed story, and when the two are similar to her satisfaction, considers herself finished with timeline layer annotation.

> A Crow was sitting on a branch of a tree with a piece of cheese in her beak when a Fox observed her and set his wits to work to discover some way of getting the cheese.
> Coming and standing under the tree he looked up and said, "What a noble bird I see above me! Her beauty is without equal, the hue of her plumage exquisite. If only her voice is as sweet as her looks are fair, she ought without doubt to be Queen of the Birds."
> The Crow was hugely flattered by this, and just to show the Fox that she could sing she gave a loud caw. Down came the cheese, of course, and the Fox, snatching it up, said, "You have a voice, madam, I see: what you want is wits."

Table 4.5: "The Fox and the Crow".

8. A similar "NL in, NL out" approach eases the process of building the interpretative layer of the encoding.

In the remainder of this chapter, we will discuss the interface, annotation process and text generation module which implement this workflow. We will use the Aesop fable "The Fox and the Crow", shown in Table 4.5, as the subject of a running example. In Chapter 5, we will describe the collection project we have carried out which shows Scheherazade to be successful at eliciting annotations from trained annotators. We present this work as a contribution.

4.3.1 Related work

As we mentioned in Chapter 3, previous models of discourse have been accompanied by corpus collection projects which assign the task of semantic annotation to trained annotators. Several of these have dealt with stories in particular. As the SIG model overlaps with several of these projects, so too does our annotation process overlap with prior work.

The marking up of a textual corpus according to a formal model of predicate-argument structure, word sense, thematic role, time, or discourse cohesion is typically done without use of feedback text. When Carlson et al. [2001] applied the Rhetorical Structure Theory model to a large scale collection project, annotators used lexical and syntactic clues to help determine the boundaries between discourse units, then selected the most appropriate RST relations to join units together—a process that required professional language analysts with prior experience in other types of data annotation. The Penn Treebank Corpus [Marcus

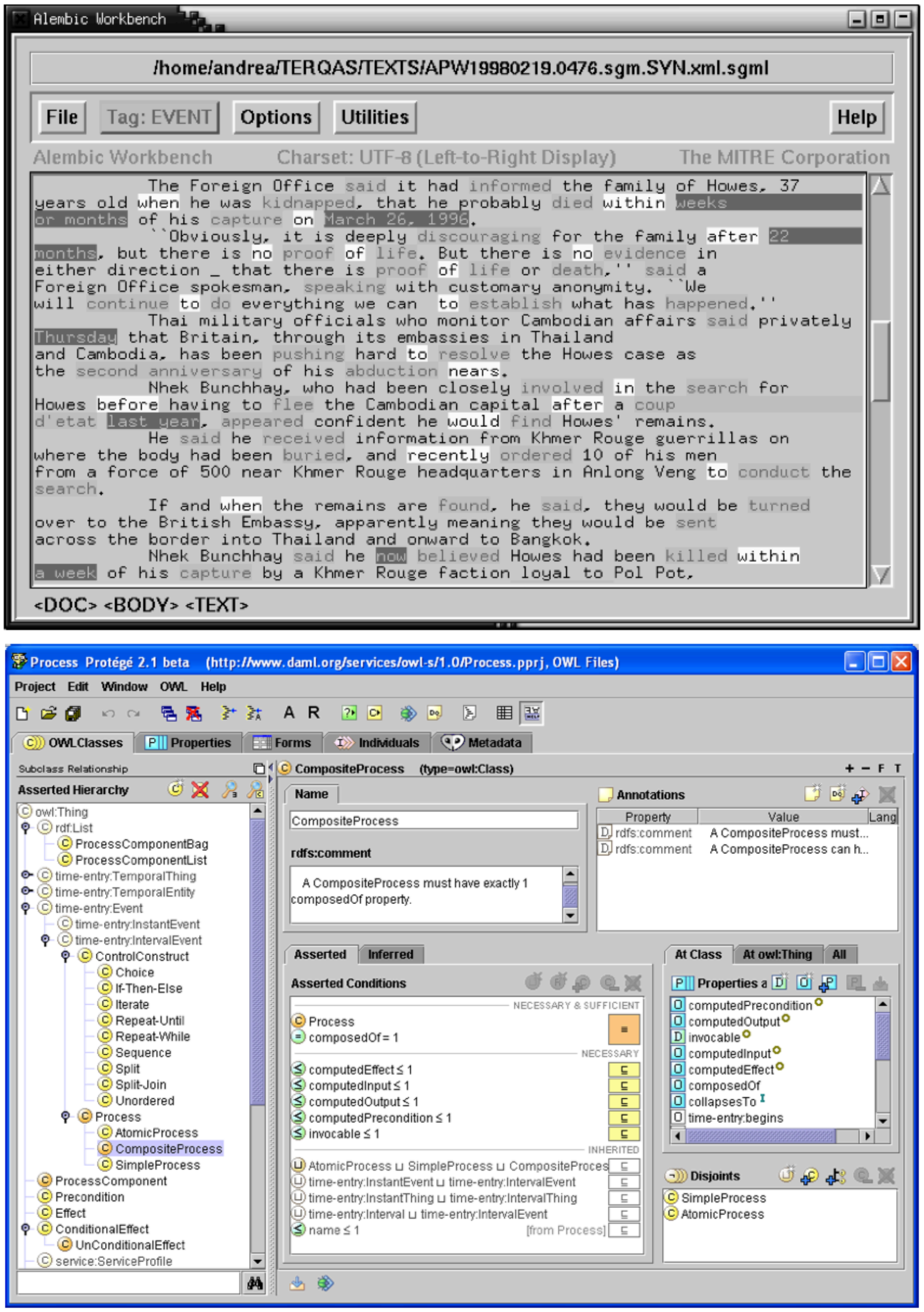


Figure 4.6: Prior annotation interfaces: Alembic Workbench (top) and Protégé/OWL.

et al., 1993] collected syntactic and part-of-speech annotations through a combination of automatic tagging and manual correction/tagging by trained annotators. Thematic role (predicate-argument) annotation was applied to the same corpus by the PropBank project [Palmer *et al.*, 2005]; this was also a direct markup of the source text, with users selecting text spans that served as arguments. The more recent Ontonotes project [Pradhan *et al.*, 2007] involves several interconnected layers of annotation, including syntax, propositional structure, word sense disambiguation, named entity classification and anaphoric coreference. The creators use the Penn Treebank and PropBank corpora, adding methods to combine fine-grained WordNet senses to arrive at 90% inter-annotator agreement on word sense disambiguation. Finally, the TimeBank corpus [Pustejovsky *et al.*, 2003b] is annotated to indicate events, times, and temporal relations. After the corpus was pre-processed using automatic tools to find likely temporal anchors, the annotators (who came from a variety of backgrounds) used a modified version of the Alembic Workbench graphical interface to mark up time expressions and connectives (Figure 4.6).

In addition to the textual markup angle, SCHEHERAZADE also lets users browse formal taxonomies of knowledge and instantiate types into instances—not only predicate frames, but nouns themselves. This process has been explored in previous projects with the aim to allow domain experts to assist in the creation of knowledge bases. A typical "knowledge capture" or "knowledge entry" system presents a knowledge-base (KB) editor which displays a taxonomy of elements, sometimes in a graphical tree or network, and allows users to add nodes and arcs to introduce new elements and new first-order relations in a point-and-click interface [Paley *et al.*, 1997; Clark *et al.*, 2001]. The most well-known and extensible knowledge capture tool, Protégé, visualizes ontologies of frames (and their instances) through a variety of graphical metaphors [Storey *et al.*, 2001]. A recent extension to Protégé is designed to ease the process of authoring ontologies for the "Semantic Web" using the W3C's Web Ontology Language (OWL) [Knublauch *et al.*, 2004]. This version of the tool, seen in Figure 4.6, allows users to define classes (frames) with slots, individuals that instantiate the classes, metadata and logical entailment rules.

The use of generated feedback text to make semantic encoding more accessible aligns SCHEHERAZADE with the WYSIWYM user-interface pattern: "What you see is what you mean." Power and Scott [1998] describe WYSIWYM as a key technique for making "symbolic authoring" tools natural, simple and self-documenting. They define a WYSIWYM system as one in which the user only sees feedback text, never the domain model itself, even in the knowledge entry process [Power *et al.*, 1998]. Biller et al. [2005] describe a WYSIWYM editor in which the representation, like ours, is a conceptual graph that incorporates the VerbNet and WordNet linguistic resources. However, compared to this work, our work is more geared toward narrative, and includes more complete representations of time, modality and agency. Power and Evans [2004] explore the effects of varying feedback text (its illocutionary force, time, polarity, modality, and modifiers) to communicate the formal properties of the entity being examined. For example, "the patient took an aspirin" implies a different formal encoding than "the patient may take an aspirin" or the imperative "take an aspirin."

Projects in the domain of *narrative* knowledge entry have typically aimed to elicit original stories, rather than annotating existing discourse. Bers [1999], for instance, elicits stories from children by using a programmable plush toy that provides encouraging feedback. Upon parsing the child's story, a back-end system finds a closely matching story in its database to tell as a reply. Other work has similarly used embodied conversational agents to engage users in conversational storytelling [Bickmore and Cassell, 1999]. As we mentioned in Section 3.2.1, Riedl et. al [2008] recently explored a methodology with which users can author stories in the QUEST formalism and have a system automatically ask relevant feedback questions such as "why did that happen?".

We see our interface as a contribution not only because it implements interpretative-layer SIG annotation—itself a novel approach to modeling discourse relations in a narrative text—but because of its wide use of the WYSIWYM technique to reflect the system's understanding of story content in feedback text.

4.3.2 Overview of annotation procedure

In the Scheherazade encoding interface, the first task for an annotator is to read the text and fully comprehend it. As a story graph is a discourse unit rather than a sentential unit, it is important for the annotator to take a holistic view of the text before beginning the annotation process. From there, the method for creating an encoding from the source text involves three tasks. The annotator does not need to complete the three tasks sequentially; one can move back and forth as needed. The tasks are:

1. **Agent, object and theme extraction.** The annotator identifies agents, objects and relevant themes and processes in the text, and represents them as *instance objects* (individuals).

2. **Propositional modeling.** The annotator builds the timeline layer of the encoding. Predicate frames and arguments are selected that best reflect the events, statives and modifiers that appear in the source text. Propositions are assigned to states and transitions on a timeline and linked to corresponding spans of source text. Modal (hypothetical) events are grouped in alternate timelines.

3. **Interpretative modeling.** The annotator builds the interpretative layer of the encoding to model his or her understanding of the overarching goals, plans and beliefs of its agents. Propositions that represent goals, plans and beliefs are modeled and placed in their appropriate agency frames (that is, either as ground truth or inside a belief frame of an agent—see Section 3.3.2). Each node is also annotated in terms of its affectual impact, if any, with respect to each agent.

After making each change to the story graph, the annotator checks the pursuant feedback text to ensure that the system has encoded the concept correctly. The process terminates once the annotator feels that he or she has encoded the story with the greatest amount of precision possible, given the formal limitations of the representation.

Figure 4.7 shows the Scheherazade GUI. (The two parts of this figure represent panels that are placed side by side in the interface, but we have rearranged them for formatting purposes.) There are three major panels to the interface that correspond to the three major annotation tasks. Minor panels, which can be summoned by the buttons along the bottom-right, allow the user to perform "housekeeping" operations such as saving and loading encodings, undoing or redoing operations, and browsing the annotation guidelines.[2]

[2]Marshall Fox wrote the component which loads the annotation guidelines from disk.

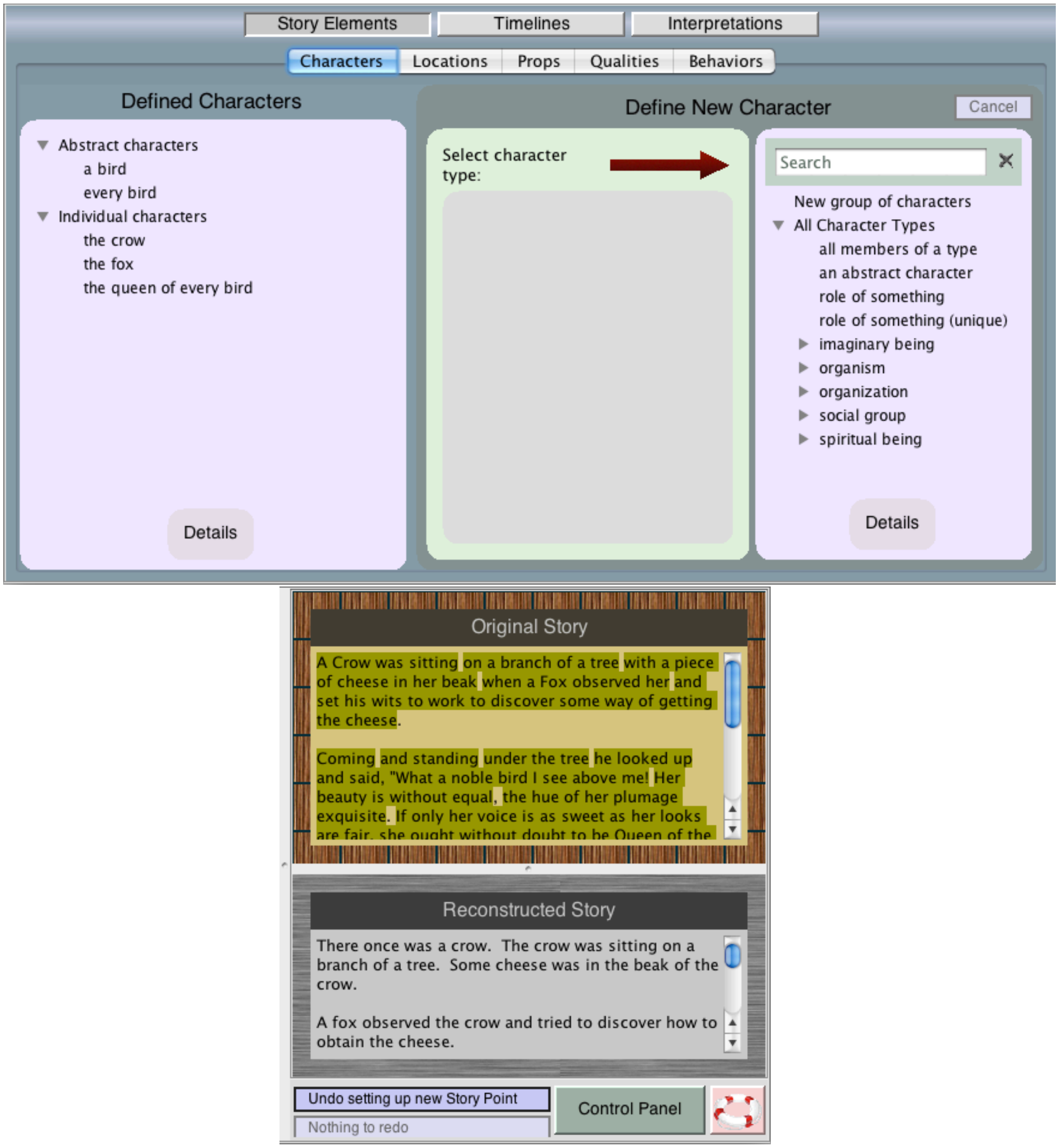


Figure 4.7: Elements (object extraction) screen of the SCHEHERAZADE interface, including the Story Elements panel (top), and the source/feedback text panels.

The three large buttons along the top of the screen bring the user to the three major panels. The *Story Elements* panel, in Figure 4.7, is used to instantiate objects, creating the story content nodes seen in the "Instantiated Objects" box in Figure 4.2. The *Timelines* panel, seen later in Figure 4.10, is used to model events, statives, and modifiers (to create the timeline content as seen in the "Reality Timeline" area of Figure 4.2). Finally, the *Interpretations* panel, seen later in Figure 4.15, provides a "canvas" on which annotators can encode an agent-centric view of the text (to draw the "Interpretative Nodes" as seen in Figure 4.2).

The smaller panels in Figure 4.7 show the source text and reconstructed story, respectively. Each of the clauses in the source text is highlighted. A highlighted clause is one that has been represented as a textual-layer (TE) node and linked to a timeline-layer node with *interpreted as*. Because all of the clauses are highlighted in this example, we can say that the presented encoding completely covers the source text. In other scenarios, annotators may leave non-narrative spans of the source text, such as news article bylines, un-annotated. Such spans remain in their original, non-highlighted appearance in Source Text panel.

4.3.3 Object and theme extraction

After reading the text, the annotator first identifies nouns from among the five classes we enumerated earlier: characters, locations, props, activities and qualities. A noun must be instantiated from a type before it can be invoked as an argument in a proposition. For instance, the first sentence of the fable at hand is "A Crow was sitting on a branch of a tree with a piece of cheese in her beak." The main event predicate, *sitting*, has two thematic roles, an agent and a destination. The agent is *a crow*, and the destination is *the branch of a tree*. We must create instance objects for these two nouns so that we can invoke them as arguments when instantiating the *sit* predicate frame. The latter clause in the source text, about the piece of cheese, is a positional stative which we can model separately once we establish *a piece of cheese* and *the crow's beak* as objects.

This system is more complex than the PropBank approach of highlighting text spans in the source text to serve as arguments. The major advantage to object extraction is that instance objects are reusable as arguments—they function as typed entities with coreferent mentions. By creating and reusing instance nodes, annotators perform coference resolution as part of the encoding process. As we saw in Chapter 3, coreference is key to an encoding that shows the cohesion of a discourse.

Figure 4.8 shows the process in the GUI for creating a new instance object for the Fox in our fable:

1. The annotator first selects the object class by clicking the appropriate tab on the top of the panel (character, location, prop, quality or behavior). Since we are creating an instance node for the fox, we select the *Character* tab.

2. The left side of the panel shows a list of instance objects that have already been created. The system allows for instance objects which are themselves abstractions within the story-world, such as the "wits" that the fox wishes upon the crow. Individuals that are concrete within the story-world are listed below abstractions.

3. The right side of the panel shows an empty **form** with a **type selector**. The selector is populated with the tens of thousands of types that we imported from WordNet, arranged hierarchically according to WordNet's hyponymy tree. A search box allows the user to type in a string and see a list of matching types, including the various types of foxes distinguished in WordNet. Immediate hyponyms are used in parentheses to allow the user to quickly disambiguate between noun senses (e.g., *fox (canine)*).

4. Once the user selects a type, the form prompts for certain metadata. An *accept button* shows generated feedback text; when the annotator clicks the button, the new character becomes fully instantiated and appears in the left-side list. The annotator can provide a gender and a name, which the feedback text generator uses to select appropriate pronouns and references, respectively.

Previous work in word sense disambiguation has shown WordNet synsets to be very fine-grained, which can hurt inter-annotator agreement [Palmer *et al.*, 2007]. For example, "fox" also represents the synset of "a shifty deceptive person;" other distinctions can be far more subtle. We address this issue in two ways. First, at the user interface level we show disambiguators for each type. The annotator sees "fox, canine, carnivore" juxtaposed

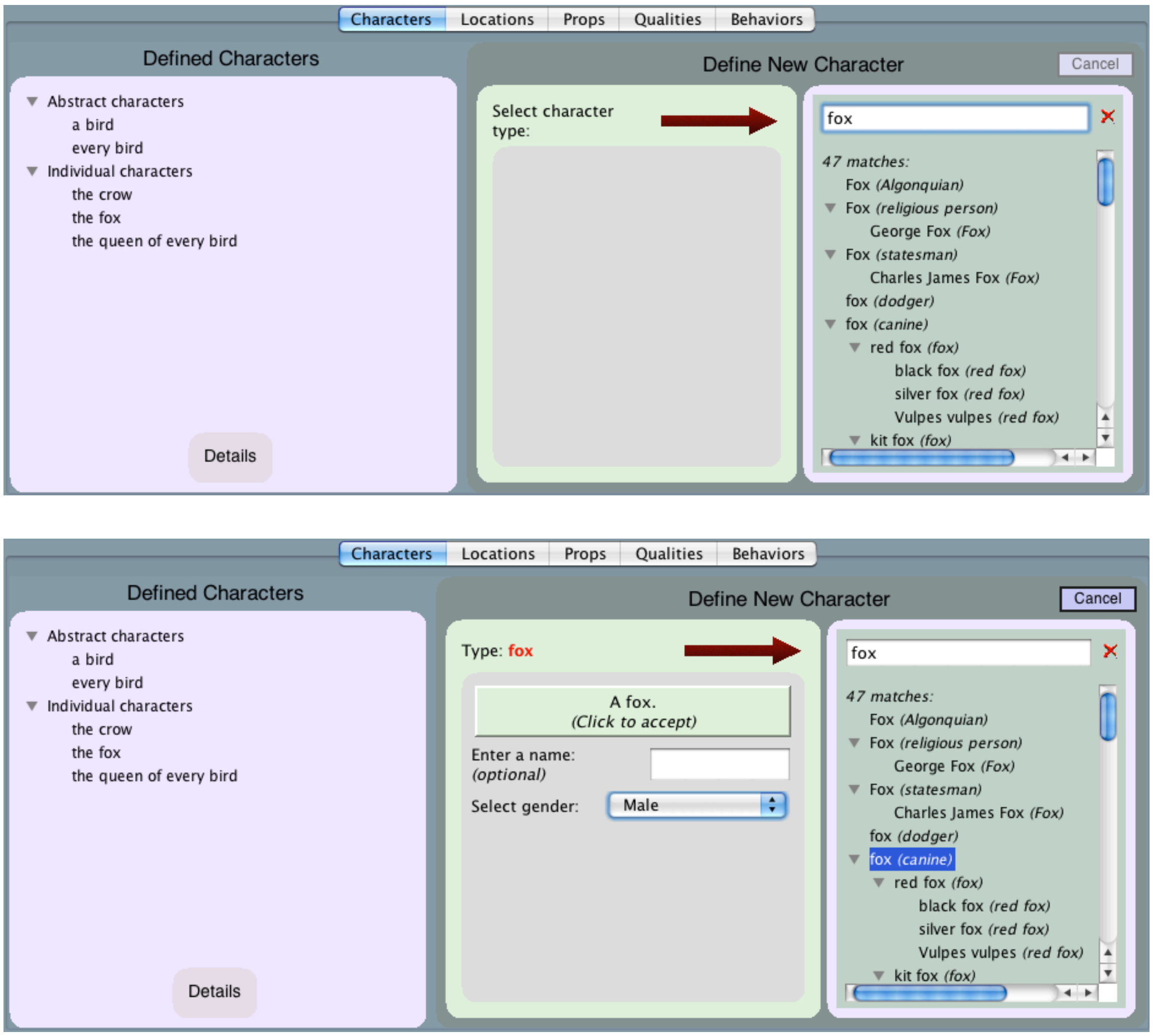


Figure 4.8: Instantiating an object in the SCHEHERAZADE Story Elements screen. Selecting an object type (top), and supplying metadata.

with “fox, deceiver, wrongdoer.” The second technique is to limit ambiguities by selectively disassociating certain lexemes from their less-used synsets. Specifically, we set a threshold for the information content a synset must have to serve as an option for an instance object. WordNet provides this attribute from a model of the usage of each synset in a naturally occurring corpus.

Noun phrases that include part-of and attributional relationships can be instantiated with two special frames: *Part of something* provides for non-exclusive part-of relationships (“a branch of the tree”) while *Part of something (unique)* refers to exclusive relationships (“the beak of the crow”) or attributed qualities (“the hue of the bird’s plumage”). Figure 4.9 shows the process of creating an instance object for the noun phrase “a branch of a tree.” At the top, the annotator selects the *Part of something* frame at the top of a list of prop types. The form is populated with two slots that need filling: the type of part and the object to which the part is attributed. Each slot is accompanied by a red **question button** which the annotator selects when she is ready to answer the question and fill the slot. Upon clicking the first question button, “What type of part?”, the user is prompted to select a type from the searchable type selector—in this case, *branch*. The accept button remains flat, indicating that the form needs additional slots to be filled before the object can be instantiated, but the feedback text on the face of the button expresses the partially completed object: “A branch of something.” When the annotator clicks the second question button, “Part of what?”, the selector presents a list of existing instance objects from which to choose. Upon picking the appropriate host for the branch, *the tree*, the accept button becomes raised and clickable, reading “A branch of a tree.” Clicking the raised accept button prompts the system to create the instance object, which can in turn be used as a “host” object for another part (*the edge of the branch of the tree*).

The object extraction panel also allows the annotator to provide **attributes** to each instance object. Attributes are statives which are immutable throughout the story-world. If a character is known to be tall, beautiful, or jealous, such a stative can be assigned as an attribute of the character. The feedback text generator uses the attribute when introducing characters: “A beautiful maiden.”

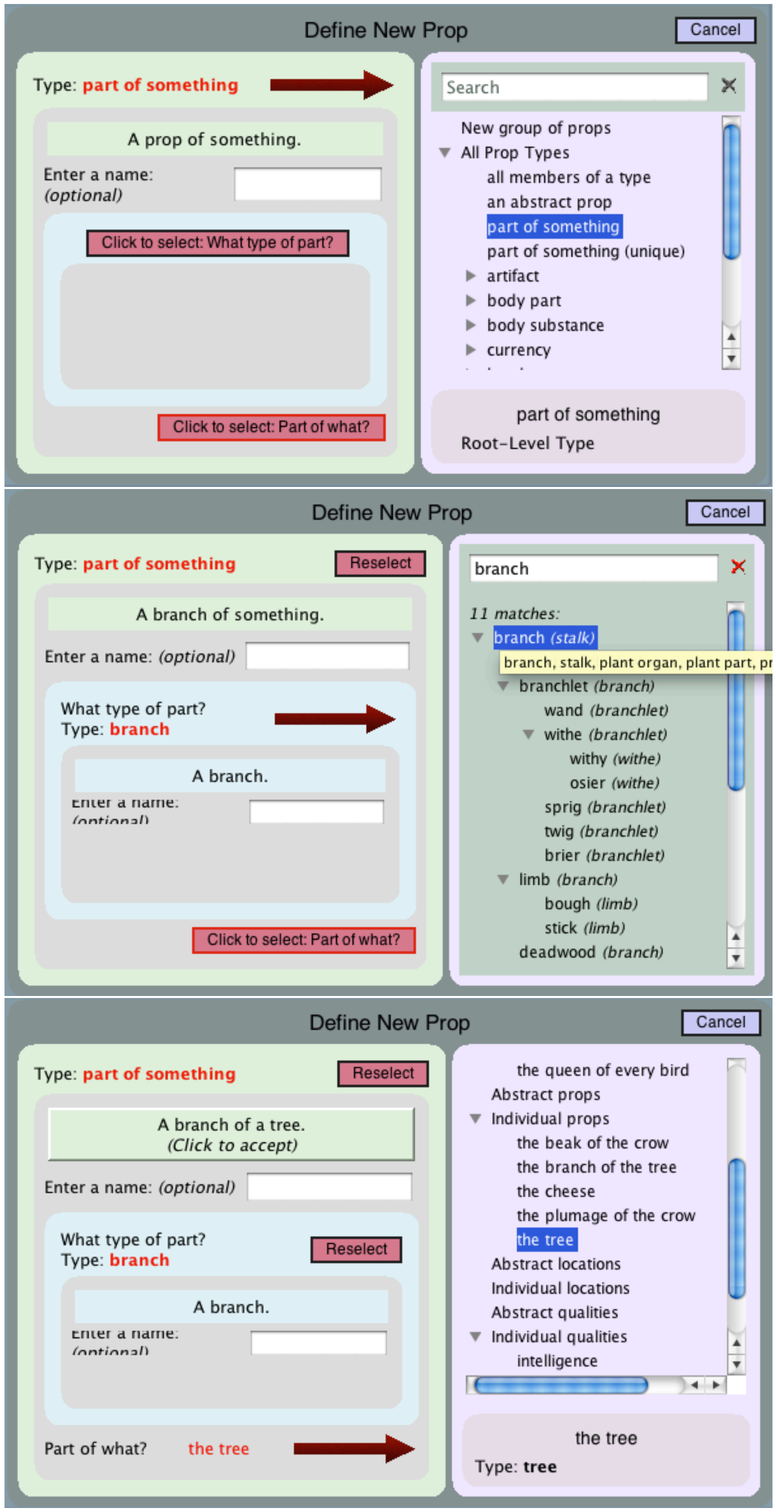


Figure 4.9: Noun phrases can be instantiated with object frames.

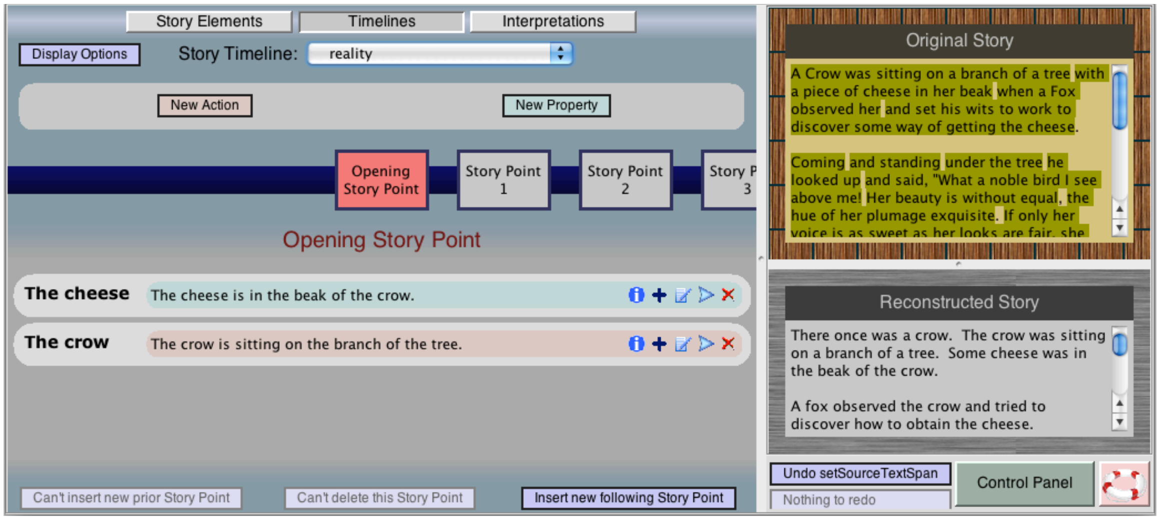


Figure 4.10: The Timelines panel provides an interface for propositional modeling.

4.3.4 Propositional modeling

The annotator constructs the timeline layer of the SIG encoding by carefully instantiating events, statives and modifiers, and assigning them to points in a main Reality timeline or an alternate modal timeline (an approach we introduced in Section 3.3.1). From a user interface standpoint, this part of the annotation process takes place in the Timelines panel of the GUI. The panel features a visualization of a timeline as a vector of boxes—"Story Points" stand in for State nodes.[3] When the annotator selects a Story Point, it slides to the center of the panel, and all **instance events** (instantiated verb predicates) and **instance statives** which occur during that Story Point are listed below the timeline (whether or not they start or end during that Story Point).

To create a new instance event or instance stative, the annotator first navigates to the Story Point where the event or stative is to begin. She then selects "New Action" or "New Property," respectively, from the top of the panel.[4] The main section of the panel

[3]The relationship between Story Points at the GUI level and State nodes in the SIG is indirect. Each Story Point refers to the span of time between two adjacent State nodes. While an event cannot occur in a single instant of time, it can occur in a single Story Point. We inserted this layer of abstraction after formative evaluations showed the state/span distinction to be confusing to users.

[4]We altered the terminology for usability reasons following formative evaluations.

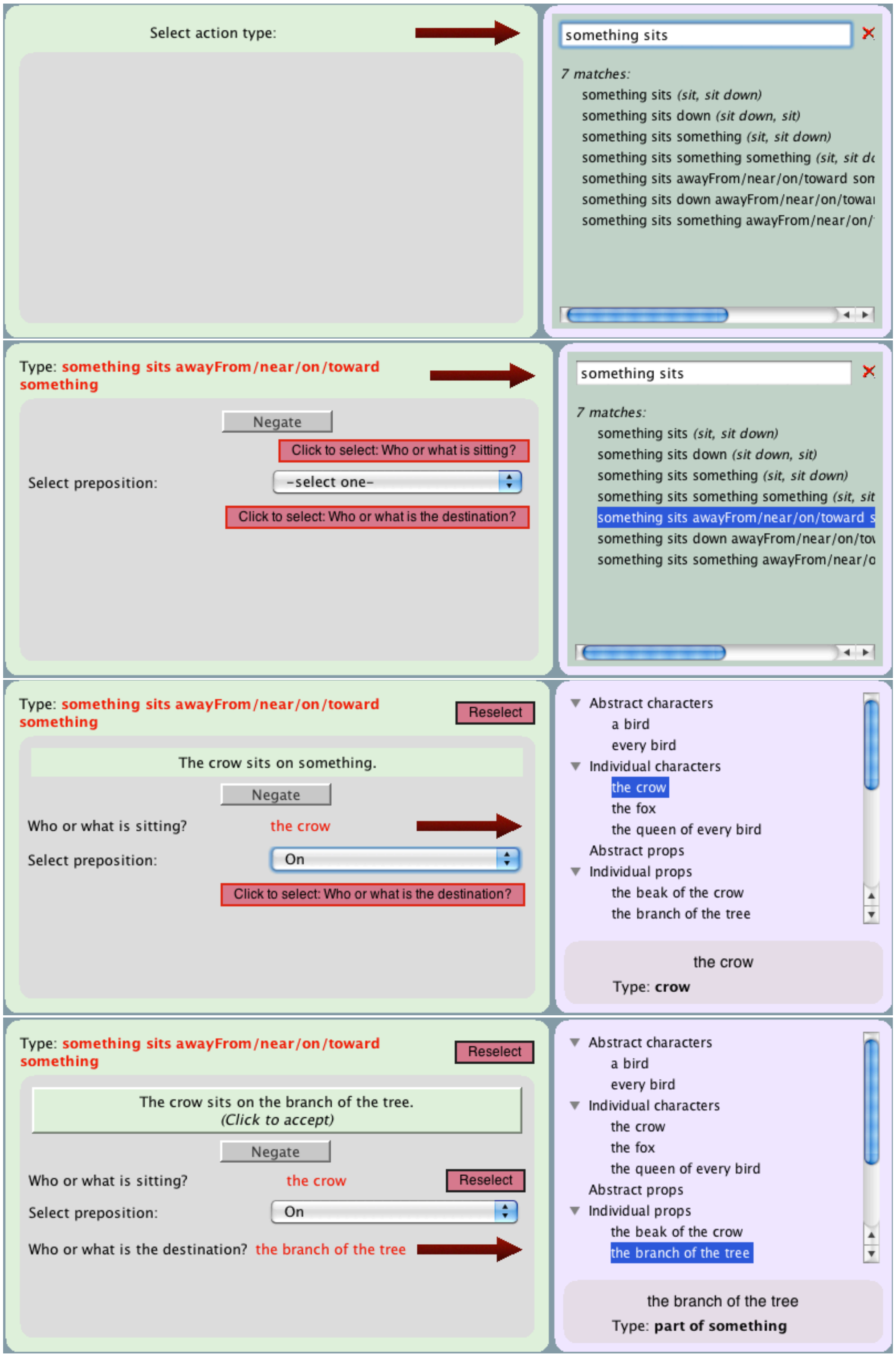


Figure 4.11: Modeling an instance stative from the SCHEHERAZADE Timelines screen.

then presents a form and a selector very similar to those used in the object extraction panel. The process is similar as well: First, select a type (in this case, a verb or stative predicate); second, fill in appropriate slots with arguments as prompted by question buttons; third, verify the results using the feedback text on the accept button, making changes as necessary; and fourth, click the accept button to create the new node.

Figure 4.11 illustrates this process with respect to the *sits* clause from the first sentence of "The Fox and the Crow": "A Crow was sitting on a branch of a tree." First, the annotator searches for an appropriate predicate frame by typing the present-tense form of a verb into the search panel of the type selector. The selector has indexed all of the "menu names" which we previously generated in the process of adapting VerbNet records (Figure 4.5). In this case, a search for "sits" returns seven frames which differ in their thematic roles (slots) and their use of prepositions. Based on the syntactic construction of the sentence she is attempting to model, the annotator selects the *Something sits away from/near/on/toward something* frame. The system populates the form with three questions: Who or what is sitting, who or what is the destination, and what is the preposition? The annotator clicks each question button in turn. For both the "who or what" questions, the annotator selects from among the instance objects she previously modeled: the crow is the agent, and the branch of the tree is the destination. The annotator then selects "on" as the appropriate preposition, and checks the feedback text that has been generated and displayed in the accept button: "The crow sits on the branch of the tree." The annotator also has the option to negate the proposition ("the crow does not sit on the branch of the tree"), but in this case, simply clicks the accept button to construct the instance stative and attach it to the timeline at the Story Point which she previously indicated.

Before clicking the accept button on any new event or stative, the user **highlights the span of source text** that corresponds to the new content. This has the effect of creating a new Text (TE) node in the textual layer of the SIG. The propositional content is attached to the new Event or Stative node with an *interpreted as* arc, as in Figure 3.7. A single span can be associated with multiple propositions. Note that the order of highlighted text spans in the Original Story box need not match the order of propositions on the timeline; this allows annotators to capture differences between the telling time and the story time

(Section 3.3).

Once constructed, the new instance event or stative appears below the timeline, as in Figure 4.10, in the form of feedback text; the overall feedback text for the story as a whole (in the Reconstructed Story panel) is updated to reflect the new content. The annotator can then edit or delete previously constructed content. Specifically, button near each span of feedback text allow users to attach modifiers (constructed as propositions, as in Figure 4.11), change the predicate frame, fill the frame with different arguments, reassign the event or stative to a different beginning or ending Story Point, or delete the story content altogether.

The argument selection area of the form changes with respect to the thematic role of the slot being filled, according to the VerbNet mappings we described in Tables 4.3 and 4.4. For an *Experiencer* slot, the interface presents a list of extracted characters; for a *Communication* slot, it prompts the user to select a predicate frame to serve as the basis for a nested dialogue proposition. Figure 4.12 shows an example of the process for nesting propositions. Upon selecting the communicative frame *Something says some proposition*, the annotator can select a predicate frame for the dialogue act from among those permitted by the encapsulating frame's selectional restrictions ("how" action, alternate timeline, and so on). A second form, color-coded differently, is graphically nested inside the first, and the annotator proceeds to fill out the inner form. (Forms can be nested indefinitely to create complex propositions.) Once the inner proposition is satisfactorily complete, the accept button shows overall feedback text such as "The fox says that the crow is noble."

The sentence we have modeled, about the crow sitting on the branch, is straightforward. At other times, a fair amount of simplification of the source text is necessary. For example, our guidelines have annotators phrase passive voice statements as active voice where possible. For "The crow was hugely flattered by this," the annotator would encode the equivalent of "The fox flattered the crow" or "the fox's words flattered the crow." Note that the word "flatter" in this case, itself a translation of the original Greek, carries certain semantic ambiguities relating to the theory-of-mind interpretation of the text. Did the fox have an ulterior motive or did he simply embarrass the crow with excessive praise? Did the crow doubt the fox's sincerity or believe that he was being genuine? We ask annotators to

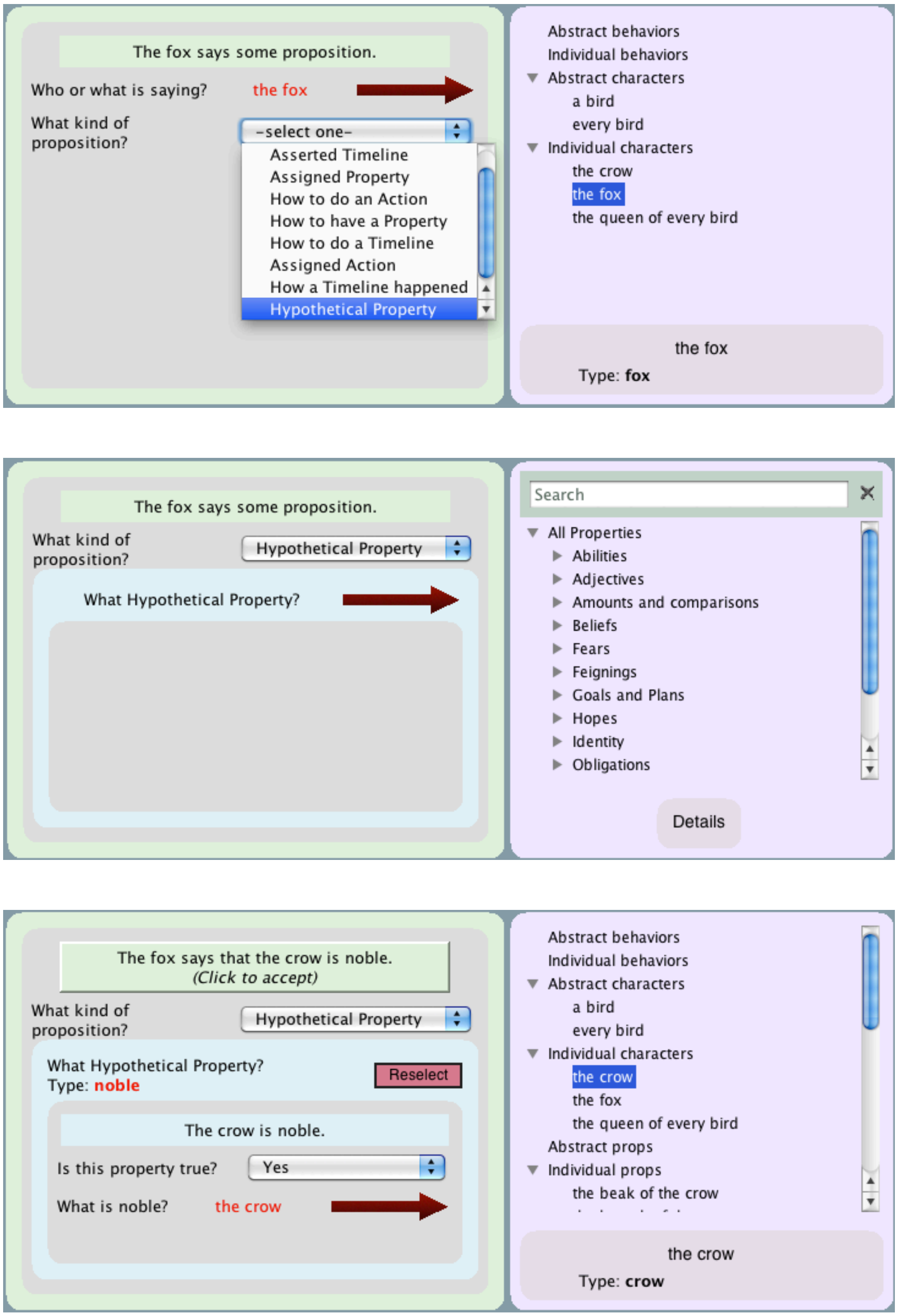


Figure 4.12: Complex propositions can be modeled by nesting an event, stative or alternate timeline as an argument.

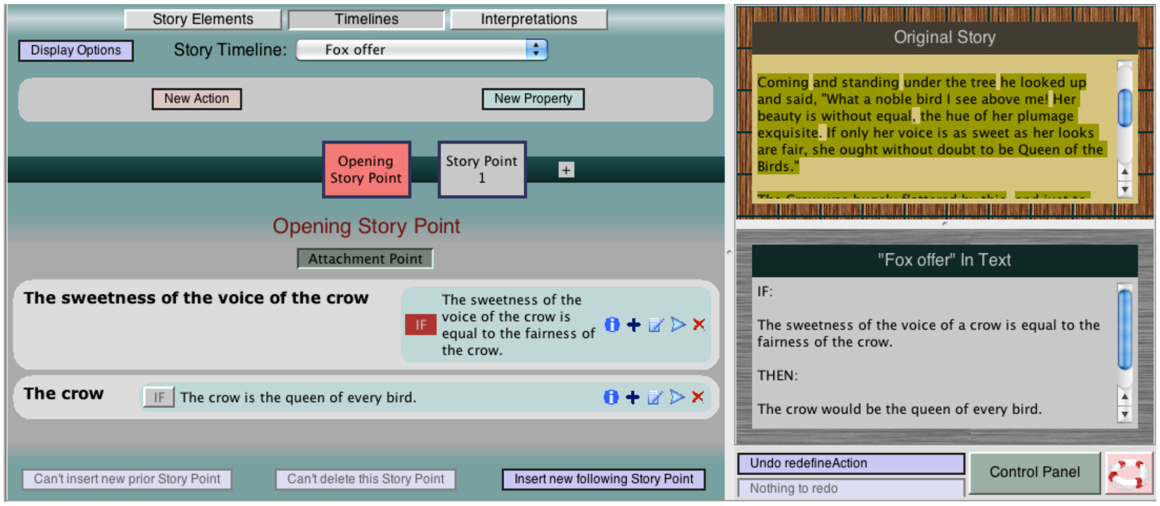


Figure 4.13: An alternate timeline in the SCHEHERAZADE Timelines screen.

use interpretative-layer relations to answer such questions whenever possible, rather than try to disambiguate the original text with complex propositional constructions. When this is not possible, annotators replace idioms and figures of speech with their closest approximations inside the controlled vocabulary. This issue appears in other annotation projects such as those dealing with the Wall Street Journal corpus, where many verbs are used metaphorically rather than literally [Gedigian *et al.*, 2006].

Alternate timelines, which we introduce in Section 3.3.1, are useful for referring to past events and statives, possible futures, hypothetical scenarios and other modal content. An alternate timeline is attached to a "parent" timeline with an *equivalent* arc that runs between those states that act as a common point of reference between the two frames of time. Annotators can easily create an attach timelines using the graphical interface. A dropdown menu in the *Timelines* panel lets users switch from the Reality timeline to an alternate timeline, and to create a new timeline with a particular parent timeline. The view of an alternate timeline is similar to that of the Reality timeline, except the color scheme is different, and the Reconstructed Story box only displays the feedback text equivalent of the alternate timeline (as opposed to a rendering of the entire story).

One situation where alternate timelines are put to use in "The Fox and the Crow" is when the fox makes a devious offer to the crow in saying, "If only her voice is as sweet as her

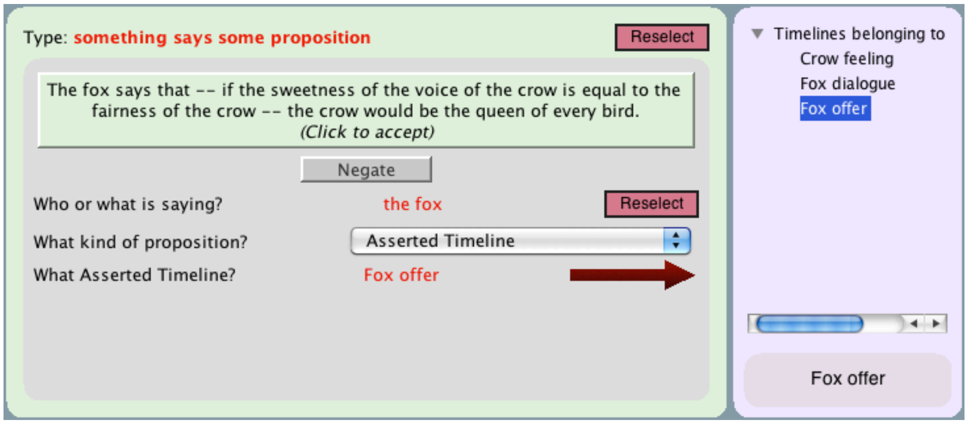


Figure 4.14: A form invoking an alternate timeline in a dialogue frame.

looks are fair, she ought without doubt to be Queen of the Birds." The fox is not referring to an actual event, but a hypothetical event, and the consequences which would follow. We set up an alternate timeline called "Fox offer" that features two statives, one in which the sweetness of the crow's voice is equal to the fairness of the crow's looks, and one in which the crow is the Queen of the Birds (Figure 4.13). In the interpretative layer of the SIG, we model the same if-then relationship between the two statives with a *would cause* arc inside a belief frame of the crow, which is itself within a goal frame of the fox—since the fox wants the crow to believe that one would lead to the other (a deception pattern, as in Figure B.10 in Appendix B). Propositional modeling in the timeline layer offers a parallel mechanism for expressing conditionals and subjunctives. Each event and stative can be marked with a flag called *IF* that indicates conditionality; if *IF* is invoked in an alternate timeline, whatever content that is not marked with *IF* is interpreted as being predicated on the *IF* content. As seen in Figure 4.13, the annotator marks only the voice-looks equivalence with *IF*, because the identity of the crow as Queen of the Birds is the consequence rather than the condition.

Alternate timelines are visualized with a button named *Attachment Point* below the currently selected State Time box. This button determines which State node is incident to the *equivalence* arc connecting the alternate timeline to its parent. The other end of the *equivalence* arc is attached to the event or stative that invokes the alternate timeline, such as in Figure 4.14. The selection of attachment point determines what parts of the alternate timeline are past, present and future with respect to the parent timeline. If the attachment point for “Fox offer” were at an earlier Story Point than either the “IF” clause or the “THEN” clause, then the alternate timeline would be a possible future, predicated on the crow’s voice becoming as sweet as her looks are fair.

We noted in Sections 3.4 and 4.2 that propositional modeling is “optional” with respect to the formal definition of the SIG schemata. Proposition and Interpretative Proposition nodes need only be annotated with the identities of their agents. At the user interface level, this is simply a matter of choosing an agent to act and using it in generic predicate frame such as *Something acts*. The annotator still highlights the span of source text she wishes to represent, but the timeline node itself only includes the identity of the acting agent (if any). The timeline node can then be connected to the interpretative-layer content. This usage pattern reduces the semantic precision offered by fuller propositional encoding, but also reduces the corresponding issues of coverage. The distinction between “story time” and “telling time” is preserved, as is the relationship between the surface form a discourse (the source text) and the rich set of discourse relations available in interpretative-layer annotation.

4.3.5 Interpretative panel

The third and final panel of the Scheherazade GUI, Interpretations, presents a canvas for annotators to draw nodes and arcs that represent their interpretative-layer (theory of mind) modeling of the source text. Annotators have a more direct control over the encoding’s graph structure here than in the previous panels, in that they can view and manipulate nodes and their incident arcs rather than use abstractions such as Story Points.

Figure 4.15 displays this panel with respect to a completed annotation for “The Fox and the Crow”. The large expanse of the panel contains three columns of boxes. The first two columns of boxes are automatically populated onto the canvas as the annotator builds the timeline: The rightmost column contains blue boxes that represent the content of the Reality timeline, chronologically ordered and “flattened” into a single vector (modifiers are offset slightly); the leftmost column contains the spans of the source text that the annotator associated with each proposition. The remainder of the panel, to the right of the second column, begins as a blank canvas. The annotator is charged with filling it with interpretative

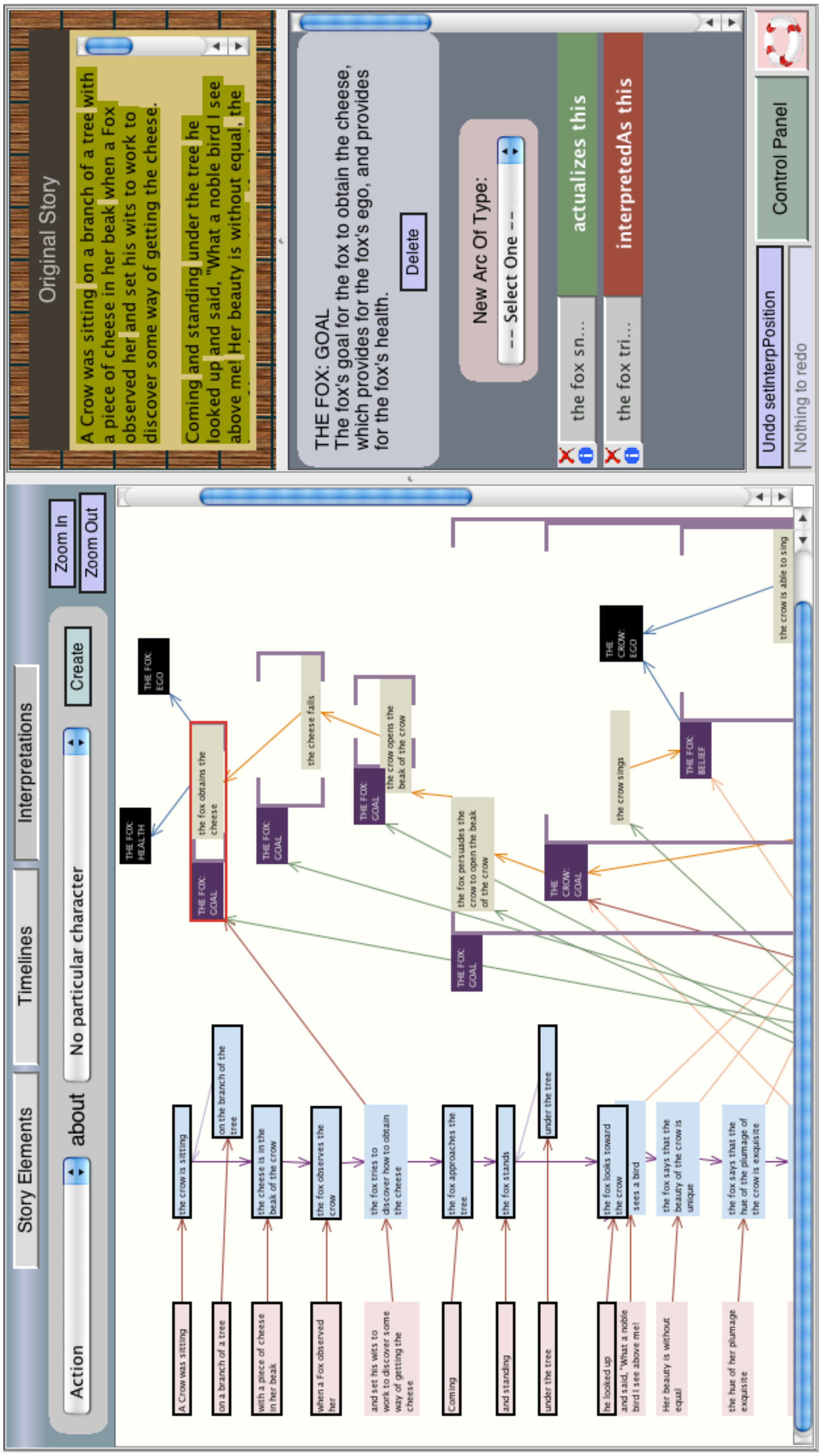


Figure 4.15: Interpretative annotation panel.

content according to the guidelines we outlined in Section 3.3.2. In a sense, this panel offers the annotator a view of the entire encoding in graph form, including the textual, timeline and interpretative layers.

Interpretative content includes agency frames (goals, plans and beliefs), Interpretative Proposition nodes and Affect nodes. These are drawn as boxes with purple labels, tan-colored labels, and black labels, respectively. The annotator can construct new nodes using the "Create" button at the top of the panel and then drag them around the canvas until they are positioned to her satisfaction. If the annotator wishes to model a proposition for an I node, the interface summons the proposition construction panel, with the same frame selector and argument-specification form as in the other two interface panels. New nodes can be created inside an existing agency frame or in ground truth (the white background canvas), a distinction whose semantic entailments we discussed in Section 3.3.2.

The Reconstructed Story panel on the lower-right corner has been replaced here with an "Interpretative Detail" panel. Whenever the user selects a node or a frame, the panel provides three features:

1. Feedback text which has been generated to describe the selected node or frame. In the case of a frame, the feedback text summarizes the goal, plan or belief inside the frame.

2. A **new arc** button allows the annotator to draw a new interpretative arc from the selected node to another node. A dropdown panel shows the arc types which can legally originate from the selected node, according to the rules of the SIG schemata.

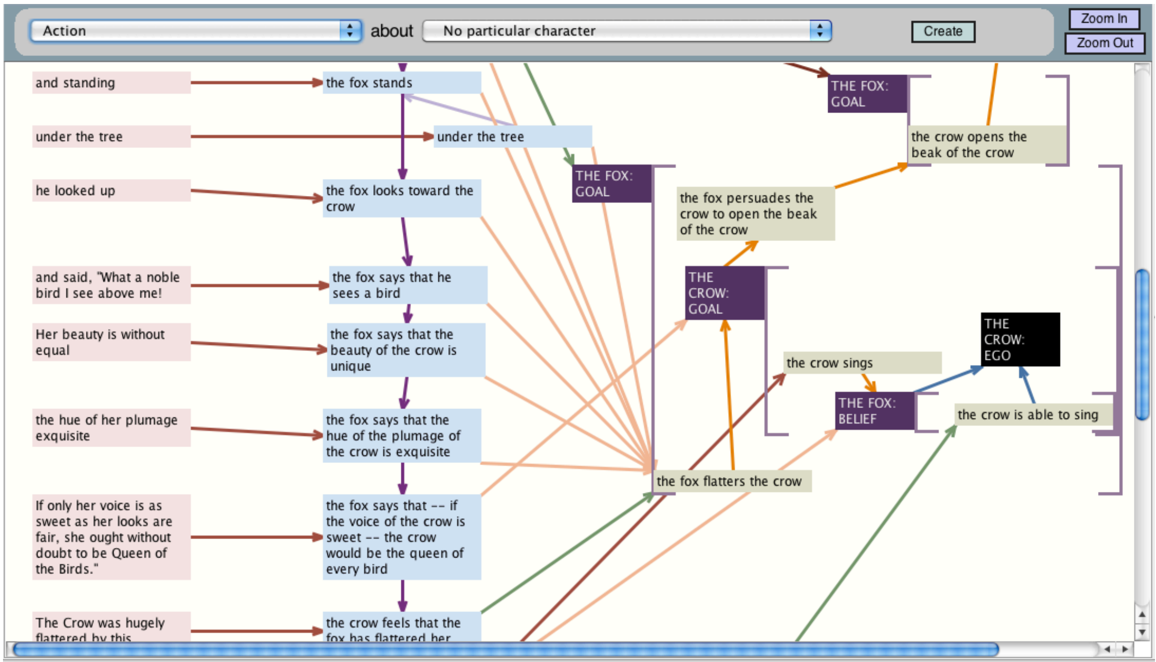
3. A list of the nodes which are connected to the selected node. The arc types that describe the connection are also displayed. The annotator can click on the adjacent node to select it, or delete the incident arc.

The bottom-right corner of Figure 4.15 shows the interpretative detail box that is displayed for the goal frame representing the fox's ultimate goal to obtain the cheese. The automatically generated feedback text describes the goal, as well as the affectual impact determined by the connected Affect nodes (positive for the fox's ego and the fox's health).

Two timeline nodes are connected to this frame, and by extension, two spans of source text are connected as well: "The fox set his wits to work to discover some way of getting the cheese" is *interpreted as* this goal box, and "the fox snatches the cheese" *actualizes* it.

We saw in Section 3.3.2.1 that each node and frame of interpretative content carries an *actualization status* that is logically entailed for each point in story time (each state in the Reality timeline). Each node begins in a Hypothetical (H) status, and then becomes either Actualized (A) or Prevented/Ceased (PC) if a timeline-layer node connects to it with an actualizing arc (*interpreted as*, *implies*, *actualizes*) or the *ceasing* arc. As the actualization status of a node determines whether it is effectively true at each point in story time, the annotator must carefully track the actualization triggers that she inserts with these four arcs. To aid in this process, the annotation tool calculates and displays the actualization status for each interpretative node when a timeline node is selected.

Figure 4.16 gives an example of actualization status highlighting. The top panel shows the section of the interpretative graph for "The Fox and the Crow" which models the fox's goal to obtain the cheese. The fox plans to flatter the crow, which would cause the crow to plan to sing in order to prove its worth, which would cause the crow to open its beak and drop the cheese. As no node is selected in this panel, the nodes and boxes are colored as usual. The bottom panel of Figure 4.16 shows the effects of selecting the timeline stative "The crow feels that the fox has flattered her." The plan node for "The fox flatters the crow" is now shaded with a dark green, indicating that it has been actualized (indeed, the timeline stative connects to this node with an *actualizes* arc). The label of the goal box (THE FOX: GOAL) is also shaded with dark green, since the fox's plan frame itself is long since actualized at this point in the story. All the other nodes are grey because they are still hypothetical—we do not yet know if the fox will succeed in prompting the crow to devise a plan to sing, or if the crow will open its beak. However, the nested goal box (the crow's potential plan) is shaded with a light green, indicating that the fox expects the frame to be actualized (because he believes the successful flattering *would cause* it). We describe expectations in more detail in Section B.1.



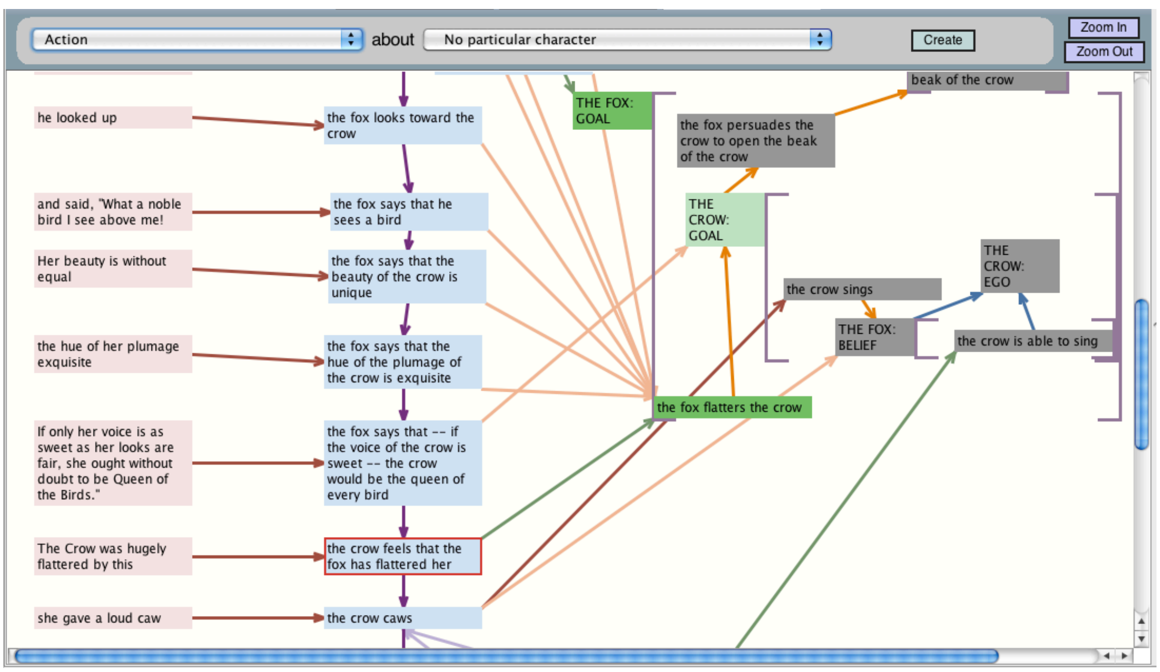


Figure 4.16: Normal interpretative graph interface (top) and interface with elements color-coded for actualization status relative to a timeline proposition.

4.3.6 Conclusion

This section has given a walkthrough of the highlights of the graphical user interface we have written on top of the Scheherazade API to make the annotation process amenable to community annotation efforts. Given a source text as input, annotators can construct nodes to represent named entities, use dynamic forms to fill out the slots of event and stative frames, and encode the meaning of the story with interpretative-layer content.

We built this interface in iterations. Three separate formative evaluations, each involving compensated users from outside our department, guided its development by attempting to encode a set of Aesop's fables. These users gave valuable feedback regarding the system's ease of use, the coverage of its knowledge base, and the expressiveness of the discourse relations. We followed up with two "production" collection experiments which we describe in Chapter 5. First, however, let us discuss in some detail the text generation module which provides the WYSIWYM feedback text. This generated text proved essential to our efforts to improve the system's ease of use.

4.4 Text Generation: Assigning Tense and Aspect

While Scheherazade is not strictly a project in text generation, we found through formative evaluations that generated feedback text greatly enhances the usability of the interface. Our text generation module reverses the process of semantic encoding by taking a portion of the encoding as a starting point and applying a model of syntax, tense, and aspect to render and display a natural-language equivalent in English. We demonstrated in the previous section how feedback text is used in every facet of the user interface, including renderings of both small pieces of the data structure (e.g., noun types) and large ones (the entire timeline layer as a "Reconstructed Story"). The Scheherazade API also provide access to the generation module, allowing third-party tools to repurpose feedback text outside of the GUI. This section provides an overview of the generation algorithm we have implemented, and goes into detail on the question of assigning tense and aspect when rendering events that are situated in a formal encoding of time. This work on tense and aspect is the most novel aspect of our approach.

STORY	→	TIMELINE
TIMELINE	→	STATE+
STATE	→	ACTION+, STATIVE+
ACTION	→	AGENT MODIFIER VERB ARG1 ARG2
STATIVE	→	AGENT MODIFIER STATIVE ARG1 ARG2
AGENT	→	NOUN
NOUN	→	NOMINAL? NAME

Table 4.6: A selection of the rules involved in the feedback text generator, in the style of a grammar.

4.4.1 Basic Planner and Realizer

The module is best summarized as a small series of rules which can be called within a generation command. Most rules are tied to a particular encoding facet that one would wish to serialize as text. For instance, a `generateCharacter` rule will return a textual rendering of an instance character (`crow1` becomes "a crow"). Upon execution, each rule carries out three tasks:

1. Constructs a *generation plan*, a series of instructions that call other rules upon execution;
2. Updates keys and values in a state object; and
3. Issues a sub-command to execute its generation plan, calling each of the rules in its plan in sequence and passing them the state object.

Some rules directly *emit* a particular lexeme or symbol to a *serializer*. The overall result is a tree structure of hierarchical plans; the tree is generated "on the fly" during the course of a preorder traversal. In a sense, this process acts as a grammar; a simplified list of the generation rules and their plans (involving other rules) is seen in Table 4.6. Unlike in a context-free grammar, though, a state object allows each node in the tree (that is, each rule in execution) a common "whiteboard" with which to communicate.

An example of this process is shown in Figure 4.17. We wish to generate text from the proposition `walk(john)`, an event on the Reality timeline, from the temporal perspective of a point in time between the proposition's begin and end times. In 4.17(i), a `generateEvent` rule takes the proposition as input, along with a constant, `PRESENT_TELLING`, that expresses

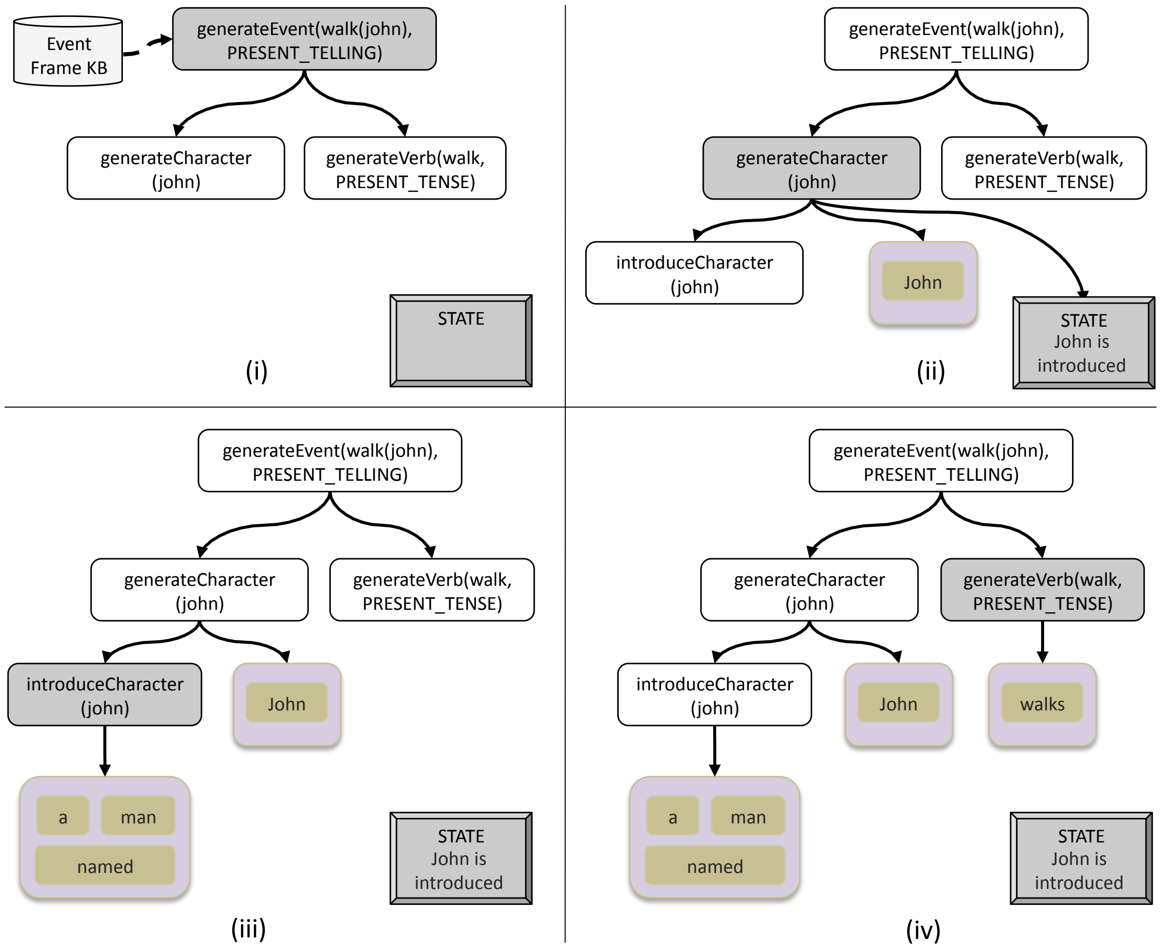


Figure 4.17: Progressive construction and traversal (execution) of a tree of generation rules. Each rule emits appropriate lexemes and updates/consults a state object.

our temporal perspective that the event is presently occurring. The rule also uses the world knowledge module—in particular, the generation frame for the `walk()` verb we earlier adapted from a VerbNet record (Figure 4.5 in Section 4.2). The knowledge base tells us that to serialize the event, `generateEvent` must call two other rules: one to render the agent who is walking, and one to render the `walk` verb.

In 4.17(ii), the algorithm recurses on the first call, `generateCharacter(john)`. Throughout this figure, the rounded rectangle that is shaded is the one currently being executed. The `generateCharacter` rule checks the state object for whether the `john` instance object is *given* or *new* to the discourse, and finding it to be new, dynamically inserts a call to a rule

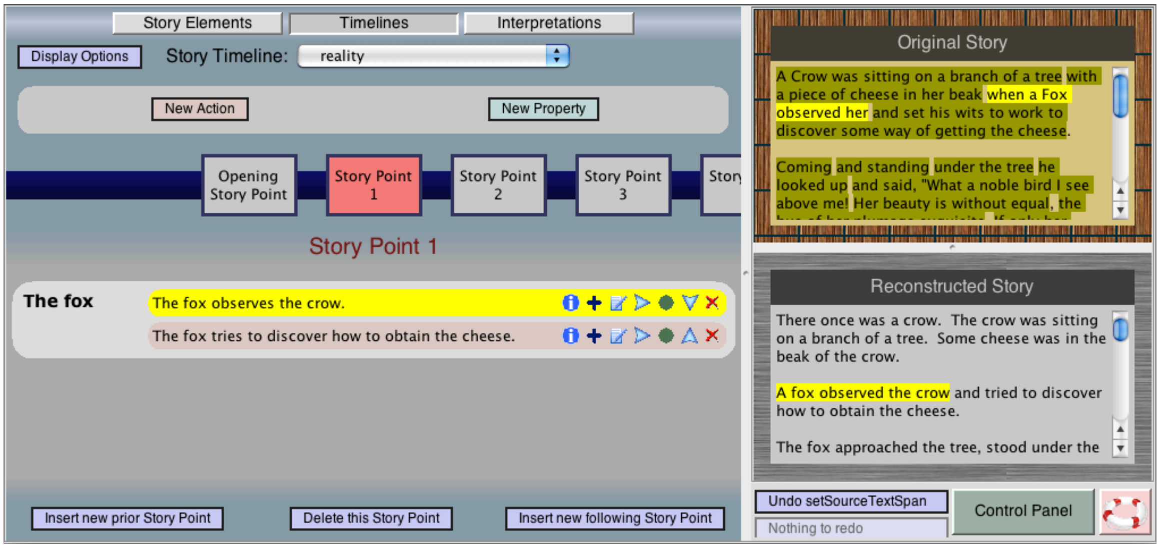


Figure 4.18: The use of feedback text in the SCHEHERAZADE GUI. Clicking on a span of source text, an event or stative in the Timelines panel, a span of feedback text in the Reconstructed Story panel, or a node in the Interpretations screen causes all four equivalent elements to become highlighted.

called `introduceCharacter` before a call to emit the name "John." It also updates the state object to record that John has been introduced. In 4.17(iii), the `introduceCharacter` rule eventually emits three words, "a man named," though for brevity we do not show a nested call to a rule `generateType` for rendering "man." (We also do not show the recursive call to a `generateModifier` rule, which emits text when modifiers are associated with events.) 4.17(iv) shows the completion of the tree traversal, in which `generateVerb` uses a model of tense and aspect to determine that "walks" is the correct form of the verb to emit.

Once the evaluation of the plan generation tree is complete, the emitted lexemes are collected and post-processed by the serializer. This function joins together clauses, sentences, and paragraphs with appropriate punctuation, spacing and capitalization: "A man named John walks." It also maintains an index which maps each clause to the semantic structure from which the clause was derived. This allows the GUI to provide a feature whereby the user can click on a word in the Reconstructed Story panel and navigate to the corresponding point in the story in any other panel—the Timelines panel selects the appropriate Story State, the Source Text panel highlights the equivalent span of source text, and the Interpretations panel selects the node as it appears in the canvas view (Figure 4.18).

To generate a full discourse from the story's timeline, we implemented several higher-level rules. A `generateTimeSpan` rule concatenates all events that begin or end during that span in a single list, aggregating by the common agent ("X did Y and Z"); a `generateStory` rule renders the entire story by "scrubbing" the timeline from the beginning to the end and generating a sentence or paragraph for each relevant time span. Note that the feedback text in the Reconstructed Story panel is therefore generated with a strictly linear telling of story content, even if the source text has temporal disfluencies (Figure 3.8 in Section 3.3.1).

We took special care with the generation of referring expressions. Entities given proper names (by annotators) are introduced as such when they are new ("a man named John"), but subsequently referred to by name alone ("John") rather than nominal ("the man"). When multiple unnamed entities are used, we use ordinal disambiguators ("a rooster fought <u>a second rooster</u>; <u>the first rooster</u> defeated <u>the second rooster</u>"). Discrete and continuous objects are given different determiners ("a" vs. "some"). We use the state object to determine which entity was the most recent to be mentioned for a particular gender, and if possible, replace a name with the pronoun appropriate to the thematic role (e.g., "<u>she</u> walked to <u>him</u>," "<u>it</u> destroyed <u>itself</u>.)" We also use the state object in conjunction with the control-verb metadata to determine when an agent in a subordinate clause can be assumed and not stated. For instance, Figure 4.19 shows the generation plan tree and resulting serialization for a timeline with a single event:

$$\texttt{ask}(\texttt{jill}, \texttt{john}, \texttt{driveTo}(\texttt{john}, \texttt{jill}, \texttt{store}))$$

Because `ask` is a control verb, binding the entity being asked, we do not include "John" in the subordinate clause. Leaving aside character introductions, the result is "Jill asked John to drive her to the store," rather than "Jill asked John for John to drive her to the store." For metadata on verbs and nouns, including control, irregular conjugations, irregular plural forms, and discreteness, we relied on the COMLEX lexicon [Macleod *et al.*, 1994].

For the remainder of this section, we focus on the **assignment of tense and aspect** by the `generateStory` and `generateTimeSpan` rules. All generation systems that communicate knowledge about time must select tense and aspect carefully in their surface realizations. An incorrect assignment can give the erroneous impression that a continuous action has

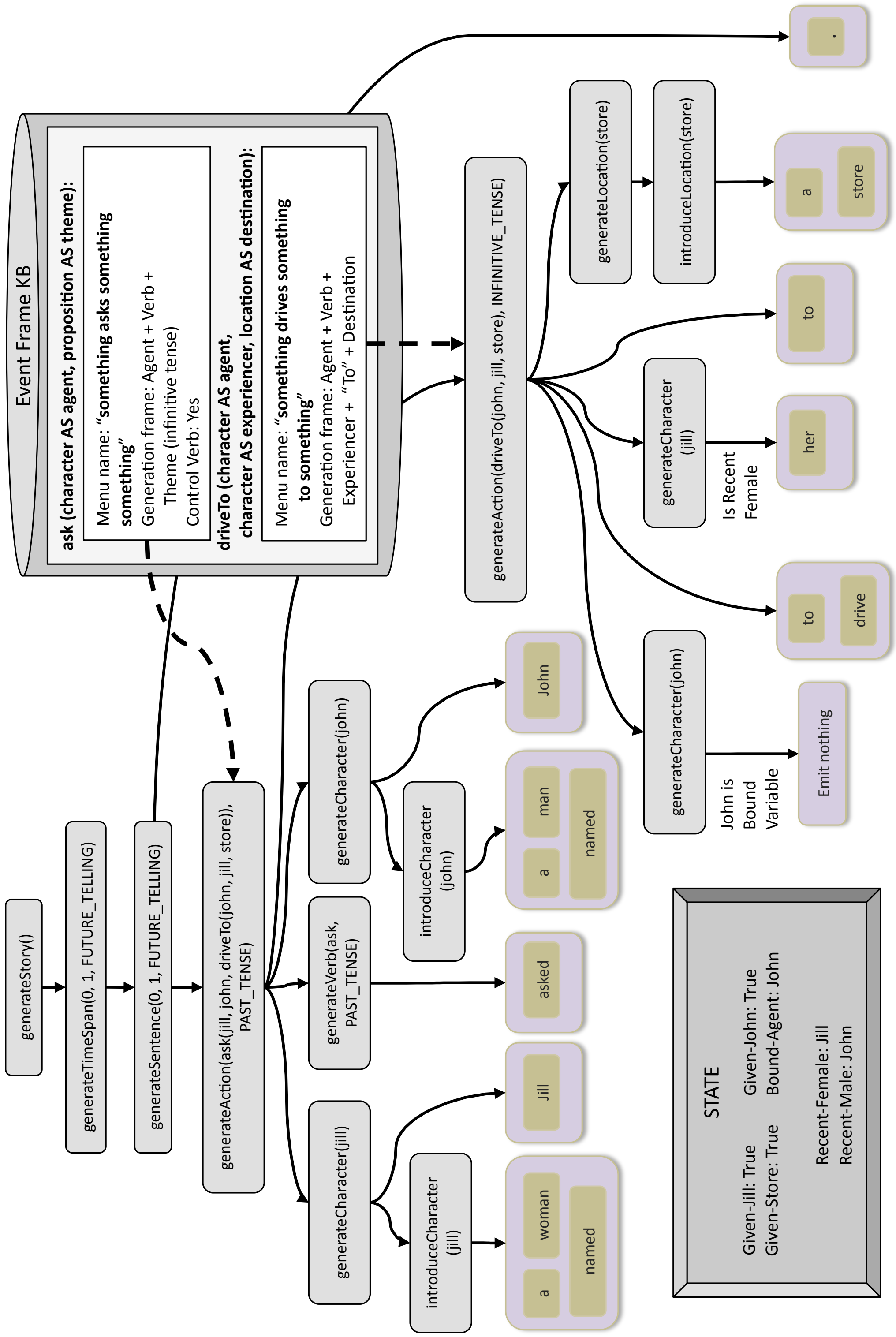


Figure 4.19: The preorder traversal of a generation plan tree involving a subordinate clause.

ended, or that a previous state is the current reality. Correct assignments are particularly important in the generation of narrative discourse, where statives and actions occur over connected intervals.

We will describe two contributions: first, a general application of theories of tense, aspect and interval logic to a generation context in which we map temporal relationships to specific tense/aspect selections. Second, an implementation of this approach in the basic sentence planner and realizer we have now described as the Scheherazade textual generation module. The first result does not depend on the second.

The discussion is organized as follows: After discussing related work in Section 4.4.2 and reviewing our formal model of time in Section 4.4.3, we describe our method for selecting tense and aspect for single events in Section 4.4.4. Section 4.4.6 follows with more complex cases involving multiple events and shifts in temporal focus (in particular, direct speech, indirect speech, modals and conditional events).

4.4.2 Related Work

There has been intense interest in the interpretation of tense and aspect into a formal understanding of the ordering and duration of events. This work has been in both linguistics [Smith, 1978; Dowty, 1979; Nerbonne, 1986; Vlach, 1993] and natural language understanding. Early systems investigated rule-based approaches to parsing the durations and orderings of events from the tenses and aspects of their verbs [Harper and Charniak, 1986; Hinrichs, 1987; Webber, 1987; Song and Cohen, 1988; Passonneau, 1988]. Allen [1984] and Steedman [1995] focus on distinguishing between achievements (when an event culminates in a result, such as "John builds a house") and processes (such as walking). More recent work has centered on markup languages for complex temporal information [Mani, 2004] and corpus-based (statistical) models for predicting temporal relationships in unseen text [Mani *et al.*, 2006; Lapata and Lascarides, 2006].

Our annotation interface requires a fast realizer that can be easily integrated into an interactive, online encoding tool. We found that developing a custom realizer as a module to our Java-based system was preferable to integrating a large, general purpose system such as KPML/Nigel [Mann, 1983; Matthiessen and Bateman, 1991] or FUF/SURGE [Elhadad and Robin, 1996]. These realizers, along with RealPro [Lavoie and Rambow, 1997], accept tense as a parameter, but do not calculate it from a semantic representation of overlapping time intervals such as ours (though the Nigel grammar can calculate tense from speech, event, and reference time orderings, discussed below). The statistically trained FERGUS [Chen *et al.*, 2002] contrasts with our rule-based approach.

Dorr and Gaasterland [1995; 2002] and Grote [1998] focus on generating temporal connectives, such as "before," based on the relative times and durations of two events; Gagnon and Lapalme [1996] focus on temporal adverbials (e.g., when to insert a known time of day for an event). By comparison, we extend our approach to cover direct/indirect speech and the subjunctive/conditional forms, which they do not report implementing. While our work focuses on English, Yang and Bateman [2009] describe a recent system for generating Chinese aspect expressions based on a time interval representation, using KPML as their surface realizer. For an English-language grammar tutoring system, Fum et al. [1991] introduce extra-linguistic entity (what they call an "objective tense") that reflects the temporal semantics of the clause or sentence being rendered, then map this entity onto a grammatical tense; we take a similar two-step approach.

Several other generation projects also involve encodings of narrative discourse. Callaway and Lester's STORYBOOK [2002] aims to improve fluency and discourse cohesion in realizing formally encoded narratives; Ligozat and Zock [1992] allow users to interactively construct sentences in various temporal scenarios through a graphical interface.

4.4.3 Temporal knowledge

The propositions that we aim to realize take the form of a predicate, one or more arguments, zero or more attached modifiers (either a negation operator or an adverbial, which is itself a proposition), and an assignment in time. Each argument is associated with a semantic role (such as Agent or Experiencer), and may include entities backed by nouns (such as characters) or other propositions. Predicates include both durative actions and statives [Dowty, 1979]; we will refer to both as *events* as they occur over intervals. For example, here are two events:

$$\texttt{walk}(\texttt{Mary}, \texttt{store}, 2, 6) \tag{4.1}$$

$$\texttt{hungry}(\texttt{Julia}, 1, \infty) \tag{4.2}$$

The latter two arguments in (4.1) refer to time states in a totally ordered sequence; Mary starts walking to the store at state 2 (as encoded by the *ba* arc) and finishes walking at state 6 (via *ea*). (4.2) begins at state 1, but is unbounded (Julia never ceases being hungry). We do not currently address the use of reference times (such as equating a state to 6:00 or "yesterday").

(4.1) and (4.2), depending on the situation, can be realized in several aspects and tenses. We adapt and extend Reichenbach's [1947] system of symbols for distinguishing between simple and progressive aspect. Reichenbach identifies three points that define the temporal position of the event: the event time E, the speech time S and a reference time R which may or may not be indicated by a temporal adverbial. The total ordering between these times dictates the appropriate tense and aspect. For example, the simple past "John laughed" has the relation $E < S$. $R = E$ because there is no separate reference time involved. The past perfect "John had laughed [by the end of the play]" has the relation $E < R < S$, in that it describe "the past of the past," with the nearer "past" being R (the end of the play). R can be seen as the temporal focus of the sentence.

As Reichenbach does not address events with intervals, we redefine E as the tuple describing the onset and termination states attached to the event (for example, (2,6) for Mary's walk). This definition deliberately assumes that no event ever occurs over a single "instant" of time. The perception of an instantaneous event, when it is needed, is instead created by dilating R into an interval large enough to contain the entire event, as in Dowty's approach [Dowty, 1979].

We also distinguish between two generation modes: realizing the story as a complete discourse (*narration mode*, as seen in the "Reconstructed Story" panel) and describing the content of a single state or interval (*snapshot mode*, as seen in the Timelines and Interpretations panels). Our system supports both modes differently. In narration mode, we realize the story as if all events occur before the speech time S, which is the style of most literary fiction. (We shall see that this does not preclude the use of the future tense.)

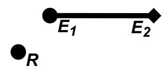
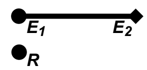
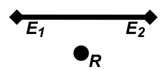
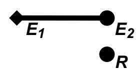
Diagram	Relations	Perspective
	$R < E_1$	Before
	$R = E_1$ $R < E_2$	Begin
	$E_1 < R$ $R < E_2$	During
	$R = E_2$ $R > E_1$	Finish
	$R > E_2$	After

Table 4.7: Perspective assignment for viewing an event from a reference state.

In snapshot mode, speech time is concurrent with reference time so that the same events are realized as though they are happening "now." The system uses this mode to allow annotators to inspect and edit what occurs at any point in the story. In Figure 4.18, for instance, the fox's observing of the crow is realized as both a present event in snapshot mode ("the fox observes the crow") and narrated as a past event ("a fox observed the crow"). In both cases, we aim to precisely translate the propositions and their temporal relationships into reliable feedback text. In the remainder of this section, we describe our method for assigning tenses and aspects.

4.4.4 Expressing single events from a reference state

In both snapshot and narration modes, we often need to render the events that occur at some reference state R. We would like to know, for instance, what is happening now, or what happened at 6:00 yesterday evening. The tense and aspect depend on the *perspective* of the reference state on the event, which can be bounded or unbounded. The two-step process for this scenario is to determine the correct perspective, then pick the tense and aspect class that best communicates it.

Perspective	Generation mode	**English tense**	System's construction	Example
After	Future Speech	Past perfect	*had* {PAST PART.}	She had walked.
	Present Speech	Present perfect	*has/have* {PAST PART.}	She has walked.
	Past Speech	Future perfect	*will have* {PAST PART.}	She will have walked.
	Modal Infinitive		*to have* {PAST PART.}	To have walked.
Finish	Future Speech	"Finished"	*stopped* {PR.P.}	She stopped walking.
	Present Speech	"Finishes"	*stops* {PR.P.}	She stops walking.
	Past Speech	"Will finish"	*will stop* {PR.P.}	She will stop walking.
	Modal Infinitive		*to stop* {PR.P.}	To stop walking.
During	Future Speech	Past progressive	*was/were* {PR.P.}	She was walking.
	Present Speech	Present progressive	*am/is/are* {PR.P.}	She is walking.
	Past Speech	Future progressive	*will be* {PR.P.}	She will be walking.
	Modal Infinitive		*to be* {PR.P.}	To be walking.
During-After	Future Speech	Past perfect progressive	*had been* {PR.P.}	She had been walking.
	Present Speech	Present perfect progressive	*has/have been* {PR.P.}	She has been walking.
	Past Speech	Future perfect progressive	*will have been* {PR.P.}	She will have been walking.
	Modal Infinitive		*to has/have been* {PR.P.}	To have been walking.
Begin	Future Speech	"Began"	*began* {INFINITIVE}	She began to walk.
	Present Speech	"Begins"	*begins* {INFINITIVE}	She begins to walk.
	Past Speech	"Will begin"	*will begin* {INFINITIVE}	She will begin to walk.
	Modal Infinitive		*to begin* {PR.P.}	To begin walking.
Contains	Future Speech	Simple past	{SIMPLE PAST}	She walked.
	Present Speech	Simple present	{SIMPLE PRESENT}	She walks.
	Past speech	Simple future	*will* {INFINITIVE}	She will walk.
	Modal Infinitive		{INFINITIVE}	To walk.
Before	Future Speech	"Posterior"	*was/were going* {INF.}	She was going to walk.
	Present Speech	Future	*am/is/are going* {INF.}	She is going to walk.
	Past Speech	Future-of-future	*will be going* {INF.}	She will be going to walk.
	Modal Infinitive		*to be going* {INFINITIVE}	To be going to walk.

Table 4.8: Tense/aspect assignment and realizer constructions for describing an action event from a particular perspective and speech time. "PR.P." means "present participle."

We define the set of possible perspectives to follow Allen [1983], who describes seven relationships between two intervals: before/after, meets/met by, overlaps/overlapped by, starts/started by, during/contains, finishes/finished by, and equals. Not all of these map to a relationship between a single reference *point* and an event interval. Table 4.7 maps each possible interaction between E and R to a perspective, for both bounded and unbounded events, including the defining relationships for each interaction. A diamond for E_1 indicates *at or before*, i.e., the event is either anteriorly unbounded ($E_1 = -\infty$) or beginning at a state prior to R and E_2. Similarly, a diamond for E_2 indicates *at* or *after*.

Once the perspective is determined, covering Reichenbach's E and R, speech time S is determined by the generation mode. Following the guidelines of Reichenbach and Dowty, we then assign a tense for each perspective/speech time permutation in Table 4.8. Not all permutations map to actual English tenses. Narration mode is shown as *Future Speech*, in that S is in the future with respect to all events in the timeline. (This is the case even if E is unbounded, with $E_2 = \infty$.) Snapshot mode is realized as *Present Speech*, in that $R = S$. The fourth column indicates the syntactic construction with which our system realizes the permutation. Each is a sequence of tokens that are either cue words (*began, stopped*, etc.) or conjugations of the predicate's verb. "Posterior" is how Reichenbach refers to the "future-of-a-past" situation, for which no tense exists in English; we use the "was going" construction as in "Mary was going to [later] walk to the store." These constructions emphasize precision over fluency.

As we have noted, theorists have distinguished between *statives* that are descriptive ("John was hungry"), *achievement* actions that culminate in a state change ("John built the house"), and *activities* that are more continuous and divisible ("John read a book for an hour") [Dowty, 1979]. Prior work in temporal connectives has taken advantage of lexical information to determine the correct situation and assign aspect appropriately [Moens and Steedman, 1988; Dorr and Gaasterland, 1995; Gagnon *et al.*, 2006]. In our case, we only distinguish between actions and statives, based on information from WordNet and VerbNet. We use a separate table for statives; it is similar to Table 4.8, except the constructions replace verb conjugations with insertions of *be*, *been*, *being*, *was*, *were*, *felt*, and so on (with the latter applying to affective states). We do not currently distinguish between achievements and activities in selecting tense and aspect, except that the annotator is tasked with "manually" indicating a new state when an event culminates in one (e.g., "The house was complete"). Recognizing an achievement action can benefit lexical choice (better to say "John finished building the house" than "John stopped") and content selection for the discourse as a whole (the house's completion is implied by "finished" but not by "stopped").

To continue our running examples, suppose propositions (4.1) and (4.2) were viewed as a snapshot from state $R = 2$. Table 4.7 indicates *Begin* to be the perspective for (1), since $E_1 = R$, and Table 4.8 calls for a tense/aspect permutation that means "begins at the present time." When the appropriate construction is inserted into the overall syntax for `walk(Agent, Destination)`, which we derive from the VerbNet frame for "walk," the result is "Mary begins to walk to the store;" similarly, (4.2) is realized as *Julia is hungry* via the *During* perspective. Narration mode invokes past-tense verbs.

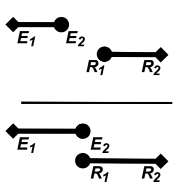
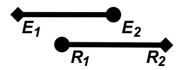
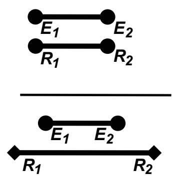
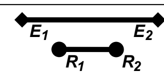
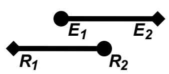
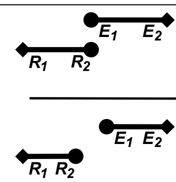
Diagram	Relations	Perspective	Diagram	Relations	Perspective
	$R_1 \geq E_2$	After		$R_1 > E_1$ $E_2 > R_1$ $R_2 > E_2$	Finish
	$R_1 \leq E_1$ $R_2 \geq E_2$	Contains		$E_1 < R_1$ $E_2 > R_2$	During
	$R_1 < E_1$ $R_2 > E_1$ $E_2 > R_2$	Begin		$E_1 \geq R_2$	Before

Table 4.9: Perspective assignment for describing an event from an assigned perspective.

4.4.5 Expressing single events from a reference interval

Just as events occur over intervals, rather than single points, so too can reference times. One may need to express what occurred when "Julia entered the room" (a non-instantaneous action) or "yesterday evening." Our system allows annotators to view intervals in snapshot mode to get a sense of what happens over a certain time span.

The semantics of reference intervals have been studied as extensions to Reichenbach's point approach. Dowty [1979, p.152], for example, posits that the progressive fits only if the reference interval is completely contained within the event interval. Following this, we construct an alternate lookup table (Table 4.9) for assigning the perspective of an event from a reference interval. Table 4.8 then applies in the same manner. In snapshot mode, the speech time S also occurs over an interval (namely, R), and Present Speech is still used. In narration mode, S is assumed to be a point following all event and reference intervals. In our running example, narrating the interval (1,7) results in "Mary walked to the store" and "Julia began to be hungry," using the *Contains* and *Begin* perspectives respectively.

The notion of an *unbounded* reference interval, while unusual, corresponds to a typical

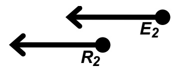
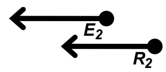
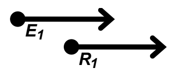
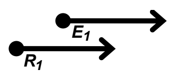
Diagram	Relations	Perspective
	$E_2 > R_2$ $E_1 = -\infty$ $R_1 = -\infty$	During (*a priori*)
	$R_2 > E_2$ $E_1 = -\infty$ $R_1 = -\infty$	After
	$R_1 > E_1$ $E_2 = \infty$ $R_2 = \infty$	Contains
	$E_1 > R_1$ $E_2 = \infty$ $R_2 = \infty$	Before

Table 4.10: Perspective assignment if event and reference intervals are unbounded in like directions.

perspective if the event is either bounded or unbounded in the opposite direction. These scenarios are illustrated in Table 4.9. Less intuitive are the cases where event and reference intervals are unbounded in the same direction. Perspective assignments for these instances are described in Table 4.10 and emphasize the bounded end of R. These situations occur rarely in this generation context.

Event Subintervals

We do not always want to refer to events in their entirety. We may instead wish to refer to the beginning, middle or end of an event, no matter when it occurs with respect to the reference time. This invokes a second reference point in the same interval [Comrie, 1985, 128], delimiting a subinterval. Consider "John searches for his glasses" versus "John continues to search for his glasses"—both indicate an ongoing process, but the latter implies a subinterval during which time, we are expected to know, John was already searching.

Our handling of subintervals falls along four alternatives that depend on the interval $E_1..E_2$, the reference time R and the subinterval $E'_1..E'_2$ of E, where $E'_1 \geq E_1$ and $E'_2 \leq E_2$.

1. **During-After.** If E' is not a final subinterval of E ($E'_2 < E_2$), and $R = E'_2$ or R is a subinterval of E that is met by E' ($R_1 = E'_2$), the perspective of E' is defined as *During-After*. In Table 4.8, this invokes the perfect-progressive tense. For example, viewing example (4.1) with $E' = (2, 4)$ from $R = 4$ in narration mode (Future Speech) would yield "Mary had been walking to the store."

2. **Start.** Otherwise, if E' is an initial subinterval of E ($E'_1 = E_1$ and $E'_2 < E_2$), the perspective is defined as *Start*. Not shown in Table 4.8, the construction for this case reassigns the perspective to that between R and E'. Our realizer reassigns the verb predicate to *begin* (or *become* for statives) with a plan to render its only argument, the original proposition, in the infinitive tense. For example, narrating (4.2) with E' =(1,2) from $R = 3$ would invoke the After perspective between R and E', yielding "Julia had become hungry."

3. **Continue.** Otherwise, and similarly, if E strictly contains E' ($E'_1 > E_1$ and $E'_2 < E_2$), we assign the perspective *Continue*. To realize this, we reassign the perspective to that between R and E', and reassign the verb predicate to "continue" (or "was still" for statives) with a plan to render its only argument, the original proposition, in the infinitive: "Mary had continued to walk to the store" for (4.1) with E' =(3,4) and $R = 7$.

4. **End.** Otherwise, if E' is a final subinterval of E ($E'_1 > E_1$ and $E'_2 = E_2$), we assign the perspective *End*. To realize this, we reassign the perspective to that between R and E', and reassign the verb predicate to "stop" (or "finish" for cumulative achievements). Similarly, the predicate's argument is the original proposition rendered in the infinitive.

4.4.6 Expressing multiple events in alternate timelines

This section covers more complex situations involving alternate timelines—the feature of our representation by which an event in the main timeline can refer to a second frame of time. Other models of time have supported similar encapsulations [Crouch and Pulman, 1993; Mani and Pustejovsky, 2004]. The alternate timeline can contain references to actual events

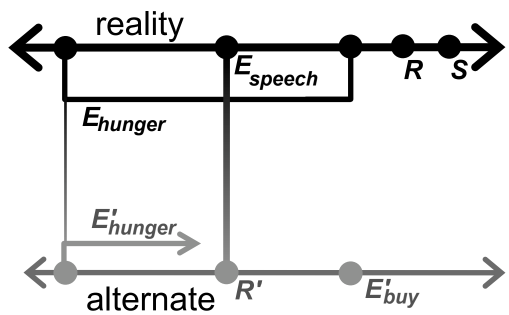


Figure 4.20: Schematic of a speech act attaching to an alternate timeline with a hypothetical action. R' and E_{speech} are attachment points.

or modal events (imagined, obligated, desired, planned, etc.) at earlier or later time states with respect to its *attachment point* (Section 4.3.4) on the main timeline. This is primarily used in practice for modeling dialogue acts, but it can also be used to place real events at uncertain time states in the past (e.g., the present perfect is used in a reference story being encoded).

Reassigning Temporal Focus

Ogihara [1995] describes dialogue acts involving changes in temporal focus as "double-access sentences." We now consider a method for planning such sentences in such a way that the refocusing of time (the reassignment of R into a new context) is clear, even if it means changing tense and aspect mid-sentence. Suppose Mary were to declare that she would buy some eggs because of Julia's hunger, but before she returned from the store, Julia filled up on snacks. If this speech act is described by a character later in the story, then we need to carefully separate what is known to Mary at the time of her speech from what is later known at R by the teller of the episode. Mary sees her purchase of eggs as a possible future, even though it may have already happened by the point of retelling, and Mary does not know that Julia's hunger is to end before long.

Following Hornstein's treatment of these scenarios [Hornstein, 1990], we attach R', the reference time for Mary's statement (in an alternate timeline), to E_{speech}, the event of her speaking (in the main timeline). The act of buying eggs is a hypothetical event E'_{buy} that falls after R' on the alternate (modal) timeline. S is not reassigned.

Figure 4.20 shows both timelines for this example. The main timeline is shown on top; Mary's speech act is below. The attachment point on the main timeline is, in this case, the speech event E_{speech}; the attachment point on an alternate timeline is always R'. The placement of R, the main reference point, is not affected by the alternate timeline. Real events, such as Julia's hunger, can be invoked in the alternate timeline (e arcs, Section 3.3.1, as drawn with a vertical line from E_{hunger} to an E'_{hunger} without an E'_2 known to Mary) but they must preserve their order from the main timeline.

The tense assignment for the event intervals in the alternate timeline then proceeds as normal, with R' substituting for R. The hypothetical "buy" event is seen in *Before* perspective, but past tense (Future Speech), giving the posterior (future-of-a-past) tense. Julia's hunger is seen as *During* as per Table 4.7. Further, we assert that connectives such as "because" do not alter R (or in this situation, R'), and that the E'_{buy} is connected to E'_{hunger} with a causality edge.

The result is: "Mary <u>had said</u> that she <u>was going to buy</u> eggs because Julia <u>was</u> hungry." The subordinate clause following *that* sees E'_{buy} in the future, and E'_{hunger} as ongoing rather than in the past. It is appropriately ambiguous in both the symbolic and rendered forms whether E'_{buy} occurs at all, and if so, whether it occurs before, during or after R. A discourse planner would have the responsibility of pointing out Mary's mistaken assumption about the duration of Julia's hunger.

We assign tense and aspect for quoted speech differently than for unquoted speech. Instead of holding S fixed, S' is assigned to R' at the attachment point of the alternate timeline (the "present time" for the speech act). If future hypothetical events are present, they invoke the Past Speech constructions in Table 4.8 that have not been used by either narration or snapshot mode. The content of the quoted speech then operates totally independently of the speech action, since both R' and S' are detached: "Mary said/says/was saying, 'I am going to buy eggs because Julia is hungry.'"

The focus of the sentence can be subsequently reassigned to deeper nested timelines as necessary (attaching E' to R'', and so on). Although the above example uses subordinate clauses, we can use this nesting technique to construct composite tenses such as those enumerated by Halliday [1976]. To this end, we conjugate the *Modal Infinitive* construction

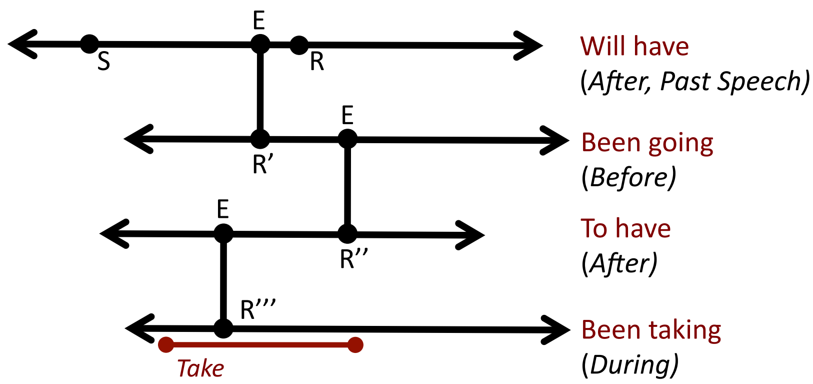


Figure 4.21: Chained alternate timelines used to model a complex tense from Halliday [1976]: "Will have been going to have been taking."

in Table 4.8 for each alternate timeline. For instance, Halliday's complex form "present in past in future in past in future" (as in "will have been going to have been taking") can be generated with four timelines in a chain that invoke, in order and with Past Speech, the *After*, *Before*, *After* and *During* perspectives. There are four Rs, all but the main one attached to a previous E (Figure 4.21).

Subjunctives and Conditionals

We finally consider tense and aspect in the case of subjunctive and conditional statements (if-thens), which can appear in alternate timelines (Section 4.3.4). The relationship between an *if* clause and a *then* clause is not the same as the relationship between two clauses joined by *because* or *when*. The *then* clause—or set of clauses—is predicated on the truth of the *if* clause. As linguists have noted [Hornstein, 1990, p.74], the *if* clause serves as an adverbial modifier, which has the effect of moving forward the reference point to the last of the *if* event intervals (provided that the *if* refers to a hypothetical future). Consider the sentence: "If John were to fly to Tokyo, he would have booked a hotel." A correct model would place E'_{book} before E'_{fly} on an alternate timeline, with E'_{fly} as the *if*. Since *were to fly* is a hypothetical future, $R' < E'_{fly}$. During regeneration, we set R' to E'_{fly} after rendering "If John were to fly to Tokyo," because we begin to assume that this event transpired. If R' is left unchanged, it may be erroneously left before E'_{book}: "Then he would be going to book a hotel."

Our annotation interface allows users to mark one or more events in an alternate timeline as *if* events. If at least one event is marked, all *if* events are rendered in the subjunctive mood, and the remainder are rendered in the conditional. For the *if* clauses that follow R', S' and R' itself are reassigned to the interval for each clause in turn. R' and S' then remain at the latest *if* interval (if it is after the original R') for purposes of rendering the *then* clauses. In our surface realizer, auxiliary words *were* and *would* are combined with the Modal Infinitive constructions in Table 4.8 for events during or following the original attachment point.

As an example, consider an alternate timeline with two statives whose start and end points are the same: “Julia is hungry” and “Julia is unhappy.” The former is marked *if*. Semantically, we are saying that `hungry(Julia)`$\rightarrow$`unhappy(Julia)`. If R' were within these intervals, the rendering would be: “If Julia is hungry, then she is unhappy” (*Contains*/Present Speech for both clauses). If R' were prior to these intervals, the rendering would be: “If Julia were to be hungry, then she would be unhappy.” This reassigns R' to E_{hungry}, using *were* as a futurative and *would* to indicate a conditional. Because R' and S' are set to E_{hungry}, the perspective on both clauses remains *Contains*/Present Speech. Finally, if both intervals are before R', describing Julia's previous emotional states, we avoid shifting R' and S' backward: “If Julia had been hungry, then she had been unhappy” (*After* perspective, Future Speech for both statives).

The algorithm is the same for event intervals. Take (4.1) and a prior event where Mary runs out of eggs:

$$\texttt{runOut}(\texttt{Mary}, \texttt{eggs}, 0, 1) \tag{4.3}$$

Suppose they are in an alternate timeline with attachment point $0'$ and (4.1) marked *if*. We begin by realizing Mary's walk as an *if* clause: “If Mary were to walk to the store.” We reassign R' to E_{walk}, (2,6), which diverts the perception of (4.3) from *Begins* to *After*: “She would have run out of eggs.” Conversely, suppose the conditional relationship were reversed, with (4.3) as the only *if* action. If the attachment point is $3'$, we realize (4.3) first in the *After* perspective, as R' does not shift backward: “If Mary had run out of eggs.” The remainder is rendered from the *During* perspective: “She would be walking to the store.”

Note that in casual conversation, we might expect a speaker at $R = 3$ to use the past simple: "If Mary ran out of eggs, she would be walking to the store." In this case, the speaker is attaching the alternate timeline at a reference interval that subsumes (4.3), invoking the *Contains* perspective by casting a net around the past. We ask our annotators to select the best attachment point pursuant to their understanding of the story content.

4.4.7 Discussion

As we mentioned earlier, we are describing two separate methods with a modular relationship to one another. The first is an abstract mapping from a conceptual representation of time in a narrative, including interval and modal logic, to a set of 11 *perspectives*, including the 7 listed in Table 4.8 and the 4 introduced in Section 4.4.5. These 11 are crossed with three scenarios for speech time to give a total of 33 tense/aspect permutations. We also use an infinitive form for each perspective. One may take these results and map them from other time representations with similar specifications.

The second result is a set of syntactic constructions for realizing these permutations in our text generation module. Our focus here, as we have noted, is not fluency, but a surface-level rendering that reflects the relationships (and, at times, the ambiguities) present in the encoding. We consider variations in modality, such as an indicative reading as opposed to a conditional or subjunctive reading, to be at the level of the realizer and not separate classes of tenses.

It has always been the goal in surface realization to generate sentences from a purely semantic representation. Our approach to the generation of tense and aspect explores a rule-based approach based on temporal intervals. We have applied prior work in linguistics and interval theory and tested our approach in an interactive narrative encoding tool. Our method handles reference intervals and event intervals, bounded and unbounded, and extends into subintervals, modal events, conditionals, and direct and indirect speech where the temporal focus shifts.

4.5 Conclusion

In this chapter, we described the development of SCHEHERAZADE as a software system capable of performing inference over general semantic networks, implementing the particular semantics of the SIG model of discourse relations we introduced in Chapter 3, and eliciting semantic encodings of narratives from trained annotators through a graphical user interface. We gave particular emphasis to the process of propositional modeling, including our use of external linguistic resources to populate a database of linguistic knowledge. We also described a textual generation module that is capable of generating feedback text from a SIG encoding, in whole or in part; in the context of this work, we described a model for the assignment of tense and aspect in narrative discourse that covers the linguistic scenarios made possible by the interval-based representation of time and modality employed by the SIG. To our knowledge, this combination of a formal representation of narrative time and mode and a generative model of tense and aspect is a novel contribution. We believe that SCHEHERAZADE can act as a foundational platform for future work in story representation and analysis, as an API allows third-party tools to utilize the library for storing, reading and comparing encodings.

In Chapter 5, we evaluate the SCHEHERAZADE implementation in the context of a series of corpus collection projects, and describe algorithms that allow us to leverage the representation to find similarities and analogies between encoded narratives.

Chapter 5

Collections and Experiments

Our premise, which we laid out in Chapter 1, is that a plethora of online discourse has a narrative component. An algorithm capable of finding thematic similarities between stories can greatly assist us in our need to filter, search, and otherwise organize the many stories to which we are exposed on a daily basis, from news articles to fiction and personal communication. Chapters 2 and 3 explored several types of "narrative components," with the former specializing in conversational networks found in literary fiction, and the latter proposing a broader set of relations describing temporal, causal, and agent-oriented relationships that provide for narrative cohesion.

The latter approach, which we call the Story Intention Graph, aims to be a middle ground between surface text and a formal model of plot, action and character. This final chapter explores both ends of its reach: On one end, we describe a corpus collection which brings SIG annotation to trained annotators (using the SCHEHERAZADE tool); on the other end, we explore some of the thematic insights that we can draw from applying formal inference rules to SIG encodings. The task that we apply to the SIG is that of **identifying similarities and analogies between stories**, a process that is crucial for all of the applications we have mentioned. If we can determine whether two stories are similar or analogous (isomorphic in structure), we can more easily interpret new stories by their multifaceted relationships with known stories. Much like a trained language model allows us to recognize n-grams as being more than the sum of their parts, a data bank of encoded stories would let us identify "narrative idioms" that recur and are likely to appear in future stories. As language models have been widely influential in tasks such as information retrieval, parsing and speech recognition at the clause level, a model of narrative idioms, and a method for accurately identifying them in previously unseen text and speech, would be influential at the discourse level. This discourse-clause distinction also distinguishes our task from that of measuring sentence-level similarity, which has been investigated by projects such as Columbia SimFinder [Hatzivassiloglou *et al.*, 2001].

In terms of process, we aim to develop procedural methods of identifying pairwise similarities between SIG encodings that relate to events (propositional content), goals, plans and affectual impact. We also strive to find overall trends that hold across a corpus of encodings. In a sense, we wish to accomplish automatically the type of structuralist analysis of similarities and trends that Propp performed on Russian folk-tales [Propp, 1969] and Bremond on French folk-tales [Bremond, 1970]. While these studies set out to find overlapping sequences of plot elements, we take as input a set of general-purpose encodings that have each been constructed for an individual story in isolation. Our contribution here is a series of algorithms for deriving such comparisons from multiple encodings, and evaluating their effectiveness at finding and expressing overlapping structure and content.

Of special note, we saw in Chapters 3 and 4 that the schemata can be used with or without the propositional modeling of surface text. Fully-realized predicate-argument structures enhance the formality of a particular encoding, but also increase the complexity of the task of constructing it. In the following sections, we investigate both the cost and benefit sides of this tradeoff, as we have collected and processed corpora using each variation of the approach:

1. **Collection A** consists of 40 encodings, two each for 20 of Aesop's fables, including propositional modeling but excluding the interpretative layer of the SIG (that is, including the textual and timeline layers only).

2. **Collection B** consists of 60 encodings covering 26 of Aesop's fables, including propositional modeling, and all three SIG layers.

3. **Collection C** consists of 10 encodings covering 8 samples of longer and more varied narrative discourse, including a news article, literary short fiction, contemporary

nonfiction, and epic poetry. These encodings include all three SIG layers, but only placeholder propositions in the timeline layer.

Collectively, these 110 encodings exist as machine-readable descriptions in a corpus we call **DramaBank**, which we have publicly released.[1]

This chapter is structured as follows: In Section 5.1, we give further details about the composition and collection of DramaBank. Section 5.2 describes an algorithm to find story similarities using propositional modeling alone. Section 5.3.1 describes a technique in which SIG patterns, interpretative-layer graph "idioms" which we devise in Appendices B and C, can be applied in order to extract features from the interpretative layers of Collections B and C. These features allow us to find story similarities and corpus-wide trends by comparing feature vectors. Section 5.3.2 describes a third algorithm for identifying similarities between encodings; here, we work "bottom-up" through the detection of dynamic graph isomorphisms, rather than "top-down" through the use of *a priori* patterns. Section 5.4 evaluates the efficacy of all three methods, individually and in tandem, for the task of determining the similarities between fables. We then conclude the chapter in Section 5.5.

5.1 Corpus Collection

We were motivated to use Aesop's fables for Collections A and B, as well as for the running examples in this thesis, for several reasons. The first was their extreme brevity, as a typical fable is only 125 words long. We were forced to find a corpus of short narratives in order to make the process of propositional modeling tractable, as we found that even after several iterations of developing and evaluating SCHEHERAZADE, the process remained labor-intensive. Indeed, Aesop's fables are a popular corpus for investigations in computational story generation and analysis, especially where the representations are more formal [Dolan and Dyer, 1985; Ryan, 1991; Goyal *et al.*, 2010a]. The second, more compelling reason to use this corpus (as opposed to another collection of brief snippets) was that they are rich with the type of thematic content we set out to model in Chapter 3, including agents with goals and plans, intentional actions, outcomes, and so on. Moreover, the author of

[1] http://www.cs.columbia.edu/~delson

these fables seems to have made them deliberately work as parables, which encourages us to find analogical mappings to other stories and real-life experiences. Although the fables have been the subject of varying interpretations over the centuries, they are most commonly seen today as a "metaphorical representation of reality used as a fictitious means of teaching ethics," as put in Zafiropoulos' [2001, 32] contemporary study of the collection. This is done by means of a "recognition function," where the fable is first understood on a level where an animal or other fable figure is a surrogate for a human (or a stereotype of a personality trait), and then understood on a level where human situations are made concrete and recognizable [2001, 38]. Where Zafiropoulos emphasizes the transference of ethical thought from the fable world to the reader's world, we leave the notion of finding ethical or moral "points" from SIG encodings to future work. It suffices for our present task that two fables are similar to each other when they describe overlapping circumstances, even if we do not identify the pedagogical point of identifying those circumstances (i.e., what an individual in an analogous situation should or should not do according to the moral code of the corpus). Since the circumstances in these fables are designed to be easily mapped onto readers' experiences, they are a rich testbed for our notion of recurring thematic content. As more than 600 fables exist in the collection, we were forced to select a subset for our collections. After reviewing each fable, we selected 26 which had causally connected events on a clear timeline—as many fables illustrate ethical points through dialogue rather than through action and example. We did not select fables on the basis of the SIG's interpretative layer, as we had not yet developed it.

The point of Collection C, in contrast, is to serve as a counterweight to the Aesop focus of Collections A and B. The most pressing motivation was to show that the SIG model and Scheherazade tool are not overfit to the fable domain or the Aesop corpus in particular; in a general sense, we aimed to show their utility for longer, more varied narrative discourse. To address the tractability issue, we did not only relax the constraint that propositional modeling be an integral component of the SIG timeline layer (allowing for placeholder propositions that only include the identity of an agent), we also relaxed the guideline we had imposed in Collections A and B that every non-trivial story clause be encoded as a timeline node of some kind. In other words, the SIG relations in Collection C do not completely cover the source texts. Instead, we asked annotators to highlight and encode as nodes only those passages in the source texts that had interpretative-layer connections according to their readings of entire texts. Digressions that did not significantly relate to goals, plans, intentional actions, affectual impacts or outcomes were left out of the timeline and interpretative layers (that is, not connected to the rest of the SIG encoding with *interpreted as* or any other arc). As we established in Chapter 3, these factors are commonly agreed by psychologists to underpin cognitive story understanding, even if not all parts of a discourse relate to them directly. The benefit of this approach is that it allows for long but diffuse texts to be annotated as quickly as short, dense texts, where a "denser" text is one with more interpretative-layer implications per unit length. It also saves time that annotators would spend modeling propositions, a task more amenable to automatic extraction (e.g., semantic role labeling) than the identification of temporal and agent-centric relations—and, as we shall see, propositional encodings are not as helpful as interpretative-layer content for the purpose of identifying narrative similarities and analogies. Put differently, an interpretative-layer-only approach to annotation focuses annotator time and effort on the aspect of the SIG which requires more narrative-specific judgment and provides greater returns for the similarity task. We feel that the potential for an assisted annotation approach, with a system offering propositional modeling while a human annotator builds interpretative content, is a promising direction for future work.

The difference between Collections A and B is that only the latter includes interpretative-layer annotation. (Collection B covers all of the fables covered in Collection A, plus six additional fables.) This is a historical artifact in the composition of DramaBank, as we began our corpus collection before we fully developed the SIG. We procured Collection A by asking two annotators to carry out propositional modeling and timeline layer annotation alone, as we had not yet extended the schemata into an interpretative layer. As we shall see, the results of processing Collection A motivated us to extend the SIG and run further collections.

We used separate annotators for all three collections. The two annotators for Collection A were undergraduates in our engineering school and native English speakers, with little formal background in linguistics. We did not determine their literary expertise, but for

Collection B, which was to include interpretative annotation, we recruited a set of graduate students from our Department of English and Comparative Literature. While all were comfortable with computers, none had performed a semantic annotation task prior to the collection experiment. Collection B also included an undergraduate annotator with background in computer science, literature and creative writing; we also contributed two encodings for Collection B ourselves (but do not include these encodings in our discussion of user satisfaction or inter-annotator agreement below). One annotator underwent training, but did not complete any annotations. The median number of encodings completed per annotator was 8.5, but two annotators, A106 and A108, each completed 19 encodings. (In each case, we discarded a 20th encoding from the collection as it was developed during the training process.) Between A106 and A108 there are 18 parallel encodings of the same fables.

We conducted Collection C after the conclusion of Collection B; it involved three other undergraduates with interest and background in literature and writing. The composition of Collection C was partly determined by their recommendations and expertise.

A summary of all the texts included in DramaBank is shown in Table 5.1. The selected fables attributed to Aesop are reproduced in Appendix D. The remaining selections include:

- Four pieces of short fiction: "An Alcoholic Case" by F. Scott Fitzgerald, "The Gift of the Magi" by O. Henry, "A Good Man is Hard to Find" by Flannery O'Connor, and Chekhov's "The Lady with the Dog" (as held over from the study in Chapter 2). All deal with the various shades of motivation, intention and action that occur between independent agents (characters); notably, they are also somewhat ambiguous in these respects. Our annotators suggested the Fitzgerald and O'Connor pieces for this reason, finding Scheherazade to be interesting in and of itself as a tool for exploring one's understanding of a text. The Collection A annotators, tasked only with propositional and temporal encoding, did not have similar feedback.

- One news article from the *Wall Street Journal* ("Bahrain Protesters Say Security Forces Fire on Crowds," by Joe Parkinson), dealing with the actions and underlying motivations of the various actors in a social and political conflict.

Title	Author/Attributed To	Length	A	B	C
"The Dog and the Wolf"	Aesop (P134)	159	2	3	
"The Donkey and the Mule"	Aesop (P181)	182	2	2	
"The Eagle and the Roosters"	Aesop (P281)	82	2	2	
"The Farmer and the Fox"	Aesop (P283)	97	2	3	
"The Farmer and the Viper"	Aesop (P176)	71	2	2	
"The Fox and the Crow"	Aesop (P124)	136	2	2	
"The Fox and the Grapes"	Aesop (P15)	80	2	3	
"The Goose that Laid the Golden Eggs"	Aesop (P87)	100	2	3	
"The Lion and the Boar"	Aesop (P338)	116	2	2	
"The Lion and the Hare"	Aesop (P148)	103	2	3	
"The Lion In Love"	Aesop (P140)	128	2	2	
"The Milkmaid and Her Pail"	Aesop (no Perry index)	185	2	4	
"The Mouse and the Lion"	Aesop (P150)	176	2	3	
"The Serpent and the Eagle"	Aesop (P395)	129	2	4	
"The Shepherd's Boy and the Wolf"	Aesop (P210)	130	2	2	
"The Tortoise and the Eagle"	Aesop (P230)	122	2	3	
"The Wily Lion"	Aesop (P469)	162	2	3	
"The Wolf and the Lamb"	Aesop (P155)	127	2	3	
"The Wolf and the Shepherd"	Aesop (P234)	130	2	2	
"The Wolf in Sheep's Clothing"	Aesop (P451)	96	2	3	
"The Ape and the Fisherman"	Aesop (P203)	110		1	
"The Cat and the Mice"	Aesop (P79)	160		1	
"The Crow and the Pitcher"	Aesop (P390)	86		1	
"The Dog and His Shadow"	Aesop (P133)	86		1	
"The Fox and the Stork"	Aesop (P426)	111		1	
"The Shepherd and the Eagle"	Aesop (P2)	166		1	
"An Alcoholic Case"	F. Scott Fitzgerald, 1937	3,120			1
"Bahrain Protesters Say Security Forces Fire on Crowds"	Joe Parkinson, The *Wall Street Journal*, 2/18/2011	1,149			1
The Battle of Maldon	Unknown, 10th-11th century	1,562			1
Beowulf	Unknown, 8th-11th century	25,649			1
"The Gift of the Magi"	O. Henry, 1906	2,081			1
"A Good Man Is Hard To Find"	Flannery O'Connor, 1955	6,485			1
"The Lady with the Dog"	Anton Chekhov, 1899	6,663			3
Sled Driver (excerpt)	Brian Shul, 1992	1,282			1

Table 5.1: Makeup of the DramaBank corpus, including length in words, and the number of encodings procured for each text (from different annotators) in the three collections (A, B, and C). For fables attributed to Aesop, identifiers from the Perry index [Perry, 2007] are shown.

- One piece of contemporary nonfiction, an excerpt from *Sled Driver: Flying the World's Fastest Jet* by Brian Shul. This is a memoir of a U.S. Air Force pilot who flew the SR-71 Blackbird. The excerpt in question is an anecdote in which Shul, on a training mission with his Reconnaissance System Officer, reports feeling a sense of mastery of the air (and camaraderie with his RSO) for the first time.

- Two epic poems, originally in Old English: *Beowulf* [2] and *The Battle of Maldon*.[3] Both were suggested by undergraduate annotators who were medieval studies enthusiasts. Both are about large battles: *Beowulf* tells the story of the titular character's fights against Grendel and other foes; *Maldon* is an account of a real-life battle between the Anglo-Saxons and the Vikings.

Each collection involved approximately 2-3 hours of annotator training. The training consisted of an introduction to the SIG model, propositional modeling and the SCHEHERAZADE user interface. Annotators practiced by collaboratively working on an Aesop fable and were given brief written guidelines. For Collection A, we supervised the two annotators as they worked, but for the other collections we allowed annotators to work on their own computers and at their own paces. (SCHEHERAZADE will run on Windows, Macintosh and Linux machines with at least 1GB of memory and Java 1.5.) We also maintained a dialogue with the annotators by having them ask questions by email; we maintained a Frequently Asked Questions page with the most common concerns. After finishing each encoding, annotators completed a survey in which they reported their satisfaction with encoding process with respect to that story, as well as the amount of time that they spent. We compensated annotators by the hour, but asked them to enter a dialogue with us if they expected the encoding to take more than two hours. We set a hard limit of three hours for the fables but allowed for more time on the longer texts. In all cases, the annotator's goal was to instantiate SIG nodes, arcs and/or propositions until the generated feedback text and the interpretative graph represented his or her sense of the story's meaning as closely as possible.

Although the task is complex, all annotators became comfortable with the tool after

[2] Translation by Slade [2011] used with permission.

[3] Translated by Wilfrid Berridge.

a period of training. In the surveys, we had each user report the tool's ease of use on a 5 point Likert scale, with 5 representing "easiest to use," and list specific aspects of the story that they were unable to encode to their satisfaction. For Collection A, the two annotators reported usability scores of 4.25 and 4.30 (averaged over all 20 encodings for each annotator). As they became familiar with the tool, the required time to encode each fable (80 to 175 words) dropped from several hours to 30-45 minutes. The most frequently cited deficiency was the lack of non-concrete nouns, such as "fair" in the sense of a community event. We expanded the Scheherazade knowledge representation to include such nouns before engaging further annotators.

For Collections B and C, the average usability score was 4.35 on the same 5-point scale. The median time spent on each of these encodings was 1.25 hours (1 hour for Aesop fables in Collection B and 2 hours for the non-Aesop stories in Collection C, with only Collection B including propositional modeling). We also asked another Likert scale question: "On a scale of 1 to 5, how satisfied are you that the system has encoded your interpretation of the story?" The average scores for Collections B and C were 4.26 and 4.00, respectively. Overall, the annotators reported satisfaction with the process, although the task was more laborious for some annotators than for others (in a distant outlier, one annotator reported taking more than six hours to complete a single encoding). We reviewed each encoding and identified any instances where the annotator may have misunderstood an aspect of SIG annotation. While the interface uses feedback text to encourage proper annotation, and does not permit outright violations of the SIG model, in some cases certain arcs called for further review. The most common of these involved actualization, especially the distinction between a frame and its content (for instance, between ceasing a goal box, meaning the agent no longer desires the goal, and ceasing the content, meaning the agent has failed to reach its goal). When such cases occurred, we discussed the issues with the annotators, who either made modifications themselves or approved small changes by email. We believe that longer periods of formal training and more extensive written guidelines can greatly reduce such cases. In the case of Collection B, certain aspects of the SIG (most significantly, actualization logic) were refined according to annotator feedback.

The survey feedback for Collections B and C reinforces the notion of overall satisfaction with the annotation process and software tool. Interestingly, when asked for descriptions of what they found the system could not understand correctly, virtually all the Collection B responses dealt with the limitations of propositional modeling rather than the arc and node types of interpretative-layer annotation. These annotators found propositional modeling more complete than the Collection A annotators had found, due to the expansion of the knowledge base, but they still were challenged by missing frames. Particular issues included not being able to specify lengths of time, idioms such as "just desserts," missing lexemes such as "mutton," quantities, and "either-or" syntactic constructions. In several cases, annotators reported that they "fell back" on interpretative-layer annotation strategies when they were unable to produce a faithful propositional modeling of the concept at hand.

Among Collection C annotators, who used placeholder propositions and thus avoided the issues that Collection B annotators had faced, the most commonly reported issue was anxiety about engaging in the "mind reading" process and settling on a single interpretation. As several of these stories are ambiguous in terms of motivations of its characters, perhaps deliberately so, annotators were sometimes uneasy about modeling a single interpretation as authoritative. "The system is wonderful," wrote one annotator about the Chekhov story, "but it forced me to choose an interpretation in a few places, where I might have wanted more room for ambiguity.... I think the main problem here is just that Chekhov's style is very difficult to break down into the goal-oriented structure, since large swaths of the story consist of characters looking over the water and realizing beauty and pointlessness." Clearly, while the theory of mind is one approach to a text, it is not the only approach, nor is it the best approach for every text; while allowing for plural encodings may mitigate such anxiety, this effect must be formally considered. In other stories, annotators reported satisfaction with the SIG perspective: "Anglo-Saxon poetry is very goal/plan-oriented and so works well with the psychological analysis program," wrote one annotator about *The Battle of Maldon*. Regarding "An Alcoholic Case," the same individual wrote, "The system worked well for this story, which involved a protagonist clearly wavering between two beliefs with two sets of clearly differentiated associated plans of action/goals—something that works very well with this system."

The large annotation of *Beowulf* merits special mention. This encoding involves some 476

Title	Nodes	Arcs	Coverage (Words)	Coverage (%)
Aesop (average)	33.7	41.6	131	100.0
"An Alcoholic Case"	30	53	609	19.5
"Bahrain Protesters Say Security Forces Fire on Crowds"	35	53	246	21.4
The Battle of Maldon	39	62	406	26.0
Beowulf	476	413	12,695	49.5
"The Gift of the Magi"	46	61	422	20.3
"A Good Man Is Hard To Find"	100	169	1,562	24.1
"The Lady with the Dog" (average)	74.7	102.3	1,261	19.0
Sled Driver (excerpt)	37	73	628	49.0

Table 5.2: Characteristics of the DramaBank corpus (Collections B and C).

nodes and 413 arcs. It excludes certain digressions that do not further the goal progression of the story (such as historical background), though when these inner stories do take place, the annotator was careful to include a character's motivation for conveying such information. Of the story's nearly 26,000 words, approximately 12,700 words (50%) are covered as directly relating to a goal/plan/intent model of the text. The annotation process took slightly more than 15 hours. This student, who was already interested in the poem and in medieval studies in general, found that the experience of SIG annotation heightened her appreciation: "With this modeling of goals, desires, intentions, and plans," she wrote, "we see that Beowulf is an intricately worked poem—more carefully and logically crafted than has been previously thought."[4]

Table 5.2 shows some characteristics of the encodings in Collections B and C. For each

[4] To follow this thread, we met with this annotator and with a medieval studies specialist in the Department of English and Comparative Literature. The student reported that SIG annotation reinforced the coherence of the poem by concretizing its structure. It enhanced her understanding of the text to document internal connections, such as when a character has a goal that is much later supported or undercut by other characters; this brings out an "emotional undercurrent" and emphasizes the integration of the text as a unified discourse rather than a series of disconnected episodes. The medievalist agreed that SIG annotation can be an important pedagogical tool, and thought that her students' papers would improve if the students used Scheherazade to mark up the text as they read it, but also argued that the theory-of-mind interpretation that motivates the SIG has a strong modernist bias. *Beowulf* and other older texts are traditionally given readings that emphasize objects, places, and especially form (structure and metre), as opposed to agency.

discourse, the number of nodes and arcs the annotator modeled in the interpretative and timeline layers are shown (excluding temporal relations and *in*, which is structural). The latter two columns indicate the amount of source text, in the form of a word count and a proportion, respectively, that is represented in a Text node and linked to the remainder of the encoding. The properties of the Aesop encodings and the three encodings for "The Lady with the Dog" are averaged. While there are too few encodings to control for genre or for differences between annotators in this metric, we can see that the "dense" Aesop texts are completely covered—as we asked annotators in Collection B to encode as much of the fable as possible—and the other texts range between 19.5% and 49.5% coverage. In fact, the data are clustered around two peaks, with the literary fiction and news article each receiving around 20-25% coverage, and the memoir joining *Beowulf* in the 50% range. The relationship between genre and the complexity of the resulting encoding is an open question pending further collection.

The following two sections continue this discussion by introducing algorithms for determining the similarity between two story encodings. We will apply these measures to determine inter-annotator agreement for those texts that were covered by multiple annotators. We first discuss an approach for finding similarity at the propositional level alone, and then two algorithms for analyzing and comparing the thematic content of complete encodings.

5.2 Propositional and Temporal Overlap

For Collection A, two annotators each created story graphs for a set of 20 of Aesop's fables. These encodings only include propositional and temporal information, not interpretative-layer content. Determining the similarities and differences between any two stories in this collection is a matter of devising and adapting techniques for finding the semantic distance between two temporally structured lists of propositions.

In order to guide our development of such an an algorithm, we note that we expect the highest similarity score to occur between two parallel encodings of the same story ("homogeneous" encoding pairs). As Collection A includes only paired encodings, we can use it as training data for developing a metric that finds *propositional paraphrases*—these occur when the same source-text phrase or concept is annotated in slightly different forms, both semantically equivalent to the phrase or concept, by different annotators. By developing a technique for measuring the distance between two propositions, and evaluating its efficacy at separating paraphrases from non-paraphrases, we can arrive at a measure of narrative similarity for heterogeneous encoding pairs (under an assumption where two encodings that appear likely to be homogeneous are more similar than two encodings that do not).

In this section, we present an algorithm for identifying the similarities between two SIG timelines, each of which is a sequence of propositions. We also describe an evaluation in which our alignment algorithm outperforms a word-overlap baseline for identifying propositional paraphrases.

Paraphrase identification typically involves algorithms designed to find segments of text with similar meaning. Along with parallel corpus alignment, this is a necessary step for tasks such as text-to-text generation [Barzilay and Lee, 2003] and question answering [Lin and Pantel, 2001], in which common ideas must be culled from a set of related texts. Though much progress has been recently made toward learning the syntactic and lexical variations behind many paraphrases [Lepage and Denoual, 2005; Fernando and Stevenson, 2008], paraphrase detection in symbolically annotated forms of corpora is less-often studied. These corpora include large-scale annotation projects such as the Penn Treebank, Propbank and Ontonotes [Pradhan *et al.*, 2007]. While these are instrumental for creating trained language models for paraphrase detection, they are not themselves parallel corpora. One similar project by Halpin et al. [2003; 2004] involves the automatic analysis of a story that is encoded in a propositional form (via an intelligent tutoring system for children). Their approach to generating feedback depends upon a method for determining the degree of plot similarity. The problem of *event coreference* [Danlos, 1999] considers semantic similarity and expected entailment when determining if two sentences refer to the same narrative event.

Our alignment problem is also distinct from prior analogues due to the nature of the representation. Our methodology is novel in its synthesis of several annotation goals and its focus on content rather than expression. We aim to capture an entire narrative *fabula*—that is, the content dimension of the story, as opposed to the rhetorical presentation at the textual surface (*sjužhet*) [Bal, 1997]. To this end, our model incorporates formal elements found in other discourse-level annotation projects such as Penn Discourse Treebank [Prasad *et al.*, 2008] and temporal markup languages such as TimeML [Mani and Pustejovsky, 2004]. As we described in Chapter 4, every element of the representation is formally defined from controlled vocabularies. The verb frames, with their thematic roles, are adapted from VerbNet [Kipper *et al.*, 2006]. When the verb frames are filled in, to construct event propositions, the arguments are either themselves propositions or noun synsets from WordNet [Fellbaum, 1998]. Annotators can also include stative elements and modifiers (with adjectives and adverbs culled from WordNet). Crucially, each proposition is bound to a state in the story's main timeline, a linear sequence of states. Annotators can create alternate timelines to encode multi-state beliefs, particularly about the past or future.

The 40 encodings in Collection A cover a total of 574 propositions, excluding those in alternate modalities. The fables average 130 words in length (so the annotators created, on average, one proposition for every nine words). However, between the two annotators there are only 29 pairs of identical propositions. In other words, even though both annotators created parallel reproductions of the same source texts under the same conditions, there is only 10% overlap between their efforts. Table 5.3 shows representative examples of the trend in which the same concept was modeled as separate, yet equally reasonable propositions.

The causes for the differences between homogeneous encoding pairs fall into three categories:

1. **Subjective interpretations.** Some differences represent individual interpretations of the story. For example, only one annotator may have inferred in the second example of Table 5.3 that the donkey died as a direct result of the mule's ignorance of his plea for help. In the case of comparing encodings of different stories, substantive differences between stories would also fall in this category. (Note that Collection A did not include the interpretative layer of the SIG, which would have offered a separate, preferred mechanism for indicating such causality. In its stead, the Collection A procedure allows annotators to indicate causality by using a `because` modifier that takes two other propositions as arguments.)

Source text	Encoding/Feedback Text, Annotator 1	Encoding/Feedback Text, Annotator 2
When once, however, [the lion] was thus disarmed, the Cottager was afraid of him no longer, but drove him away with his club.	stop(cottager, fear(eat(lion, cottager))) chaseAway(cottager, lion, daughter). with(chaseAway$_1$, club). The cottager stopped fearing that the lion would eat the cottager. The cottager chased the lion away from the daughter with the club.	stop(cottager, fear(dangerous(lion))) driveFrom(cottager, lion, cottage) with(driveFrom$_1$, club) The cottager stopped fearing that the lion was dangerous and drove him from the cottage with the club.
The Mule paid no attention to the request. The Donkey shortly afterwards fell down dead under his burden.	¬listen(mule, donkey) die(donkey) The mule didn't listen to the donkey. The donkey died.	ignore(mule, donkey) die(donkey) because(die$_1$, ignore$_1$) The mule ignored the donkey. The donkey died because the mule had ignored the donkey.
[The fox] walked away with an air of dignity and unconcern.	walkAway(fox, trellis) unconcernedly(walkAway$_1$) The fox unconcernedly walked away from the trellis.	walkAway(fox, trellis) begin(dignified(fox)) begin(unconcerned(fox)) The fox walked away from the trellis, began to be dignified and began to be unconcerned.

Table 5.3: Three propositional paraphrases from our corpus of encoded narratives. The latter two columns show the propositions created by the annotators, as well as the feedback text generated by our system to guide their annotations.

2. **Propositional paraphrases.** Due to the aggregation of synonyms and various morphological variations in the knowledge base, there are multiple correct ways to express the same concept using different syntactic constructions. For example, one annotator indicated "The lion is asleep" as a stative, while another chose "The lion was sleeping" as a progressive action.

3. **Sampling error**. Some differences are introduced by the tool itself during the collection process. One annotator may have felt more comfortable with *fabula* extraction, or was more able to explore advanced features of the tool such as setting action modalities.

The third source of error, as always, can be reduced through greater training and iterated user interface design. The first source of error, differences in subjective interpretations, illustrates the need for the interpretative layer of the SIG. In Collections B and C, the encoding timelines represent "what happens," and the interpretative subgraph indicates "why it happens" and "why it matters." Collection A conflates these aspects in the timeline alone, so we do not presently address interpretative content in Collection A. Our focus here is on reducing the second source of error; a method for normalizing propositional paraphrases will separate what is intended to be the same content from what is intended to be different content. This will not only allow us to better determine inter-annotator agreement in homogeneous pairs (since we assume that "intended" agreement is greater than 10%), but provide a less noisy metric for assessing the similarities between heterogeneous pairs. As such, we turn now to an algorithm for identifying and normalizing propositional paraphrases.

5.2.1 Paraphrase and Alignment Algorithms

Of the 20 story pairs in Collection A, we used 3 to develop an algorithm for detecting propositional paraphrases, and set aside the remaining 17 for testing. We implemented a two-step approach that first finds a list potential paraphrases among all MxN pairs of propositions between two encodings of the same discourse, then refines the list using the ordering property (such that propositions are more likely to be paraphrases of the same story concept if they are at similar points in their respective tellings). The second step leverages the parallel nature of homogeneous encoding pairs, and is <u>not</u> later applied for finding similarities across heterogeneous pairs; we describe it here as a refinement specific to a sub-task of homogeneous encoding alignment. In more detail, the steps are:

1. **MxN pairwise semantic distance measurement.** The semantic distance between every pair of propositions between the two encodings is automatically rated on a scale from 0 (completely disjoint) to 1 (completely identical). This returns a ranked list of possible paraphrases that serves as a cost/benefit assessment for the following step.

2. **Story alignment.** We refine the set of "candidate" propositions pairs by performing an iterative, constraint-based alignment between the two encodings. By using the most

likely paraphrases as anchors, then progressing to less closely related proposition pairs that fall between the anchors, we derive a complete alignment and return an overall similarity measurement for the two encodings.

Semantic distance heuristics

We identify three features for predicting the semantic distance between any two propositions among the MxN pairs present in two encodings: information content, morphology, and synonymy/antonymy.

Information-content relatedness. Following prior work [Budanitsky and Hirst, 2001], we considered finding the nearest common ancestor for two proposition types in the WordNet hypernym tree. However, WordNet is known to have an uneven hypernym tree, such that simple edge-counting is an unreliable way to measure the degree of generality between a lexeme and a hypernym ancestor [Resnik, 1999]. Instead, we adopt the information-theoretic definition of similarity described by Lin [1998] by using a trained model included in an off-the-shelf WordNet similarity library. Our score function also considers the overlap between the predicate's arguments (i.e., the attributes, as previously suggested by Tversky [1977]), by recursively scoring each argument (a noun or proposition) using the same formula. Both measures contribute evenly to a final score $s(a, b)$ between 0 and 1, with 1 being reserved for identical propositions, so that the impact of nested propositions telescopes downward in influence:

$$s(a, b) = \frac{c(a, b)}{2} + \frac{\sum_{i=1}^{r(a,b)} s(p(a, b, i))}{2r(a, b)} \tag{5.1}$$

where $c(a, b)$ is information-content similarity between two predicates; if the WordNet synsets are not covered by the similarity library, we fall back on counting edges in the hypernym tree. Specifically, in this case, $c(a, b)$ is given by $\frac{1}{1+h(a,b)}$ where $h(a, b)$ is the average path length from the two predicate synsets to their nearest common hypernym (or ∞ if that is the root type). The remaining terms deal with finding the similarities between sets of arguments: $r(a, b)$ is the size of the union of the thematic roles covered among the arguments to both propositions, and $p(a, b, i)$ retrieves the two nested propositions which serve as arguments in a and b for some thematic role i. In other words, half the score is dictated by the predicate distance, and half is the mean of the similarity scores of the parallel arguments (recursively evaluated).

Morphology. To improve recall in morphological cases such as the "unconcerned"/ "unconcernedly" distinction in Table 5.3, we follow the rule-based approach offered by [Jacquemin, 1997] for identifying adverbs and adjectives that convey the same meaning (such as the appending of *-ly*). For irregular morphological connections, we incorporate the Categorical-Variation Database provided by Habash and Dorr [2003]. This resource, trained on resources such as the Brown corpus [Kucera and Francis, 1967], features clusters of lexemes with their categorical variants. We leverag this in a heuristic that attempts to match a stative proposition with a modifier attached to an event (verb) proposition. If such a match exists, the propositions are considered partial paraphrases, and scored accordingly.

Synonymy/antonymy. As we have seen, many cases of propositional paraphrasing go beyond morphology in nature. Previous work has examined the prevalence of lexical paraphrases, such as the substitution of "high" for "tall" (which occurs in two encodings of "The Fox and the Grapes"). Following Zukerman et al. [2003], we use WordNet's synonymy relations as valid lexical substitutions. Synonymy links are more prevalent in adjectives and adverbs than verbs, for which the hyponymy formula described above is preferred.

Because our annotation scheme supports negations, paraphrases sometimes take the form of antonymous relationships, which are more difficult to detect. In Table 5.3, the negation of "listen" in "The mule didn't listen to the donkey" is a propositional paraphrase of "The mule ignored the donkey," which the opposite annotator preferred. "Listen" and "ignore" are not strictly antonymous, but in this context one is assumed to be equivalent to the lack of the other. To recognize such paraphrases, we employ the broad-coverage semantic network of verbs offered by Chklovski and Pantel [2004]. This publicly available resource, culled from carefully crafted search engine queries, includes derivations such as that "ignore" is antonymous to "perceive" in common usage. We then invoke inference in our own graph representation: since "perceive" is a direct hypernym of "listen," "ignore" and "not listen" are inferred to be equivalent. While this resource is somewhat noisy, we have found that it has boosted the recall of paraphrase detection without significantly harming precision.

Alignment

To complete an overall story similarity measure that normalizes propositional paraphrases, we broaden our view from single propositions in isolation and instead consider the temporal structure of each encoding. In the case of homogeneous encoding pairs, it is not likely that a proposition from the beginning of one encoding is based on the same sentence as a proposition from the end of the opposite encoding. However, the total ordering of all propositions is *not* guaranteed to be the same between such encodings, because within a single story state (time slice), annotators can place propositions in any order they wish without altering the semantics of the timeline. For example, when the fox in Table 5.3 "walks away with an air of dignity and unconcern," the act of walking and the stative of possessing dignity begin simultaneously; either order is correct in terms of machine understanding. In this respect, propositional alignment is analogous to parallel corpus alignment in machine translation, where variations in order can fall between alignment anchors. These anchors can be based on features such as sentence length [Gale and Church, 1993] or cross-cutting lexemes [Brown *et al.*, 1991; Barzilay and McKeown, 2001]. In our case, we use aligned states as anchors but allow for variation within states and among states that fall between anchor points.

The alignment algorithm chooses individual pairs of propositions, one from each encoding, as alignment points. Each selection adds to a vector of constraints for subsequent selections to satisfy. In more detail, alignment occurs in four stages:

1. An empty set of *alignment constraints* is initialized. An alignment constraint either forbids proposition matches from certain story states, or requires that identifiers in one story (such as declared characters) can only be bound (made analogous) to identifiers in the opposite encoding.

2. All the MxN proposition pairs between the two encodings are ranked as candidates for alignment. The ranking depends on both semantic distance and the relative positions of the propositions in their respective stories (similarly positioned propositions are ranked higher). Proposition pairs that violate any of the set of alignment constraints are avoided, as are pairs that do not meet a minimum degree of closeness. If no

suitable candidates are found, the algorithm terminates.

3. The highest-scoring proposition pair is declared an alignment point. Two new alignment constraints are added based on this selection: First, events that occur following one proposition are constrained from mapping to events preceding its counterpart (thus presenting a conflict in state ordering). Second, the story elements found to be analogous are constrained to remain bound. (A "Man" in one story, for example, cannot map to "Father" in the opposite story if it had already been mapped to a separate "Brother" character.)

4. The algorithm repeats from Step 2 to repeat the selection process until no further pairs of propositions can be aligned.

Figure 5.1 shows the alignment algorithm running on a segment of both encodings of "The Donkey and The Mule". Each column represents a timeline, with boxes signifying states. Every state contains one or more propositions that occur during that point in time. The solid arrows between the columns indicate the alignment points; the numbers give the order of alignment. "The donkey dies" is the first attachment because it is repeated verbatim in both stories at nearly the same point. "The donkey ascends the mountain" and "The donkey climbs the mountain" follow, because "climb" and "ascend" are close in the hyponymy tree and their respective arguments are identical. At this point, the first two states are considered aligned, as are the fourth and fifth states in Timeline 1 and Timeline 2, respectively. The algorithm would thus exclude any alignment between the first Timeline 1 state and the last Timeline 2 state. In subsequent iterations, other alignments follow in increasing order of semantic distance. "The muleteer is uncertain" is not mapped to any proposition on the right, as it was not modeled by the second annotator.

The dotted arrows in Figure 5.1 indicate gaps in the alignment. These are propositions that were not rated as paraphrases by the semantic distance heuristic, but fall between state alignment anchors. In this case, the latter alignment gap contained a valid inference: the mule would "rue" not helping the donkey with his load because a bit of selflessness would have prevented him from taking on the entire load. (In the story, the donkey dies from his burden shortly after appealing to the mule for help.) In general, alignment gaps are

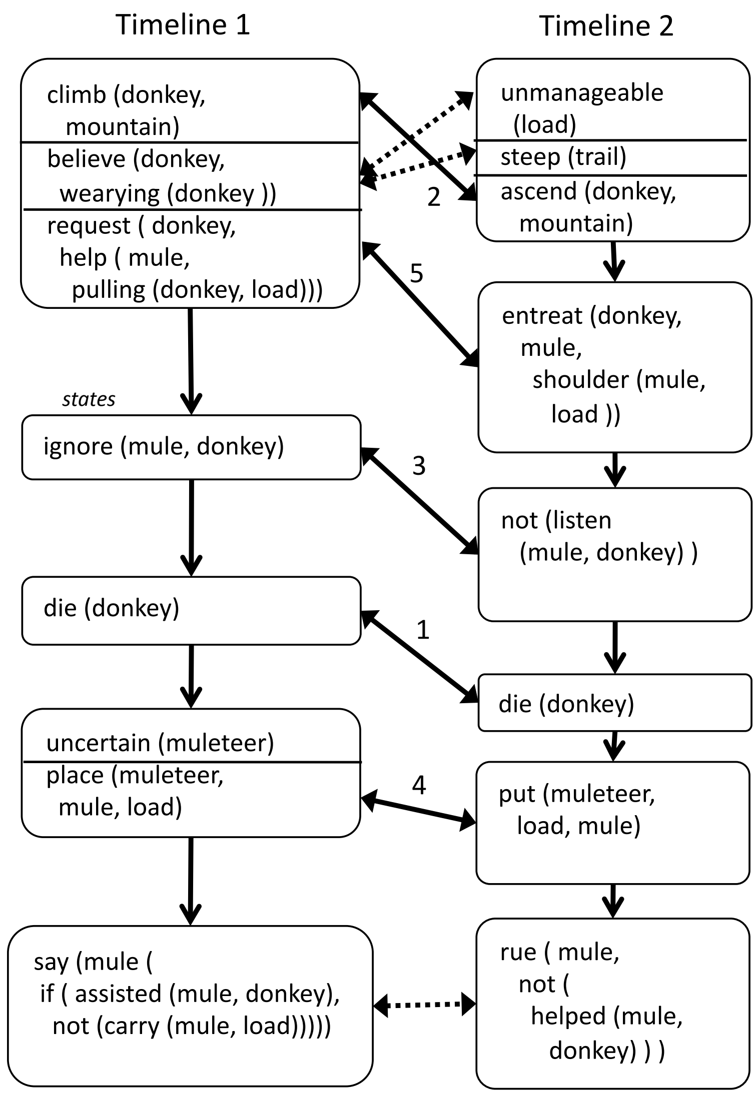


Figure 5.1: Aligning two encodings of "The Donkey and The Mule". Boxes represent time states; sections within the boxes hold individual propositions. Numbered arrows show the proposition pairs selected for alignment in order of their selection.

correlated with paraphrases that were not detected due to erroneously low similarity scores: Of the 462 proposition pairs in alignment gaps in Collection A, our evaluators (described in the following section) rated 33% as either paraphrases or inferentially related. By comparison, they only rated 15% of all proposition pairs similarly. However, this correlation is not strong enough to declare a paraphrase for every gap; we found that this approach reduced our overall accuracy when comparing homogeneous encoding pairs, and naturally would not be appropriate for heterogeneous pairs at all.

The output of our algorithm is a ranked list of likely paraphrase alignment points, as well as a list of suspected paraphrases that fall in the alignment gaps. By setting a confidence threshold, we can classify proposition pairs as simply "paraphrase" or "not paraphrase," and evaluate the accuracy of this method.

5.2.2 Evaluation

In Section 5.4, we will present an evaluation of the effectiveness of this approach for the task of finding meaningful similarities across opposing fables (that is, in heterogeneous encoding pairs). As an initial check, though, we conducted a separate evaluation specifically on the issue of detecting propositional paraphrases between homogeneous encoding pairs. We shall see in the final evaluation that the propositional similarity approach errs on the side of being specific but not sensitive to thematic connections; here, we will specifically evaluate both specificity and sensitivity on cases where very high similarity is to be expected between a pair of encodings.

To evaluate our alignment algorithm against the judgments of human annotators, we compiled a list of all pairs of propositions between homogeneous encodings in Collection A. From this we created an evaluation set that included pairs of sentences, as generated by our system as feedback text (Section 4.4). To reduce the work load on our annotators, we excluded permutations where the relative positions of the sentences in their respective encodings differed by more than 40%; these were assumed to be non-paraphrases. The trimmed set included 2,700 sentence pairs. While an evaluation based on propositions in their logical form would have been more direct, we felt that we would elicit better overall judgments from non-expert annotators by exposing them only to natural language (notwithstanding error that may have been introduced by the feedback text generation module).

Our survey asked annotators to mark each sentence pair as best fitting one of four categories: paraphrase (conveying the same information), partial paraphrase (one sentence conveying the same information as part of the other), inference (one sentence able to be reasonably inferred from the other without being a direct paraphrase), and unrelated (conveying different information). We employed Amazon's Mechanical Turk distributed work service to gather three annotations for each sentence pair. Of 2,700 sentence pairs, 1,554 had a three-way consensus and 1,027 showed a two-to-one majority, which we considered canonical. We excluded from our evaluation the remaining 4.4% of cases in which no majority or consensus emerged against which we can measure our performance. We also re-sampled 10% of the original annotations, as some annotators did not properly follow instructions (as determined from their very high rates of their voting against a majority—noise reduction is a typical concern with Mechanical Turk).

According to the ground truth culled from these survey results, there are 248 paraphrase connections (conflating partial and total paraphrases) among the 574 propositions in Collection A. 86% of propositions were determined to convey the same information as at least one counterpart; the remaining 14% of propositions were either involved in more indirect (inferential) variations, or were unmatched "singletons" where one annotator chose to encode a concept that the other annotator felt was non-notable or commonsense knowledge (such as the ownership of a character over his body parts).

We compared our system's performance against a baseline of word overlap: The Jaccard index is defined as $\frac{|A \cap B|}{|A \cup B|}$ for the sets of words present in sentences A and B. We calculated the overlap score over the feedback text versions of the propositions, so that they would include the same verb and noun lexemes as intended by the annotator (rather than the source text segments themselves, which are more rhetorical and less structured).

Both our algorithm and the Jaccard baseline express similarity in a continuous scale from 0 to 1. We picked a confidence threshold between "paraphrase" and "not paraphrase" for each approach by maximizing its F-measure performance against the manual ratings (predicting which pairs are rated as "paraphrase" or "partial paraphrase"). We used the

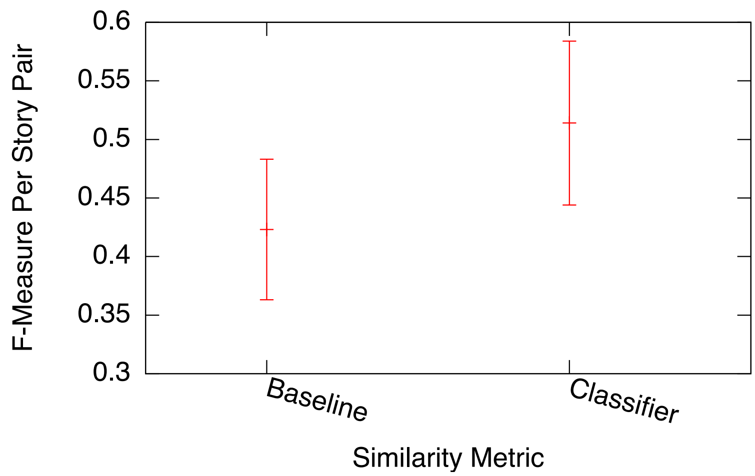


Figure 5.2: Performance of our algorithm on identifying propositional paraphrases in Collection A compared to the Jaccard word-overlap baseline, as distributions of F-measures over 17 homogeneous encoding pairs in the test set. Error bars indicate 95% confidence.

3 development encoding pairs for this tuning, and then tested against the remaining 17 encoding pairs. The distributions of F-measures (one for each story pair) are shown in Figure 5.2 with error bars indicating 95% confidence. On average, our approach outperformed the baseline by about 9 points. However, we are not able to show significance at the 95% level, with p=.054 on the paired Student's *t*-test.

False negatives were the main source of error; our analysis indicates that we face the familiar challenge [Palmer *et al.*, 2007] of recovering intent when annotators choose from among the many fine-grained distinctions present in WordNet and VerbNet. While many of the finer verb distinctions were removed during the adaptation process, broader distinctions remained. For instance,"sets a trap" and "lays a trap" are not related, according to our knowledge base, because the nearest common ancestor to the two verbs is the root event. Other missed paraphrases involve more subtle inference, such as "the dog lies" and "the dog rests," or "the cornfield is destroyed" and "the farmer loses the corn." False positives were less of an issue; one potential improvement is to adjust the propositional similarity metric to weigh certain thematic roles more heavily than others.

One drawback to our approach in general is the paucity of training data that precluded statistical tuning of the scoring formula (Equation 5.1). While we relied on other models which had been statistically trained, such as Lin's information-theoretic lexical similarity metric, Collection A is small compared to the lexical and syntactic range of the model that it utilizes. We believe this imbalance is limited to the propositional aspect of the SIG. While the schemata of the interpretative layer also supports an effectively unlimited range of possible configurations (that is, patterns of nodes and arcs that can become indefinitely large), the far smaller "vocabulary" makes it more amenable to bottom-up approach. We describe such an algorithm in Section 5.3.2.

In a larger context, we interpret from these results that when a pair of SIG encodings includes propositional modeling, we are able to leverage that data in order to find cases of very similar material along the lines of "what happened" in a declarative vocabulary. In Section 5.4, we will take this a step further and determine whether the detection of such cases is helpful to the greater cause of finding meaningful thematic similarities and analogies between heterogeneous encoding pairs. First, though, let us examine Collections A and B using the approach we have just described, first to determine inter-annotator agreement in homogeneous encoding pairs, then to consider intersections between heterogeneous pairs.

5.2.3 Corpus Analysis

Our purpose for paraphrase detection and alignment has been to reduce the impact of minor differences in encoding style, which in turn will allow us to better isolate substantive differences between encodings in the timeline layer. In the case of parallel encodings of the same story, these differences could indicate the shadings of each annotator's subjective interpretation of the *fabula*, a potentially useful result. However, for the larger goal of finding similarities and analogies in heterogeneous story pairs, we would like the similarity between redundant encodings to be high—if the system cannot reliably detect areas of agreement between encodings of the same story, it cannot detect analogies between stories.

In practice, this method indicates a significantly higher similarity for same-story encodings than for different-story encodings. In Collection B, which we did not use as a development corpus for this algorithm (as it had not yet been collected), there are 40 homogeneous pairs of encodings and 1,383 heterogeneous pairs. The far greater coefficient of similarity determined by our algorithm for heterogeneous pairs is seen in Figure 5.3; the increase is significant to $p<.001$ under the two-tailed Student's t-test.

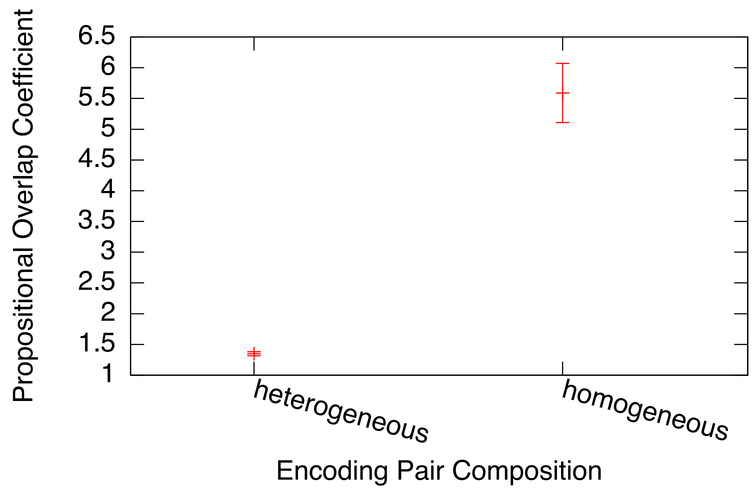


Figure 5.3: The propositional similarity algorithm can easily separate homogeneous encoding pairs (inter-annotator encodings of the same story) from heterogeneous pairs of different stories. Error bars indicate 95% confidence.

Among those heterogeneous pairs are certain instances where this approach does reveal significant overlap between stories. While the collection process for propositional modeling is labor-intensive, it exposes a high-precision set of "intersection" propositions which, by means of hyponymy trees, are only abstract enough to cover both contributors to a proposition pair (Equation 5.1). Table 5.4 gives example intersection propositions from the ten most similar heterogeneous encoding pairs in Collection B according to this metric. Note that for purposes of finding heterogeneous similarities, we have "flattened" modal content, so that a hypothetical event in an alternate timeline in one story (typically, a goal or a fear) may match an actual event in the opposite story. We also relax the strict binding constraint that prevents characters from mapping to more than one counterpart. We shall see that despite these allowances, the accuracy (precision) of this approach is still the highest of the three similarity metrics we attempt in this chapter. We will restore these constraints in the other two similarity metrics, which operate on the entire SIG (including the interpretative layer).

The story pairs that score highly for propositional overlap feature verbs and animal species that are closely related in the WordNet hypernym tree (such as the cat and the lion both being felines). The third example, where "The Dog and the Wolf" is a close match

Story 1	Story 2	Propositional Overlap
"The Serpent and the Eagle"	"The Wolf in Sheep's Clothing"	In both stories, a vertebrate changes; an organism kills an organism; a person thinks; a thing moves; a vertebrate eats a character; an organism decides to act; a vertebrate travels to a character; an organism consumes; a person consumes; an organism protects a character; a person moves an object; an organism kills an organism; an organism prevents something; an organism seizes a vertebrate.
"The Cat and the Mice"	"The Mouse and the Lion"	In both stories, a character states something; a placental mammal travels; a placental mammal says to a feline that a thing has some property; a placental mammal says to a feline that a character is or is not able to act; a thing acts; a mouse states something; a placental mammal hears about some event; a feline acts; a character tells another to do something; a mouse perceives something; a character puts something somewhere; a character states that a feline has some property; a placental mammal interacts; an organism catches a character.
"The Dog and the Wolf"	"The Lion and the Hare"	In both stories, an animal lies; a carnivore acts toward an animal; a carnivore sees something doing some event; a carnivore arrives; a carnivore returns to a thing; an animal says that a thing has some property; a carnivore finds that an animal has some property; a carnivore eats.

Table 5.4: Three of the pairs of Collection B encodings with the highest degrees of propositional overlap.

to "The Lion and the Hare", is an interesting case, as these stories are actually closely analogous: In both cases, a predator is within striking distance of catching its prey, but then stops, believing it can increase its bounty by holding out for better returns later. (In one case, the dog convinces the predatory wolf to return once the dog has fattened up; in the other, the lion is distracted by another animal which it thinks would make for a better meal.) In both cases, the predator then returns to the original prey only to find that the prey has become unavailable in the interim, yielding a lesson about overreach and delay. (Zafiropoulos also identifies these stories as analogous, in that both advocate "the immediate satisfaction of the protagonist's interests [Zafiropoulos, 2001, 59].) The propositional similarity between these two stories, though, does not get at this moral directly. The use of certain verbs to indicate opportunity, delay and resumption ("lies," "sees," "returns," "finds," "eats" as a hypothetical) could be seen as latent indicators of the larger ethical point that the stories share; however, this effect only occurs when the two analogous stories are in very similar or identical domains. In a sense, *Sled Driver* is also a story about taking opportunities when they arise: The SR-71 pilot hears Navy aviators bragging about their speeds to civilian air traffic controllers; while his aircraft is one of the fastest ever built, the narrator has only a few seconds to call down and settle the matter in his favor, and the officer flying with him is nominally in charge of radio communications. The verbs listed above do not figure into this story in any meaningful way, even though the message is analogous.

Even in the homogeneous Aesop domain, it is difficult to use the propositional overlap metric of story similarity to discover recurring themes about actions, motivations and consequences. We attempted to carry out such an analysis of Collection A by asking our annotators which characters "win" and "lose" (experience an overall positive or negative affectual impact) in each story. Using these annotations as training data, we implemented an algorithm which gathers all of the actions that involve "winning" characters and all those that involve "losing" characters, and by holding out a story with cross-validation, predict whether the characters in that story win or lose by measuring the similarity of their experiences to those of known winners and losers. We found that the held-out actions were not significantly closer in propositional similarity to either of the training sets. Our error analysis suggested that data sparsity was a likely reason. Even in the Aesop domain alone, for instance, there are many ways to indicate that a character kills another character—"devoured," "inflicted a fatal bite" and so on—whose predicates are not related in WordNet or in Lin's information-theoretic model of similarity. If determining affectual impact were the lion's share of our overall goal, so to speak, we might have at this point followed in the footsteps of those who have trained larger statistical models of emotion and sentiment in language, such as Wiebe et al. [2005] and Tackstrom and McDonald [2011]. We were, however, more interested in preserving our focus on narrative discourse, with its integration of not only sentiment and affect but also plans, goals, time manipulation, and so forth. This motivated us to develop the interpretative layer of the SIG and compile Collections B and C, which utilize it.

5.3 Interpretative Similarities and Analogies

As we mentioned above, Collections B and C involve encodings which include the interpretative layer of the SIG model. We defined the topology of the interpretative layer in Section 3.3.2: Instead of a linear sequence of propositions, the schemata calls for frames that represent intentional states on the part of agents; inside of those frames are goals and beliefs that the agents may hold. The interpretative layer can also include "hypothetical" propositions that exist outside of any belief frame. The arcs that connect these goals to the timeline include *interpreted as*, *implies* and *actualizes*, which (with differing connotations) assert that an interpretative-layer node is true ("actualized") at the respective point in story time. Another arc, *ceases*, has the opposite effect, asserting that a particular node is untrue at the respective timeline state, and unlikely to become actualized at a later point in story time. For example, an agentive character may have a goal frame; from the perspective of one tracing the main timeline from the first state to the last state, the goal within the frame "begins" life as a hypothetical and remains so until it is actualized or ceased by a timeline node at the point where the goal either transpires or is definitively prevented. This indicates a positive outcome and a negative outcome, respectively. Frames themselves can also be actualized and ceased, to indicate that a character does or does not have a certain goal or belief. Goal nodes can be linked into sequential plans (Section 3.3.2.3). Affect nodes indicate the affectual impact of interpretative content on an agent (Section 3.3.2.7).

The purpose of the interpretative set of discourse relations is to offer a controlled vocabulary for constructing a theory-of-mind reading of a narrative text, one that allows for the representation of certain key elements without introducing the complexity of a total semantic understanding of the story-world. (We explained our motivation for adopting goals, plans, beliefs, intentions, and affectual impacts as our interpretative "primitives" in Section 3.2.) While interpretative-layer annotation is accessible enough for non-technical annotators to use, as we saw earlier in this chapter, it is still formal enough for the limited degree of logical inference that relates to those aspects of narrative logic which we favor. One can, for example, deduce that if a timeline event prevents a hypothetical, and that hypothetical was a necessary condition for a character to reach its goal, then by a certain transitive property the goal has also been prevented.

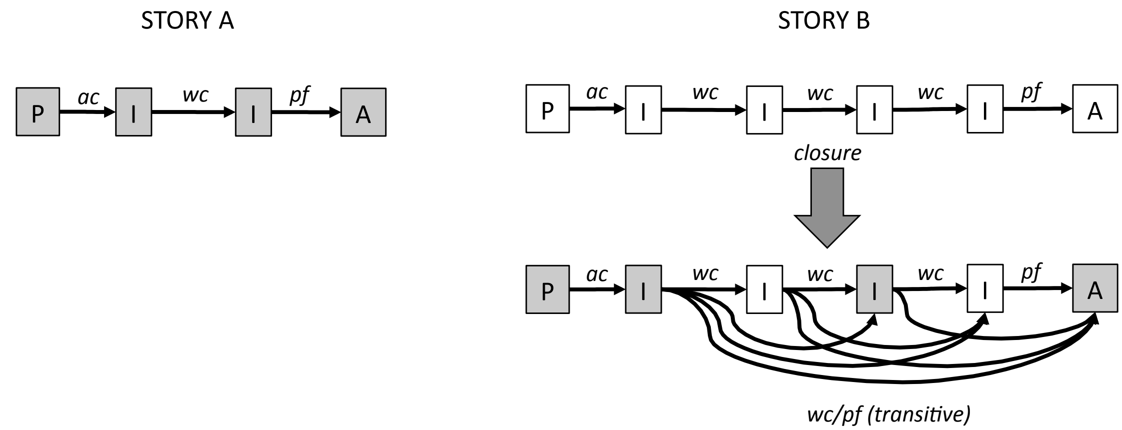


Figure 5.4: The application of closure rules allows us to procedurally identify isomorphic subgraphs (shaded nodes) between two SIG encodings that are otherwise disjoint, such as one with a two-step plan and one with a four-step plan.

Indeed, we have in practice defined a series of what we call **closure rules** that insert arcs into a SIG encoding where the interaction of two or more other arcs triggers a transitive entailment. There are closure rules that govern time (if A is followed by B and B is followed by C, A is followed by C); hypothetical causality (if A would cause B and B would cause C, then A would cause C); actualized causality (if A caused B and B caused C, then A caused C), affectual impact (if A would cause B and B would have a detrimental effect on C, then A would have a detrimental effect on C), intention (if A is an action that attempts to cause goal B, B would cause C, and both B and C are in the goal frame of A's agent, then A is attempting to cause C), and more. We give a complete and formal description of these rules in Appendix C.

From a graph theoretic perspective, the strong typing of interpretative nodes and arcs means that the task of finding similarities between encodings is one of finding **isomorphisms** between their topologies. Without analyzing the propositions themselves (that is, without losing domain independence), we can say that if each of two encodings has a pair of nodes with an "A attempts to cause B" relationship, the stories have a similarity—they are both about an agent intentionally striving to cause something to happen. The SIG closure rules assist such a detection of isomorphism by adding transitive relations (arcs) to each graph. With closure arcs in place, a four-step plan in one encoding can be mapped to a two-step plan in another encoding, because both encodings have within them two-step plans (Figure 5.4).

The remainder of this chapter describes two approaches toward finding isomorphisms in the interpretative sub-graphs of multiple SIG encodings. They can be thought of as one "top-down" approach and one "bottom-up" approach. We first describe the top-down approach, which relies on an *a priori* series of meaningful graph fragments which can be mapped onto particular stories in a manner similar to Story A's mapping onto Story B in Figure 5.4. The second approach compares two graphs directly against one another to dynamically determine the largest contiguous isomorphic subgraph that exists between them. We follow both approaches with an evaluation that also includes the propositional overlap routine we have discussed.

5.3.1 Static Pattern Matching

Appendix B defines the *SIG pattern* as a fragment of a hypothetical SIG encoding which minimally describes a certain narrative scenario. We describe 80 such patterns as demonstrating the expressive range of the SIG formalism. To summarize some of these in brief, the patterns fall into 12 groups:

1. Basic affectual status transitions (Figure B.2). A Gain occurs when an interpretative node is actualized that has a positive affectual status transition; a Loss carries a negative transition; a Promise is the actualization of a hypothetical that *would cause* an event.

2. Complex affectual status transitions (Figure B.3). A Partial Resolution involves multiple Losses followed by a single Gain that reverses only one loss; a Compounded Transition occurs when an impactful event or stative is actualized, ceased, and actualized again in alternation.

3. Simple single-agent goals and plans (Figure B.4). A Problem is a Loss followed by a desire to reverse the loss; Change of Mind involves the actualization and subsequent cessation of a goal frame; Perseverance involves one or more attempts to fulfill a goal.

4. Simple single-agent goal outcomes (Figure B.5). Success and Failure occur when an hypothetical node inside an actualized goal frame (that is, an agent's active goal) is itself either actualized or prevented/ceased, respectively. Deliberate actions are indicated through the *attempt to cause* and *attempt to prevent* arcs, side effects through additional *actualize* arcs originating from the same actions. A Backfire is a harmful side effect that defeats the intended purpose of the action.

5. Complex single-agent goal outcomes (Figure B.6). Goal substitution occurs when an agent compensates for failure by devising an alternate plan to achieve the same end; Giving Up involves both Failure and Change of Mind.

6. Single-agent beliefs and expectations (Figure B.7). Mistaken beliefs are those that are contradicted by opposing nodes in ground truth (outside of any agency frame). Surprise occurs when a character believes an event is unlikely, due to the actualization of a factor that would seem to prevent it, but the event indeed occurs.

7. Dilemmas (Figure B.8). A dilemma can arise from an agent's belief that two of its goals are mutually exclusive, or that a single event would bring some aid but also some harm. Goal Prioritization involves an agent "deciding upon" a dilemma by deliberately pursuing a goal whose success would also harm the agent in some way.

8. Two-agent interactions (Figure B.9). Selfish acts involve actions intentionally meant to hurt one's self and harm another; selfless acts are the opposite. Conflicts occur when a single hypothetical is the subject of an *attempt to cause* by one agent and an *attempt to prevent* by another.

9. Persuasion and deception (Figure B.10). Persuasion involves a belief frame of one agent being itself inside the goal frame of another (that is, one agent's intentional stance being the subject of another agent's goal). Deception is similar, but the persuader knows the fact in question to be untrue.

10. Complex two-agent interactions (Figure B.11). We may recognize that an agent is motivated to take revenge upon another if it had been previously harmed by that other, and now wishes to harm it out of a sense of justice. A hidden agenda involves

the persuasion of another agent to actualize a certain plan without understanding its benefit to the persuader.

11. The manipulation of time (Figure B.13). Disfluencies in the orderings of Text nodes and Proposition nodes can indicate either flashbacks or flash-forwards. Suspense is invoked when the resolution of a narrative question, such as goal outcome, is delayed in the story's telling.

12. Mystery (Figure B.14). An event has an ambiguous causal antecedent, or an agent pursues a plan without first explicating what its intentions are. (The encoding of the Forster citation in Figure 3.19, with which we ended Chapter 3, would match this SIG pattern as the first event precedes its causal antecedent.)

These and other patterns are permutations of the nodes and arcs we defined in Section 3.3.2. Their enumeration is an effort to identify thematically rich, prototypical narrative scenarios. In Appendix C, we give the formal pattern definitions that we use for the present experiment. These rules work alongside the closure rules to find where SIG encodings instantiate each of the patterns. In short, by writing an extension to Scheherazade that exports a first-order logic version of a SIG encoding, with nodes as atoms and arcs as relations, we may draw a feature vector out of each encoding. Each of the 80 features indicates whether a certain pattern occurs in the encoding. By comparing these features to one another, we can determine the degree of similarity that occurs between stories, and detect corpus-wide trends as well.

Agreement

We extracted feature vectors for the 80 SIG patterns from each of the 70 encodings in Collections B and C. (Recall that Collection A lacks the interpretative-layer annotations that are necessary for this approach.) We first applied these vectors to the question of inter-annotator agreement.

As its name implies, interpretative-layer annotation is designed to allow an individual receiver to encode his or her reading of a narrative. Our goal for inter-annotator agreement cannot be absolute, as it may be for more objective annotation tasks. One reason for this,

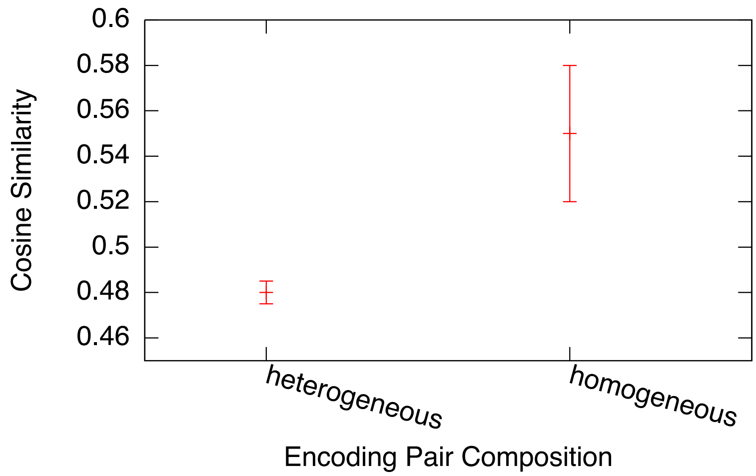


Figure 5.5: Interpretative-layer inter-annotator agreement is shown through cosine similarity between feature vectors. Error bars indicate 95% confidence.

as we mentioned in earlier Section 5.2, is that annotators may have appreciable differences of opinion that we would wish to study rather than marginalize. Our present study is in search of an algorithm for finding similarities between stories, but there is no reason we cannot pivot to study the differences between individuals should we so desire. The second reason is that complete agreement is impractical in a task that is, by its nature, subjective; one cannot expect perfect agreement in a task that involves "mind-reading" fictional agents. Reasonable individuals may come to different conclusions about the motivations of narrative agents, for lack of explicit evidence for one take over another; our annotators made this point in their survey responses. The contribution of the SIG approach to modeling narrative is that it allows us to concretize each alternative interpretation, so that we might learn from each in turn, and from all in concert.

In lieu of perfect agreement, we ask again as an initial check: Are homogeneous story pairs, from different annotators on the same source story, significantly more similar to one another than heterogeneous story pairs? The answer is yes. Figure 5.5 shows the cosine similarity between the extracted vectors, split between homogeneous pairs of encodings (same story, different annotators) and heterogeneous pairs (different stories). To normalize for story length, we limit each feature to 1 (the encoding instantiates the pattern at least once) or 0 (the pattern never matches). By the two-tailed Student's t-test, homogeneous pairs are more similar to p<.001. This extremely low p-value demonstrates that our method of measuring similarity is quite sensitive to parallel encodings, and from this we infer that it is an acceptable measure of similarity in general. If either the metric had an unacceptable accuracy, or homogeneous encodings did not have measurable inter-annotator agreement for us to detect, we would not see such significant results.

If we apply Cohen's kappa statistic to the same data, considering each of the 80 features as a potential agreement or disagreement and taking the overall distribution of feature values as the basis for chance agreement, the result is k=.55. This indicates moderate agreement. This measurement considers all annotators and all stories involved in Collections B and C. The result is virtually the same (k=.56) if we only consider the 18 paired encodings of the users A106 and A108. Collection C does not have enough parallel encodings for us to break down kappa by genre or text length; determining the effect of story "density" (as we discussed above) on agreement is left for future work.

Corpus Trends

Having established the soundness of static SIG pattern matching as a metric for measuring the similarities between any two encodings, we can briefly examine corpus-wide trends in Collections B and C. To begin, we aggregate our collection to reconcile parallel encodings. If $\vec{P_e}$ is the vector of 80 pattern features $P_{e,f} = \{0, 1\}$ for some encoding $e{\in}E_s$, were E_s is the set of encodings of a particular story, then each feature of the reconciled vector $\vec{P_s}$ is the arithmetic mean of the contributing encodings:

$$\vec{P_s} = \frac{\sum_{e \in E_s} \vec{P_e}}{|E_s|}$$

This has the effect of causing each feature to be a quotient of agreement that is normalized with respect to the number of parallel encodings for the story. For instance, if the Change of Mind pattern matches one out of two parallel encodings for a story, the canonical feature value for that story will be $\frac{1}{2}$.

The most commonly occurring patterns in Collections B and C (among 34 canonical vectors) are shown in Table 5.5. Not surprisingly, they consist of the smaller building blocks of plan and goal machinations featured in Figure B.2 (basic affect status transitions) and

Pattern	Coverage	Figure	Description
Goal Declared	34	B.4	An agent actualizes a goal frame (declares a goal).
Gain	34	B.2	An agent receives a positive affectual impact.
Goal failure expected	33	B.4	Something occurs which would cause an agent's goal to become prevented/ceased.
Loss	33	B.2	An agent receives a negative affectual impact.
Goal Enablement	33	B.4	Something occurs which satisfies a precondition for an agent's goal.
Mixed Blessing	33	B.2	Something occurs which has both a negative and a positive affectual impact on the same agent.
Promise	32	B.2	Something occurs which anticipates another event.
Perseverance	32	B.4	An agent makes an attempt to actualize its stated goal.
Success	31	B.2	An agent succeeds at actualizing its stated goal.
Promise Fulfilled	31	B.2	A promise is followed by the actualization of the anticipated event.

Table 5.5: The ten most highly covered static SIG patterns in DramaBank among the 34 modeled stories.

B.4 (simple single-agent goals and plans). Every story actualizes at least one goal, and has at some point a beneficial impact on an agent. The data are too sparse for a statistical comparison between the Aesop and non-Aesop segments of the corpus, but anecdotally we observe that certain SIG patterns occur more frequently in the non-Aesop texts, especially those dealing with complex and sometimes competing goals (Goal Prioritization [B.8], Goal Substitution [B.6], Goal Avoidance [B.4], and Partial Resolution [B.3]). This is not surprising, given that the Aesop fables are too short for the narrator to have much time to describe an agent planning ahead or recovering from loss.

Affect typing (Section 3.3.2.8) can also be seen in aggregate to characterize a corpus. A mixed blessing (B.2), for instance, is when the same event hurts an agent in one way, but helps it in another. In aggregate, the most common type of mixed blessing in DramaBank is one which involves "ego" and "health"—the lion's fateful decision to pursue an elusive stag instead of the quotidian hare, for instance, was intended to be beneficial for his ego but was ultimately bad for his health, because he could not catch the stag and found the hare missing when he tried to return to it. "The fable's disapproving of such behavior as a result of the vice of greed forms part of the general message that one should compromise," remarks Zafiropoulos [2001, 194] in regard to this fable.

Determining the similarity between two heterogeneous encodings using this approach involves the same cosine distance metric we used above to measure inter-annotator agreement. We evaluate this method for finding story similarity in the next section.

We do not claim that these 80 features document all of the narrative concepts that the SIG can express; indeed, in the Aesop corpus there are others that would be helpful for the similarity task. One might, for instance, define a pattern that detects jealousy as a goal to `have` something, with a causal antecedent that is another agent's possession of the same item or ability. Such domain-specific feature engineering would be suitable for a directed exploration of a corpus, i.e., testing hypotheses about a specific story or about corpus trends. The closure rules then act as a "search engine" to find instances of a "query" pattern. For the unsupervised detection of story analogies, we propose a separate algorithm below.

5.3.2 Dynamic Analogy Detection

Our third and final method for finding similarities between SIG encodings takes a "bottom up" approach, as opposed to the "top down" approach characterized by static SIG pattern matching. In other words, rather than map each encoding against an *a priori* set of compounded relations, we compare encodings to one another directly in order to organically find the largest and most complex areas of overlap that can be found. The other key difference compared to the pattern-matching approach is that we wish to maintain a **consistent binding** when determining similarity. Consistency takes several forms, notably with regard to time (so that the direction of time is consistent) and agency (so that each agent maps exclusively to one agent in the opposite story). It is one matter to claim that two stories are similar because they both involve a broken promise and a motivation to avenge; it is another matter to claim that both stories are about the *sequence* of a broken promise and a motivation to avenge. Not only must the ordering be the same across both contributing encodings, but the individual who is both betrayed and motivated to avenge in one story must map exclusively to an individual who fulfills the same roles in the opposite story. In a dynamic analogy, we would not say that the lion is like the fox for the first half of the story, and like the crow in the second half of the story.

Analogy has been studied in artificial intelligence for decades (e.g., see [French, 2002] for a review), but not traditionally in the narrative sense. Typically, an analogy detection task is one that proposes an analogical relationship between two objects and queries the system to extend the analogy along one or more relations ("if a bike is like a car, what is the bike's steering wheel?"). However, there is no fundamental difference between functional and part-whole relations such as these, and goal- and plan-based relations such as ours. In both cases, the basic "connectionist" algorithm is the same: Consider two elements to be analogically linked, and conduct a search through the nodes that connect to them, determining which parallel neighbors can also be linked in such a way that does not violate the overall consistency of the analogy.

The ACME model [Holyoak and Thagard, 1989] is a well-known example of connectionist analogy detection. It uses such a "probing search" that considers three factors when extending an analogical mapping:

1. Structural, which ensures that the predicates being mapped are the same, the arguments are parallel with respect to type, and mapping the arguments to one another does not violate the consistency of the global binding;

2. Semantic similarity, which ensures that the correspondences being considered as analogous have a similar meaning, and

3. Pragmatic, which favors those correspondences which the "analogist" (i.e., the user) feels are important.

Holyoak and Thagard applied their system to the narrative case, but used only synthetic stories with fewer interconnections than a SIG encoding. At present, we model our narrative analogy algorithm after this connectionist approach, adding a few extra conditions and optimizations which are specific to the SIG model. Differing approaches to narrative analogy have also recently been attempted, though. In one contemporary attempt to replicate Propp's process for finding story functions, Finlayson [2009] models each story in a corpus as a sequence of states, and then uses a technique called Bayesian Model Merging for inducing a grammar from the corpus. The operation finds Hidden Markov Models (HMMs) that describe event-transition sequences for each story, and then merges HMMs from different stories to find models that maximize the posterior probability of generating the most stories. This model, though, is limited to events in the textbase. In its present formulation, it does not include implied goals, plans, or aspects of agency which we have established as important to our model of narrative discourse.

Algorithm

Our dynamic analogy algorithm works by seeding a set of "globs" in the interpretative layer. A glob is consists of a binding (a list of parallel nodes that we have declared to be analogically bound, as well as agents committed to one another) and two sets of relations (arcs), one from each encoding, that connect the bound nodes.

The first step of the algorithm is to apply the closure rules we have described to each encoding separately, so that approximate matches can be found when strict isomorphism is not present. Figure 5.4 illustrates an example of this by using transitive *would cause* arcs to match plans of different lengths as roughly analogical. The second step is to determine a list of initial "glob seeds" to begin the search. Simply put, in two encodings with M and N propositions in their linear timelines, each MxN pair of nodes across both encodings is considered as the beginning of a potential analogy. For each pair, any adjacent and isomorphic pair of interpretative-layer nodes is joined with the timeline-node pair as a glob seed. For example, at the top of Figure 5.6, two Proposition nodes on the timelines of two stories are considered for an analogical connection (as one shaded pair). Originally, the P node in the left encoding connects to one interpretative node with an *attempt to cause* arc, while the counterpart P node in the right encoding connects to two nodes with the same type of arc. When closure rules are applied, two additional *attempt to cause* connections are added (one on each side, drawn with dotted curves). These Proposition nodes therefore seed six globs, as there are six permutations of arcs/nodes that we can trace from this one pair of P nodes and the six possible pairs of adjacent I nodes. Two of these six globs are drawn in the figure, with red arcs between nodes denoting binding equivalences.

After the globs are seeded, the process repeats. Each glob considers expanding one analogical binding further among the possible node pairs that are adjacent, unbound and isomorphic with respect to node and arc type. The structural restrictions are:

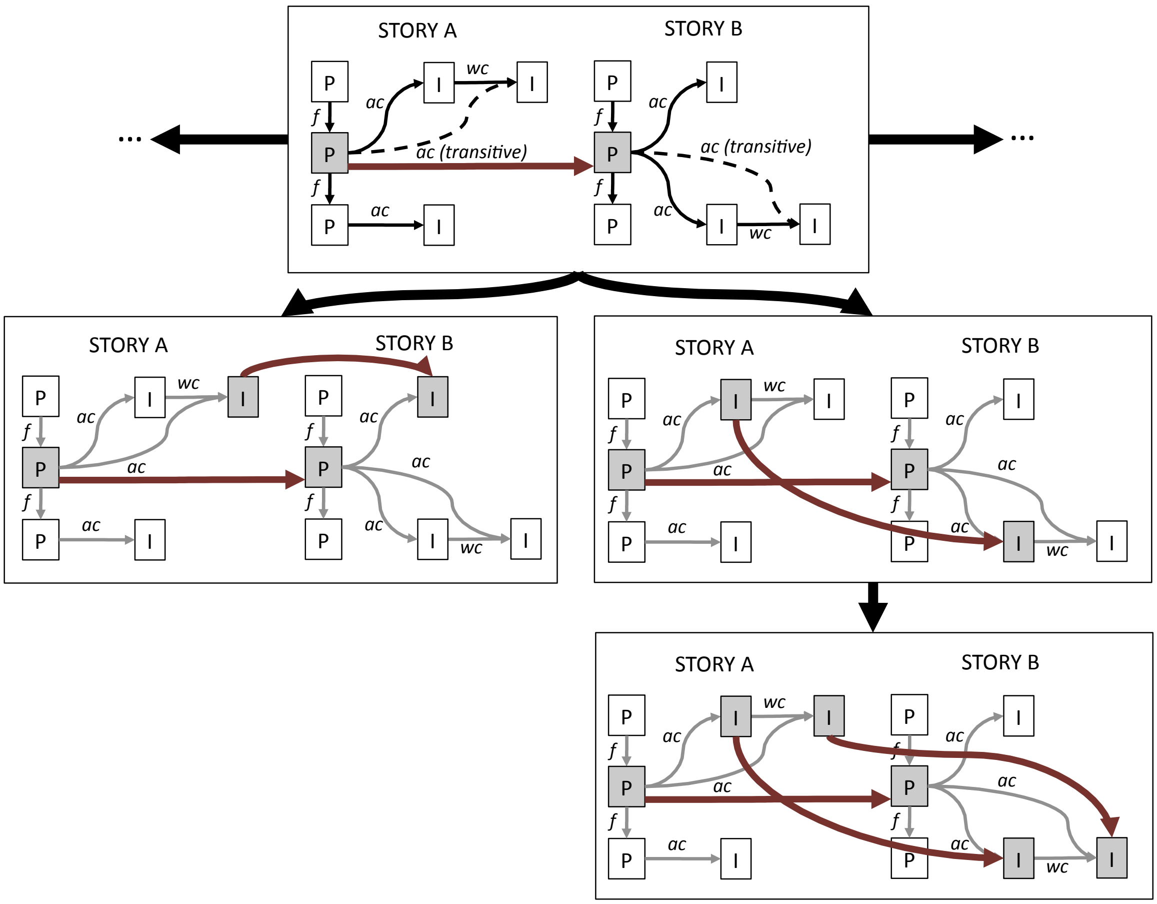


Figure 5.6: The dynamic analogy search routine traverses multiple SIG encodings in search of isomorphisms.

1. The globs can only expand along forward arcs. This is an optimization to prune out duplicate globs.

2. Globs can only expand to two nodes that are in analogous agency contexts. Goals must map to goals; beliefs must map to beliefs; goals inside beliefs must map to goals inside beliefs; and so on. For instance, to expand to a node I that is in a belief frame of agent X, we must find a node J in the opposite encoding that is in the belief frame of agent A, where *A is known to be the agentive binding for X*. If X does not have an agentive binding, any opposite agent will suffice, as long as that opposite agent is also unbound; in this case, the two agents become bound in the scope of this glob.

A node I within a belief of X within a goal of Y can match only a node J within a belief of A within a goal of B, provided that X and A are either already bound to one another or both unbound, and Y and B are either already bound to one another or both unbound. (When a glob is seeded, an initial agentive binding is seeded as well, joining together the agents of the two Proposition nodes.)

3. Globs cannot expand in such a way that has been previously considered by the search routine. A would-be duplicate glob is avoided. This optimization, known as memoization, is similar to the technique behind dynamic programming—we know that the optimal expansion for any glob with a certain binding does not depend on the "path" the search routine traced to arrive at that glob, so only one glob needs to be considered for any unique binding.

4. If a glob can expand in more than one way (i.e., if there is more than one possible pair of bindable neighbors), the routine forks a duplicate glob to accommodate each additional possible expansion. The base case, when the algorithm finishes processing a glob, occurs when there are no bindable neighbors to which the glob can expand.

The result is a set of interpretative-layer analogical bindings that expand as "trees" from the timeline P nodes (as roots) to a set of Affect node drains (as leaves), since we only allow expansion along forward arcs and no cycles are allowed in this layer. The algorithm then takes two additional steps to arrive at an overall set of analogical bindings between the two encodings:

1. Each glob determines which pairs of timeline Proposition nodes, out of the MxN intersection of the two timeline vectors, would be consistent with the constraints of the glob's binding. Notably, the agents involved in each potential Proposition join must be consistent with the agentive bindings drawn from the interpretative-layer agency frames. For instance, if a belief of X has been bound to a belief of A, an action in which X is the agent cannot be bound to an action in which B is the agent.

   Given a set of legal bindings, each glob then runs the Needleman-Wunsch alignment algorithm [Needleman and Wunsch, 1970] to maximize the number of Proposition nodes that can be aligned (that is, added to the glob and its binding). This dynamic programming algorithm rapidly maximizes the number of node pairs that can be added to the binding without violating temporal consistency (if Proposition nodes A:B::X:Y, and A precedes B, X must precede Y).

2. Each glob, from the largest to the smallest, attempts to absorb smaller globs. When one glob finds a smaller neighbor such that the two bindings can be merged without conflict, it carries through with this merge. Note that this is a greedy algorithm that assumes that the largest available glob is the best one to absorb, even though this may preclude the absorption of multiple smaller globs that add up to a larger whole (a version of the knapsack problem in combinatorics). We have found the greedy technique to successfully find the largest possible analogies without cornering them into local maxima.

The result is a set of mutually incompatible globs that represent the possible analogical bindings between the two encodings. The globs are culled from propositional nodes (who does what) and agency frames (who believes what and who desires what). By counting the relations, nodes and agents found to be analogous in the binding, the glob can be given a score. The top-scoring (largest) glob becomes the top-line result—a dynamically generated isomorphic subgraph joining together two SIG encodings.

Results

We have found this algorithm to return substantive analogies, as measured by the sizes of the isomorphic subgraphs that are found: 8.8 bound node pairs, 1.5 agentive bindings and 14.1 analogous relations on average (including inferred, transitive relations) across 1,015 heterogeneous encoding pairs in Collections B and C.[5] The runtime is exponential with respect to the lengths of plans, due to the internal transitive arcs; however, through aggressive optimization in the form of pruning, and by postponing the integration of the timeline into each glob until a sequential process (rather than a compounding one), we are

[5]We did not include encodings of "The Milkmaid and Her Pail" in this data set, which is used in the forthcoming evaluation, due to implementation details.

able to find the most complete analogies between two fables in Collection B in one to three seconds on a modern laptop machine. Collection C analogies converge in a few seconds to several minutes (*Beowulf* being responsible for the higher end). Further optimizations, especially regarding the handling of transitive arcs, are an area of focus for future work.

The largest analogies found in the corpus, by the number of bound node pairs, were between two particular encodings of "The Wily Lion" and "The Fox and the Crow". This is an initial check on our approach, as while we did not develop the dynamic analogy algorithm using this pair of encodings, we did select these two fables for inclusion in Collections A and B in part due to their strong analogical connection (as we discussed in Chapter 3). By abstracting each bound pair of nodes into a single compound node, we visualize this procedurally detected analogy as a single hybrid encoding in Figure 5.7. In this case, there are 11 aligned timeline propositions, two goal frames (one nested within the other as part of a four-stage plan), and two Affect nodes that indicate the affective "orientation" of the entire analogy. The latter point is key: While the static pattern approach we described above considers each overlapping pattern in isolation, the dynamic analogies generated here are internally consistent with respect to agent bindings. In the present example, the overall result is that "the fox is like the lion" and "the crow is like the bull," in that in both stories, one is an inciting agent who devises a plan that has the other agent devising and executing its own plan. This particular analogy is further enriched by the aligned use of the *flatter* predicate in the plan, although in general the dynamic analogy search routine considers graph topology alone. We experimented with combining this algorithm with the previously described approach to propositional similarity, with the latter helping to guide the former through the search space, but found that there were few cases (even in the highly declarative Aesop domain) where any heterogeneous propositional alignments were possible between globs. In a moment, we will evaluate the performance of both approaches separately for the story similarity task.

As a further check on our confidence in this approach, we can carry out the same diagnostic as we did on the other two similarity routines—that is, determine whether homogeneous encoding pairs, or parallel encodings by different annotators of the same story, produce quantitatively larger analogies than heterogeneous pairs. The answer, seen in Fig-

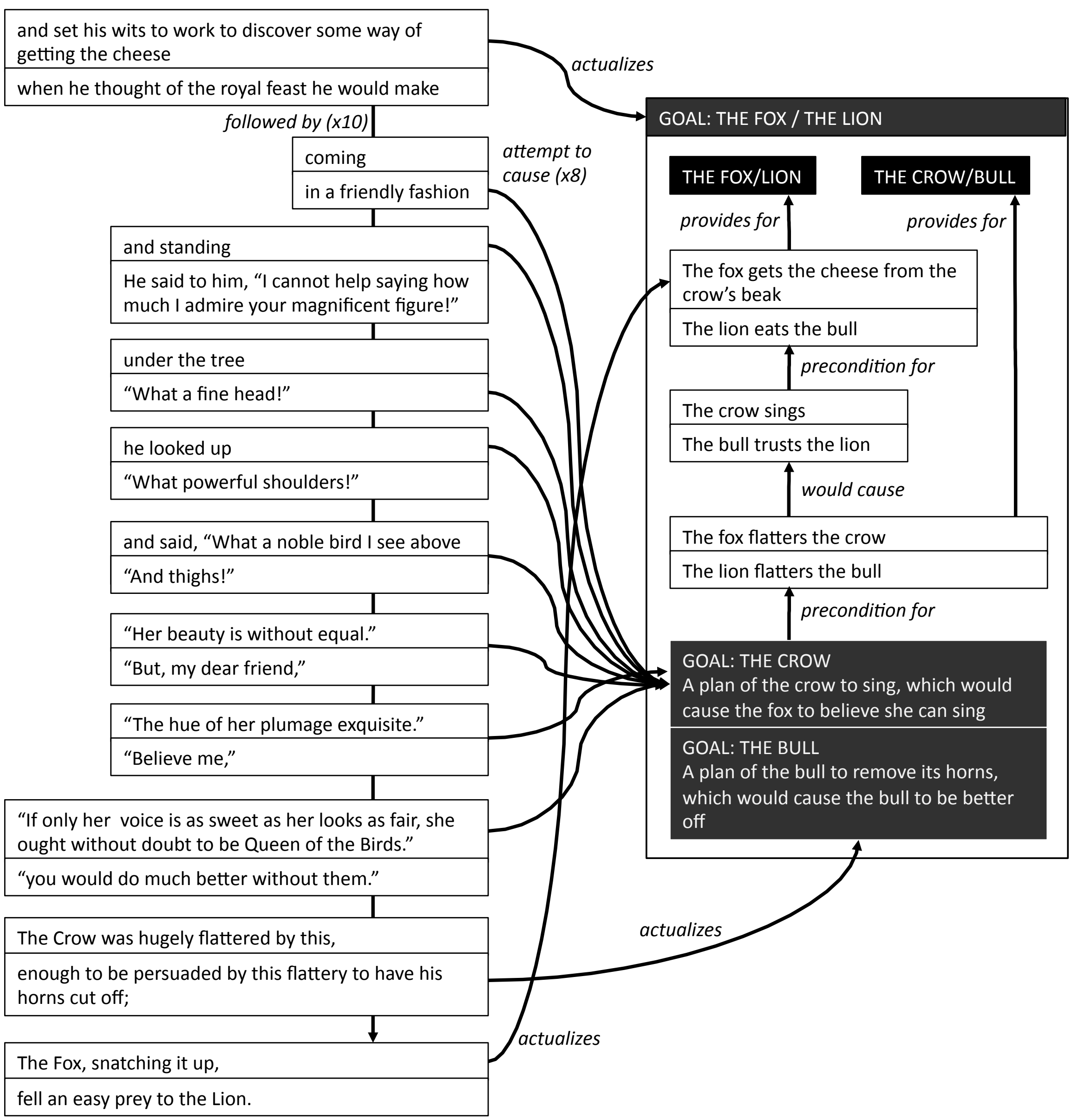


Figure 5.7: Analogy procedurally drawn between SIG encodings of "The Wily Lion" and "The Fox and the Crow".

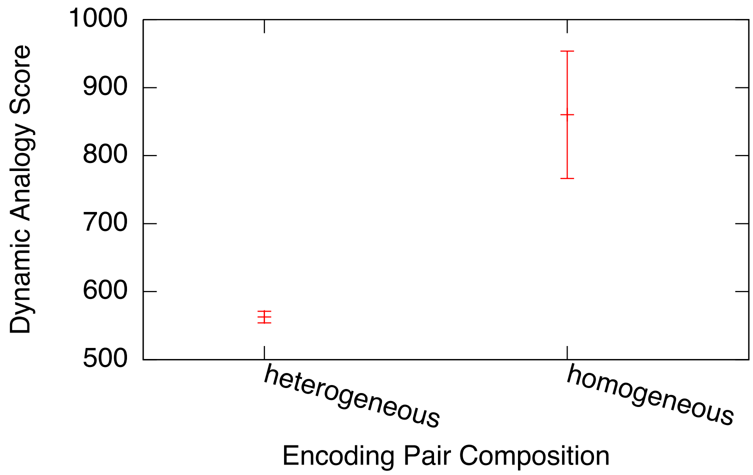


Figure 5.8: Interpretative-layer inter-annotator agreement is shown through the scores derived from the largest dynamically generated analogies found between heterogeneous and homogeneous pairs of encodings. Error bars indicate 95% confidence.

Story 1	Story 2	Node Pairs	Agentive Analogy
"The Fox and the Crow"	"The Wily Lion"	15	Fox:Crow::Lion:Bull
"The Donkey and the Mule"	*Beowulf*	19	Donkey:Muleteer::King:Thane
"The Dog and the Wolf"	"The Wily Lion"	14	Dog:Wolf::Lion::Bull
"The Serpent and the Eagle"	*Beowulf*	17	Countryman:Eagle:Serpent:: King:Thane:[Grendel's] Mother
"The Dog and the Wolf"	"The Lion In Love"	12	Dog:Wolf::Cottager:Lion

Table 5.6: Story pairs whose encodings yielded the top-scoring dynamic analogies.

ure 5.8, is that homogeneous pairs yield significantly larger analogies than heterogeneous pairs (p<.001), more than 50% larger on average. The unit used in the Y axis, Dynamic Analogy Score, is a linear combination of the counts of bound node and relation pairs (with node bindings weighed more heavily than transitive relation bindings, which can be redundant). The highest-scoring analogy found between two encodings is used to represent the encoding pair as a whole. We conclude that annotators agree more on interpretative-layer graph structure when working from the same story than from opposing stories.

The story pairs whose encodings yielded the highest-scoring analogies are listed in Table 5.6. We see that certain stories are repeated in the list; as analogical similarity is transitive, it follows that certain "cliques" will appear in the similarity matrix. The analogy found between "The Lion In Love" and "The Dog and the Wolf" is shown in Figure A.4.

5.4 Evaluation

In this chapter we have described three separate methods for measuring similarities and detecting analogies among SIG encodings:

1. **Propositional overlap.** Using the measurement of similarity between propositions, we can find similar events among multiple encodings.

2. **Static SIG pattern matching.** Once we have extracted a feature vector from each SIG encoding, where a feature corresponds to the presence or absence of a particular graph fragment from a curated set meant to represent narrative tropes, we can measure similarity by taking the cosine distances between vectors.

3. **Dynamic analogy detection.** By running a search algorithm that finds the largest isomorphic subgraph between two SIG encodings that maintains a consistent binding, and measuring the size of that subgraph, we can determine the degree of analogy that exists between the two encodings.

To determine which of these corresponds to an applied notion of story similarity, if any, we ran an evaluation with users of Mechanical Turk. The evaluation has two parts. For the first part, we showed users story pairs as well as analogies generated by one of our three methods, and asked the annotators to grade the analogy on a pair of Likert scales relating to accuracy. For the second part, we elicited "gold standard" ratings of story similarity, and used them to train regression models based on each similarity method.

Direct Analogy Ratings

For the first part of the evaluation, users saw an interface that included both stories and the results of one of the three methods. Users only saw the output of one of the three algorithms at a time. To increase the breadth of the data collection, we only tested one pair of encodings for each story pair (taking the largest similarity found under each approach). In order to make the evaluation easier, we wrote heuristics to convert all three symbolic analogy structures to natural language feedback text:

- For propositional overlap, we modified our feedback text generation module (described in Section 4.4) to accept not only story propositions as input, but "synthetic" propositions as well—those whose predicates and arguments are the nearest common ancestors of their contributing story propositions. Where there was no argument overlap, predicates were given with placeholder arguments or no arguments. For instance, "the cat climbed" and "the lion pounced" would yield the intersection "a feline moved."

- For static pattern matching, we wrote a set of 80 short descriptions, one for each pattern, and invoked the descriptions that were associated with the patterns matched in both of the stories being compared. For instance, if the encodings of two stories both matched the Gain pattern, the analogy presented to the raters included the sentence "Both stories have characters who have good things happen to them." We did not attempt to indicate which aspects of the stories were relevant to each pattern.

- For dynamic analogies, we developed a method that attempts to best communicate a connected graph structure as a series of short feedback sentences. Because dynamic analogies are agent-centric, each serialization began with a summary of agent bindings in the form of "X is like A, and Y is like Z." The method then visits each aligned pair of timeline nodes and generates a short sentence that describes the SIG relation keyed to that bound pair. For instance, a bound pair of timeline propositions joined to the remainder of the analogy with an "attempt to cause" arc would yield a sentence such as "Both X and A strive to reach their goals." In certain cases, this is followed by a reading of parallel goal propositions, as generated by the Scheherazade feedback text generator.

An example of the output of this process is shown in Table 5.7: Raters saw both source stories (top), then one of three suggested analogies (bottom). We asked raters to grade the quality of the similarity outline in two ways. First, we asked a "precision" style accuracy question: "Are the above similarities accurate? In other words, how correctly do they describe common ground between the two stories?" We then asked a "recall" style completeness question: "How many of the similarities that exist between these stories are included in the outline?" Each question was followed by an option to choose from among

Story 1	Story 2
There was once a house that was overrun with Mice. A Cat heard of this, and said to herself, "That's the place for me," and off she went and took up her quarters in the house, and caught the Mice one by one and ate them. At last the Mice could stand it no longer, and they determined to take to their holes and stay there. "That's awkward," said the Cat to herself: "the only thing to do is to coax them out by a trick." So she considered a while, and then climbed up the wall and let herself hang down by her hind legs from a peg, and pretended to be dead. By and by a Mouse peeped out and saw the Cat hanging there. "Aha!" it cried, "you're very clever, madam, no doubt: but you may turn yourself into a bag of meal hanging there, if you like, yet you won't catch us coming anywhere near you."	A Dog was lying in the sun before a farmyard gate when a Wolf pounced upon him and was just going to eat him up; but he begged for his life and said, "You see how thin I am and what a wretched meal I should make you now: but if you will only wait a few days my master is going to give a feast. All the rich scraps and pickings will fall to me and I shall get nice and fat: then will be the time for you to eat me." The Wolf thought this was a very good plan and went away. Some time afterwards he came to the farmyard again, and found the Dog lying out of reach on the stable roof. "Come down," he called, "and be eaten: you remember our agreement?" But the Dog said coolly, "My friend, if ever you catch me lying down by the gate there again, don't you wait for any feast."

In both stories, a carnivore perceives. In both stories, an animal says to a carnivore that an animal has some property. In both stories, an animal states that an animal has some property. In both stories, an animal moves. In both stories, a carnivore travels. In both stories, a carnivore communicates. In both stories, an animal states something. In both stories, an animal says to a carnivore that a thing has some property. In both stories, a carnivore moves. In both stories, a carnivore travels to a structure. In both stories, a placental mammal sees something doing some action. In both stories, an animal interacts.	Both stories have characters who have good things happen to them. Both stories have characters who have bad things happen to them. Both stories have a mixed blessing—something happens that is good in one way and bad in another way. Both stories make clear that their characters have certain goals. Both stories have characters who have goals to help themselves or help others. Both stories have characters who have goals to harm themselves or harm others. Both stories have characters whose goals come within reach when certain actions make them possible to achieve. Both stories have characters whose goals become harder to reach when certain obstacles make them more difficult to achieve. Both stories have characters who come very close to reaching their goals when certain actions put them within reach. Both stories have characters who come very close to failing to reach their goals when certain actions seem to defeat them. Both stories have characters who persevere toward reaching their goals. Both stories have characters who succeed in reaching goals. Both stories have characters who fail to reach their goals. Both stories have characters who deliberately help others. Both stories have a tragic turn of events, with characters who seem to be within reach of their goals before encountering setbacks. Both stories have characters who expect something to happen, but the expected event never happens. Both stories have characters who act selfishly, putting their own needs in front of the needs of others.	Analogically, the mouse is like the dog, and the cat is like the wolf: Both the cat and the wolf have goals. The goal of the cat is "the cat eats the group of mice;" the goal of the wolf is "the wolf eats the dog." Both the cat and the wolf strive to reach their goals. Both the mouse and the dog have goals. The goal of the mouse is "the cat stops eating the group of mice;" the goal of the dog is "the dog is fattened." Both the mouse and the dog strategize toward reaching their goals. The goal of the mouse is "the cat stops eating the group of mice;" the goal of the dog is "the dog is fattened." Both the mouse and the dog strive to reach their goals. Something bad happens to both the cat and the wolf. Both the mouse and the dog have important beliefs. The belief of the mouse is "the cat isn't dead;" the belief of the dog is "the wolf sees the dog being near the gate of the farmyard."

Table 5.7: Three example prompts from our evaluation of story similarity metrics: Raters saw two source stories (top) and one of three sets of proposed similarities.

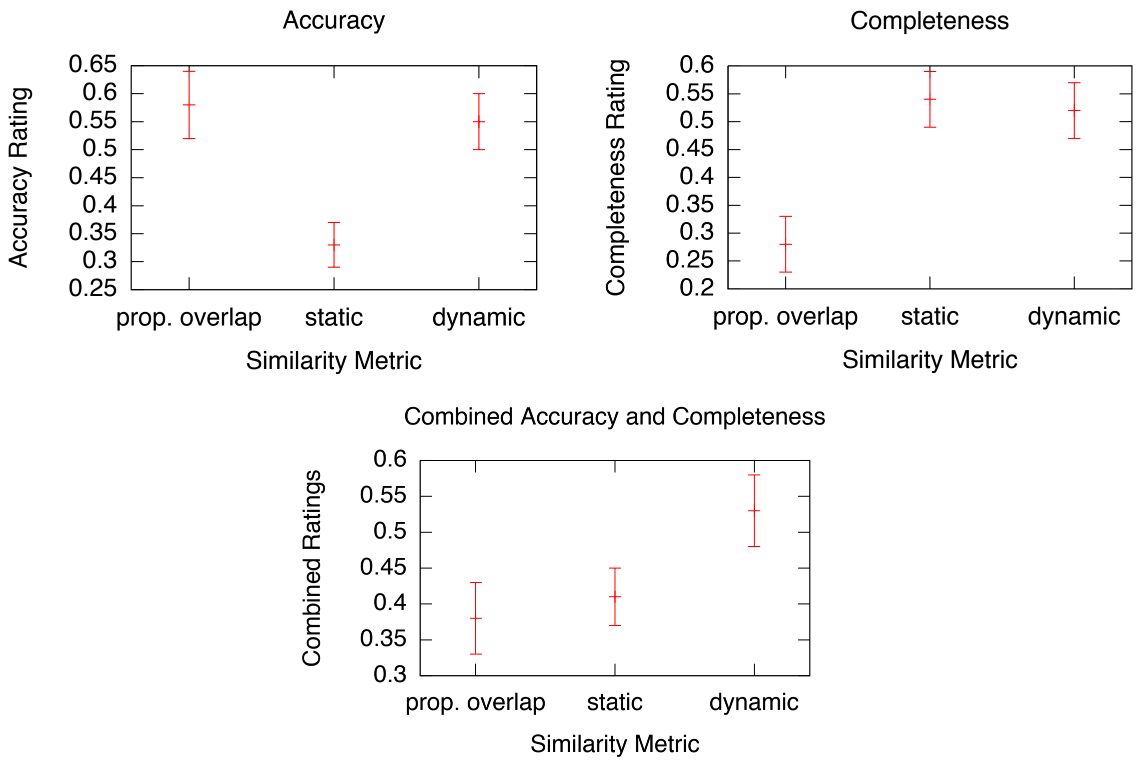


Figure 5.9: Story-similarity accuracy and completeness as judged by Mechanical Turk users. Error bars indicate 95% confidence.

three possible ratings on a Likert scale.

We collected three data points (six ratings) for each of three similarity algorithms over approximately 100 story pairs. The rate of unanimous consensus for each rating was 27%; a two-way majority occurred in another 61% of cases. Although this is only fair agreement, we were able to take the average of the numerical equivalents of the responses (rather than take a majority vote).

The results are shown in Figure 5.9. On the accuracy question, the propositional overlap method showed the best performance on average, although its improvement over the dynamic analogy method is not significant. The static pattern method performed significantly worse on this question. We believe a portion of this is due to the generic nature of the feedback text, which offered the same phrasing every time the pattern was invoked. (Investigating a variation on this approach that presents specific details regarding the manner in which each story covered each SIG pattern is a direction for future work.) On the completeness question, the results are reversed, with static patterns significantly outperforming propositional overlap. We infer that SIG patterns are highly sensitive to thematic story overlap, but non-specific (this is underlined by their disconnected nature, compared to the more cohesive dynamic analogies). Propositional overlap, as we previously discovered, is highly specific but has low coverage, despite our earlier efforts toward normalizing propositional paraphrases.

Dynamic analogies perform well on both metrics and appear to strike a balance between specificity and sensitivity. If we continue the analogy to precision and recall, we can take an "F-measure-style" combined measurement by computing the harmonic mean of the accuracy and completeness scores. The result, shown at the bottom of Figure 5.9, shows that dynamic analogies significantly outperform both static patterns and propositional overlap for the task of suggesting accurate and complete similarities that occur between two short fables. We believe that the overall results show a preference among raters for expressions of story similarity that are goal-oriented and bottom-up, as opposed to originating from a list of tropes or themes. However, further work is needed to corroborate these results, such as a baseline condition that is propositional but hand-authored for each story pair (that is, action-oriented and high coverage). Such a baseline would eliminate the sources of error we discussed in our previous dedicated evaluation of the propositional similarity algorithm (Section 5.2.2).

Evaluation Against Gold Standard

We examined the same issue from another angle in the second part of our evaluation. Rather than ask raters to consider a suggestion of a story similarity, we instead asked them to directly rate the degree of similarity between two fables. We then trained a linear regression model with our three metrics acting as predictor variables against their preferences. This approach has the advantage of eliminating any potential interference from the feedback text generation module, unlike the previous part of the evaluation, as raters here are never exposed to any of our analogy results.

The Mechanical Turk-powered interface presented raters with two fables from the Aesop corpus along with two prompts: First, rate the degree of similarity on a three-point Likert

156	character	30	animal	17	man	14	killed	11	predator
105	have	28	dog	16	lost	14	unhappy	11	own
83	something	28	more	16	hungry	14	eagle	11	lose
78	get	24	crow	16	thing	13	poor	11	consequence
73	involve	23	farmer	15	punish	13	pride	11	prey
63	end	21	tortoise	15	action	13	theme	10	goal
60	wolf	20	food	15	someone	12	result	10	due
56	lion	20	make	14	there	12	thought	10	judgment
54	similar	19	only	14	cat	12	meal	10	wife
51	story	19	sheep	14	could	12	deception	10	rooster
49	want	18	eat	14	shepherd	12	creature	10	mouse
42	try	18	plan	14	bad	12	use	10	however
41	greed	18	bull	14	monkey	12	mule	9	death
39	fox	18	case	14	foolish	11	boy	9	better
34	because	17	main	14	couple	11	life	9	obtain

Table 5.8: Frequently used words (and their frequencies) in rater descriptions of the similarities between fables.

scale, and second, if there is a similarity, describe it in a text box. Our prompt asked for "similarities about story structure and content, such as similarities in plots (what happens) and characters (desires and personality traits)." We presented each story pair to three raters. The unanimous agreement on the Likert question was 46.3%, with another 50.4% of cases showing a two-to-one majority. To control for nonsense input, we identified and discounted those individuals whose rate of participation in unanimous agreement was less than 20%; this affected 3.9% of the total vote count. As before, we took the arithmetic mean of the Likert-scale ratings to arrive at a gold standard (producing a more continuous distribution of ratings for training the regression model).

A look at the free-text similarity descriptions reveals some interesting, if anecdotal, insights. A histogram of the 75 most frequently-occurring words that appeared, after stemming and excluding stopwords, is shown in Table 5.8. Although these raters were never exposed to the SIG model or its emphasis on agent goals, attempts, outcomes, causal connections and so on (aside from the use of "desires" in the prompt), their language suggests an inclination toward comparing narratives along these structural lines: "get," "want," "try," "because," "plan," "action," "result," "thought," "consequence," "goal," and "obtain." Another series of words seems to describe thematic scenarios like those we identified as SIG patterns: "greed," "punish," "bad," "unhappy," "deception," "better." Most of the

Predictor Variable Sets	Correlation	RMSE	F-statistic
Propositional	0.0551	0.1986	p<.0191
Static	0.2729	0.1923	p<.0001
Dynamic	0.2117	0.1948	p<.0001
Propositional+Static	0.2724	0.1924	p<.0001
Propositional+Dynamic	0.2174	0.1947	p<.0001
Static+Dynamic	0.3257	0.1893	p<.0001
Propositional+Static+Dynamic	0.3299	0.1891	p<.0001

Table 5.9: Cross-validated performance of various linear regression models against continuous similarity ratings for 1,015 encoding pairs; (right) p-value of F-statistic for entire model for each variation.

rest of the key words refer to nouns and verbs, sometimes generic ones, that are invoked in multiple stories: "character," "get," "animal," "thing" and so on. The overlap between these words and our approaches to finding similarity is encouraging.

We trained the linear regression model on 100 predictor variables separated into three sets, one for each of our three similarity metrics. Variables regarding propositional similarity included the number of overlapping propositions between the two encodings and the closeness of the overlaps (as given by Equation 5.1). Each of the 80 static SIG patterns was included as a variable. For the dynamic analogy metric, we included the Dynamic Analogy Score, as well as the raw counts that constitute it (numbers of bound node pairs, relations of various types, and so on). These distributions were normalized and fit against the similarity ratings using M5 attribute selection, and evaluated using cross-validation. We ran the evaluation for all combinations of variable sets to gauge the relative impact of each.

The results are shown in Table 5.9. Propositional overlap variables by themselves were weak predictors of story similarity ratings, as compared to the other two sets, with a Pearson correlation coefficient of only 0.06. The variables regarding static SIG patterns and dynamic analogies were highly influential by comparison, with correlations exceeding .20; the combination of all variables yields a model which correlates with similarity ratings at .33. This model makes significant progress toward the prediction of story similarity, with an F-statistic of p<.0001. The root-mean-square error is .19, compared to .20 for the model with only propositional predictors. In fact, we note that the model including all <u>but</u> propo-

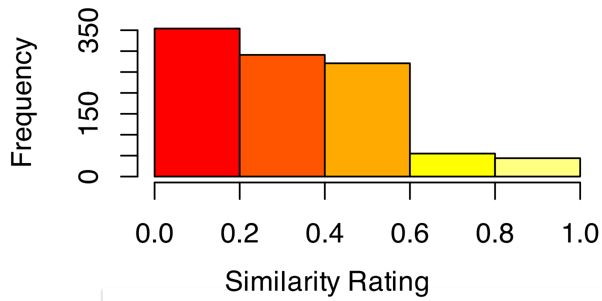


Figure 5.10: Distribution of similarity ratings given to 1,015 encoding pairs (over 300 unique story pairs from Collection B) used in the evaluation.

sitional predictors performed virtually as well as the all-inclusive model, as measured by both correlation coefficient and RMSE. Propositional modeling, while labor-intensive, did not provide helpful returns on the story similarity task.

A breakdown of predictor variables by t statistic highlights those that highly impact the regression equation. All but one are from the Static set: Mixed Blessing, Desire to Aid, Lost Opportunity, Mistaken Belief, Tandem Attempts, Conflict (type 1), and Persuasion are highly impactful ($p<.001$); Loss, Desire to Harm, Change of Mind, Goal Enablement, Perseverance, and Bad Side Effect also influence the regression ($p<.01$). The influential variable from the dynamic set is the amalgam (Dynamic Analogy Score), at $p<.05$.

The largest caveat of these results is the particularly lopsided distribution of similarity ratings—to most raters, nearly all story pairs had few to no appreciable similarities. Figure 5.10 shows the distribution of similarity ratings for the 1,015 encoding pairs we included in our regression analysis, with 1.0 corresponding to total similarity. (We solicited pairwise similarity ratings for 25 Aesop fables, 300 pairs in all, and used those judgments for multiple encoding pairs when there were several encodings for a single fable.) Only 99 encoding pairs (<10%) were rated above 0.5. An increase in the amount of training data, or an expansion of the raters' notion of story similarity, would mitigate this factor.

5.5 Conclusions

In this chapter, we set out to identify algorithms which could leverage the features of a SIG encoding to identify similarities and analogies between stories, a vital task for automatic analysis of narrative discourse. As part and parcel of this task, we also measured the "cost-benefit" balance of including full propositional modeling in a SIG encoding, which (as we saw in Chapters 3 and 4) is supported but not required by the schemata.

We first found that propositional modeling substantially changed the tenor of the corpus collection process by being considerably labor-intensive. Collection A, which included propositional modeling but not interpretative-layer encoding, and Collection C, with the opposite approach, were quite different: While A was limited to extremely short fables, C included far longer and more varied texts (on the scale of a few thousand words, versus a few hundred) while only taking about twice the annotation time per story. Annotators working on Collection B, who were required to encode both aspects, reported more difficulty with the propositional aspect for the task of conveying their understanding of story meaning. In other words, propositional modeling has a high "cost" not only in terms of the amount of narrative discourse that can be annotated in a given unit time, but also in its limited coverage of the aspects toward which annotators tend to gravitate when trying to find an appropriate mechanism for modeling their comprehension—goals, plans, outcomes, and causal inter-relationships.

In Section 5.2, we then focused on the "benefit" of propositional modeling: Given a carefully crafted temporal and propositional model of a corpus, what similarities and analogies can we find? By applying techniques for the detection of lexical and semantic similarity between two propositions, and developing a novel method for timeline alignment, we were able to find agreement between homogeneous encoding pairs (overcoming a small rate of just 10% direct propositional overlap). An evaluation showed that we outperformed a word-overlap baseline at detecting such inter-annotator similarities, even though the baseline worked from serialized versions of the same declarative propositions rather than the more rhetorical spans of source text. However, this success at finding homogeneous connections did <u>not</u> translate to the ability to find similarities between heterogeneous encoding pairs—that is, similarities between different stories. Rather, we found that the propositional overlap metric had a very low sensitivity to the story similarities present between two fables.

We then investigated two separate methods for finding story similarities and analogies based on the interpretative layer of the SIG. One "top-down" method took as *a priori* a set of 80 features, each defined as a static, archetypal SIG fragment representing a narrative trope which an encoding may or may not instantiate. Story similarity was then measured as the cosine similarity of the feature vectors drawn from two encodings. The second "bottom-up" method dynamically examined each pair of graphs for the largest possible isomorphic subgraph with consistent agentive bindings. The common subgraph represents an analogical relationship, and its size (the number of bound nodes and arcs) determines the degree of similarity between two stories. Like the propositional similarity method, both of these approaches found significantly higher agreement between homogeneous encoding pairs than between heterogeneous pairs.

In our evaluation, we observed that predictor variables regarding static SIG patterns and dynamic analogies were more effective, both individually and in tandem, than those involving propositional overlap at predicting reader assessment of story similarity in the Aesop collection. Combining both interpretative-layer methods allowed us to build a regression model that correlated with ratings of story similarity by a coefficient of .3257, virtually the same as the correlation of the model including all three methods (.3299). We also found that raters preferred dynamic analogy detection over the other two methods when presented with sample output and asked to rate the completeness and accuracy of a proposed analogy between two fables.

We interpret these findings as evidence that the SIG model represents notions of narrative discourse which are meaningful to the similarity and analogical retrieval task. The complete SIG encodes meaning that gives narrative discourse a cohesion not offered by predicate-argument structures alone. The "cost-benefit" analysis of propositional modeling shows a high cost, but a low benefit. In practice, SIG relations offer a significantly more useful set of symbols for designing an algorithm to detect similarities and analogies.

Chapter 6

Conclusions

This thesis has been an exploration of new methods for modeling discourse as a narrative artifact. While storytelling is ubiquitous, appearing in forms ranging from news to gossip to great works of literature, limited progress has been made toward the goal of procedurally isolating the "storiness" that separates a narrative text from an expository text or a collection of disjoint facts. In the future, an emphasis on the distinct and unique properties that make a narrative a highly tellable, easily remembered mode of communication will impact many areas of text processing, such as summarization, search and text visualization. For instance, two news stories may be related in terms of their themes but not by their facts; meanwhile, a thousand novels may show an overall trend toward social change through gradual changes in plotting and characterization. While certain newer techniques such as topic modeling [Steyvers and Griffiths, 2007] and narrative event chains [Chambers and Jurafsky, 2008a] point the way toward identifying meaning through latent lexical patterns, we take as inspiration work on identifying discourse relations, such as Rhetorical Structure Theory (RST) [Mann and Thompson, 1988] and the more recent Penn Discourse Treebank [Prasad *et al.*, 2008]. Though these models include elements that are considered essential to narrative, such as causality, they do not cover goals, plans and other aspects of agency that have been isolated by cognitive psychologists as crucial to the mind's conception of a story.

6.1 Summary of Findings

Literary Social Networks

We began our investigation in Chapter 2 with an applied, tractable idea about what "storiness" is, namely, the conversational relationships to be found among characters in novels based on their quoted speech patterns. Dialogue, particularly in the form of quoted speech, is an extremely common feature of 19th century British literature, and this experiment performed a "distant read" of a corpus of 60 novels totaling more than 10 million words. We took as motivation the opportunity to use narrative discourse modeling as a method to find evidence for or against literary theories that have been advanced by scholars on the basis of close readings of far smaller corpora. We identified two theories in particular that relate to the social worlds depicted in these novels: First, that larger communities tend to be less talkative, and second, that urban communities are more alienating (that is, less connected) than rural communities. By casting the notion of social connectedness as one of dialogue interaction, we found a conduit through which we could procedurally investigate these claims on the basis of all 60 samples of the genre.

We began with a technique that identified the characters in each novel, and grouped together coreferent named entities on the basis of title and gender. After detecting spans of quoted speech (those that occur between quotation marks), we proceeded to train a classifier to consider each quote and determine who among the list of characters, if anyone, is the speaker. We collected a development/testing corpus consisting of over 3,000 instances of quoted speech from passages of Austen, Dickens, Flaubert, Twain, Conan Doyle and Chekhov, and an analysis of this corpus found that over 80% of the quoted speech instances were in the form of a syntactic pattern such as "Person-Said-Quote" (where "Said" is a speech verb). Some of these categories entailed predictions that, collectively, identified the correct speaker of the quote with 97% or higher accuracy on the test set. We extracted feature vectors for each candidate-quote pair and built a model to predict a likelihood that the candidate is the speaker of the quote for each data point. After tuning the parameters of the learning methods, we achieved an overall accuracy of 83% on the test set for the task of determining who says what.

We then moved on to the larger task of characterizing the overall conversational networks to be found in our selection of 19th century literature. By taking quoted-speech adjacencies as evidence of interaction, but applying filters to remove false positives, we arrived at a graph in which nodes represent characters and arcs represent conversational interactions (weighted to be proportional to the amount of interaction—see Figures 2.1 and A.1 through A.3). As a check on the parameters we built into this extraction routine, we ran an evaluation in which human annotators determined conversational interactions in four chapters from four different works in the corpus (excerpts from Conan Doyle, Austen, Dickens and Henry James). Each potential character interaction was cast as binary data point (interaction/non-interaction), allowing us to calculate the precision and recall of our approach. Our method demonstrated a precision of .95 and a recall of .51, significantly outperforming two baselines.

Our aggregate study of all 60 social networks did not find evidence to support the literary theories which had been advanced. We found, for instance, a strong Pearson correlation of r=.42 between the size of each graph (the number of characters) an the average degree of each node, showing that networks do not break apart as they get larger, but instead become *more* connected. In addition, we found no evidence of a relationship between a novel's setting (urban or rural) and its size or connectedness. We did, however, find a striking relationship between an element of the novel's form—its perspective as a first- or third-person telling—and several measures of connectedness, including density. We inferred that this is due to the nature of a first-person novel as depicting one central character's experience in the social world, rather than a more omniscient view of the world as a whole.

This experiment demonstrated the tractability and the intrinsic value of modeling narrative discourse in the form of quoted-speech interactions. These promising results point the way toward future interdisciplinary collaborations in which the distant read can become a commonplace tool beside the close read.

Story Intention Graphs

The notion of "storiness" includes social connections among characters, but is not limited to them. In Chapter 3, we took a step away from the surface text to propose a new set of discourse relations for narrative. These relations, which we collectively call the Story

Intention Graph or SIG, aim to find a middle ground between what have traditionally been two separate modes of analysis for story understanding: lexical, syntactic or propositional features of the surface text on one side, a full planning or first-order logic model of the story-world on the other.

By giving a brief history of prior narrative models in Section 3.2, including several from the cognitive science literature, we motivated our design of the SIG as a model that carefully selects particular aspects of narrative to represent. These aspects, defined in Section 3.3, include character and object coreference, temporal relationships in an interval and modal logic (the "timeline" layer), and crucially, *agency frames* that indicate the goals and beliefs of agentive characters in the discourse (the "interpretative" layer, drawing from the contemporary theory-of-mind approach to narrative analysis [Zunshine, 2006; Palmer, 2010]). Spans of a source text, once embodied as actions on the story-world timeline, can be annotated as representing intentional attempts to trigger certain events, the outcomes to those attempts, and changes to the agents' mental states, among other roles in a cohesive structure. The SIG also feature relations that represent plans, linking goals to superordinate goals, and affectual impacts on agents.

The semantics of the SIG include logical entailments that are triggered by certain compounded relations; chief among these are a set of rules for establishing whether certain hypothetical events become "actualized" or "prevented/ceased" with respect to a certain instant of story time (that is, a particular state in the timeline). For instance, an agent may establish that it has a goal at State X, and at the next state X+1, an event may occur which actualizes the object of desire (e.g., rain arrives after a farmer wishes for rain). The SIG relates the mental state of desire to the successful outcome, giving the text greater cohesion by using the goal as a bridging element. Various permutations of nodes and relations can express a wide array of narrative tropes in the abstract; in Appendix B we show by example the capability of the SIG formalism to express many situations involving the interactions of goals, plans, attempts, outcomes, beliefs, time and causality.

The SIG schemata allows an annotator to provide a propositional equivalent of a text span within its corresponding timeline node, including a predicate and a set of arguments for the predicate's thematic roles. This is an "optional" aspect of the SIG, in that the agentive relations do not require them to be present. However, if propositions are present, they may greatly enhance the formal detail provided in a particular encoding.

Scheherazade

In Chapter 4 we described the implementation of the software platform that we have built as a tool for encoding, managing, analyzing and exporting SIG encodings. SCHEHERAZADE, as we call it, includes a custom transactional database to build and store arbitrary semantic networks. The particular semantics of the SIG relations are then applied to the database so that it can calculate actualization logic, interval logic associated with timelines, and other incident aspects of the SIG. We then described the manner in which SCHEHERAZADE supports the encoding of propositions; in our implementation, all verb predicates and nouns are drawn from the external linguistics databases including VerbNet [Kipper *et al.*, 2006] and WordNet [Fellbaum, 1998].

A graphical annotation interface allows trained annotators to construct SIG encodings from source texts. After iterating several times on the design with formative evaluations, we arrived at an accessible tool that allows users to model noun entities (such as characters), build propositions out of predicates and arguments, arrange events and statives in a representation of time (including alternate modalities), and instantiate interpretative-layer nodes and relations (agency frames, plans, outcomes and so on). A custom natural language generation module serializes every aspect of the formally encoded story as “feedback text,” making SCHEHERAZADE a “what-you-see-is-what-you-mean” semantic annotation tool. We made special mention of the model of English tense and aspect that we developed to properly communicate events and statives in a variety of temporal contexts. For example, the interface may call for an action that is “currently happening” to be realized from the perspective of a future telling of the story: “He <u>was</u> walk<u>ing</u>.”

Collections and Experiments

We concluded our investigation in Chapter 5 with several collections and experiments involving the SIG and the SCHEHERAZADE tool. Our goal was to determine the efficacy of the SIG for the story similarity and analogical retrieval task. We were also interested in determining whether propositional modeling was a necessary and cost-effective enhancement to the SIG for purposes of this task. As such, we recruited annotators to provide three collections, A, B, and C. Collection A had propositional and temporal modeling only; Collection C had interpretative-layer annotations only; Collection B included both aspects. Because propositional modeling is time consuming, we were limited to a corpus of short stories for Collections A and B; we chose the fables attributed to Aesop due not only to their brevity, but also to their high degree of thematic content (that is, easily identifiable goals, plans and outcomes). For Collection C, we moved away from Aesop and into much longer and more complex texts including literary short fiction, a news article, contemporary non-fiction and even medieval epic poetry (translations of *Beowulf* and *The Battle of Maldon*). Annotator feedback showed that interpretative-layer relations were easier to encode than propositions, and in some cases, the process measurably enhanced the annotators' appreciation of the source texts.

We developed three separate methods for identifying specific similarities and analogies between SIG encodings:

- By propositional and temporal logic (Section 5.2). By applying methods for finding the semantic distances between predicates and lexemes, we were able to find the "nearest common ancestor" for any two propositions. For instance, "the cat climbed" and "the lion pounced" would yield the intersection "a feline moved." For the special case of comparing homogeneous encoding pairs—two encodings by different annotators of the same story—we also described a method for finding an optimal alignment such that the overall temporal consistency and propositional overlap is maximized.

  We evaluated this metric to determine whether the semantic distance approach could identify "propositional paraphrases," in which two propositions encode the same essential idea but vary in terms of syntax or morphology. This served as a check for the sensitivity of the propositional overlap approach for the similarity and analogy task, as Collection A featured 20 pairs of parallel encodings with only 10% agreement overall in terms of the particular propositions used. Testing against a gold standard drawn from human annotators, we found that the approach significantly outperformed a word-overlap baseline on the task of classifying proposition pairs as paraphrases or non-paraphrases, with an overall F-measure of .58. However, despite this algorithm to normalize paraphrases, the propositional similarity approach did not succeed in detecting substantive similarities between stories.

- By static SIG pattern matching (Section 5.3.1). This approach applied the archetypal SIG patterns we introduced in Appendix B. Representing such high- and low-level narrative tropes as Loss, Mixed Blessing, Deception, Persuasion, and Recovery, these 80 *a priori* structures served the basis for feature extraction. We determined whether each encoding in Collections B and C instantiated each pattern, then compared these feature vectors to one another to determine the overall thematic similarity between the stories described by any two encodings. A check of the soundness of this approach revealed significantly higher similarity scores for homogeneous encoding pairs than for heterogeneous pairs.

- By dynamic SIG analogy detection (Section 5.3.2). As opposed to the previous approach, which is a "top-down" algorithm depending on *a priori* features, this is a "bottom-up" algorithm that finds the largest possible isomorphic subgraph that exists between two SIG encodings. Each subgraph must have a consistent binding in which one agent is matched with one opposing agent; all other elements that are analogically matched, including interpretative-layer nodes and timeline propositions, must conform to these agentive equivalences.

  The results show that substantive analogies can be found among many SIG encodings in Collections B and C; as a check, we determined that homogeneous pairs have substantially larger analogies than heterogeneous pairs. Two examples of procedurally detected, heterogeneous SIG analogies are shown in Figures 5.7 and A.4.

To determine the comparative effectiveness of these three techniques, we ran a two-part evaluation. For the first part, we asked human raters to read two fables and judge the similarities suggested by one of the three approaches. We employed two Likert scales that approximated precision ("how accurate are these suggested similarities") and recall ("how complete"). The results showed that the propositional overlap method was precise but insensitive and the static SIG pattern method was sensitive but imprecise. The dynamic analogies significantly outperformed both of the other two metrics on a combined fitness score, having performed comparably to one or the other on both accuracy and completeness.

For the second evaluation, we asked raters to judge the degree of similarity between two fables directly, and trained a linear regression model against those judgments with predictive variables from all three metrics. We also trained variations of the model that only included subsets of the variables, in order to gauge their comparative effectiveness. We found that the static SIG patterns had the highest predictive power, followed by variables based on dynamic analogies; together, both approaches trained the model to correlate to the similarity ratings at a Pearson coefficient of .33 (significant by the F-statistic to $p<.0001$). The propositional similarity features were not nearly as powerful; adding them to the model did not appreciably change its performance.

We concluded this chapter by assessing a low cost-benefit ratio to propositional modeling as an enhancement to a SIG encoding. The relations themselves proved to be effective at providing the basis for extracting meaningful similarities and analogies; the encoded propositions did not provide a significant benefit, especially compared to the highly involved process of gathering them.

6.2 Limitations and Future Work

6.2.1 Literary Social Networks

The results of the Victorian corpus experiment carry certain caveats. First, as we noted in Section 2.4, more work is necessary to ensure that the extraction of character entities and their coreferent mentions is proper. Combining Dr. Jekyll and Mr. Hyde may be quite difficult (and perhaps unwise) but we must, for instance, accurately normalize certain recurring aliases such as “Eliza” or “Lizzy” for Elizabeth Bennet in *Pride and Prejudice*. One potential solution is a probabilistic model that may be trained using such features as edit distance and the distributions of the referents in the text (e.g., the narrator refers to a character with a formal name, but a conversational partner uses a nickname in the same scene).

Another area to improve the generality of our model is in quoted speech attribution.

More work is needed to test the applicability of our syntactic categories for quote attribution to other genres and languages, although we note that our results stand even if machine learning results are used without any of the automatic category predictions. The QSA corpus remains small and may be expanded to other authors, texts and languages. Another direction to take this experiment would be to move to a learning model designed for sequence analysis, rather than rely on feature extraction to provide contextual information such as the most recent attributed speaker.

As far as the results of the analysis of all 60 Victorian novels, we must remember that lack of evidence is not evidence of lack. We did not find evidence in support of the previously asserted literary theories, but our metric carried certain assumptions, such as that any salient social interaction is manifest at some point in an exchange of quoted speech. As we saw in our evaluation in Section 2.6, our measurement tilts toward precision rather than recall; it is possible for certain relationships to surface only in reported speech or back-channels [Agarwal and Rambow, 2010]. Further work is necessary to determine the impact of reported speech on our results in Section 2.7.

Beyond these caveats, there are several directions to take this line of work in the future; two of these interest us in particular. The first direction is an analysis of conversation content. As we already know who is speaking with whom in this corpus, we can aggregate the quoted speech that falls along each edge in the conversational network. A preponderance of certain words or topics [Chang *et al.*, 2009] can characterize each edge as belonging to a certain category of interaction, such as *neighborly* or *professional* (to borrow from a similar study of several Victorian texts [Sack, 2006]). Automatically categorizing edges as belonging to a certain archetype of social interaction would greatly enrich the networks we have already extracted, and allow us to study the properties of more nuanced networks.

The second appealing direction for future work changes the nature of the social network from an *aggregate* graph to a *temporal* graph. We have seen examples of the former, which collects all the conversations throughout the entire novel into a single, static network. However, by splitting the text into segments and extracting a social network for each segment, we can introduce narrative time as a dimension to this analysis. One might apply the statistical methods that have been developed for so-called longitudinal networks, in genres such as email, to model the comings and goings of characters throughout the course of a discourse [van de Bunt *et al.*, 1999; Huisman and Snijders, 2003; Blei and Lafferty, 2006; Kossinets and Watts, 2006; Goldenberg and Zheng, 2007].

Such an analysis would lend itself to further collaboration with literary experts, as temporal networks begin to resemble various graphical notions of story "plot." The influential theorist Viktor Shklovsky, for instance, wrote that the Dickens novel *Little Dorrit* "moves simultaneously on several planes of action" [Shklovsky, 1990, 124]. These plot strands are then "interwoven" with one another, either "by involving the characters of one plot line in the actions of another plot line or by stationing them in the same place." However, the nature of these connections is left mysterious to us, the reader, until the end of the novel, when certain key relationships are revealed. If many novels are constructed using the same architecture of multiple interwoven plot strands, an analysis of longitudinal social networks may detect and illuminate this technique.

We took some initial steps toward this experiment by adapting our current approach. We divided each text by chapter and complied a separate social network for each chapter; long chapters are further partitioned. We then wrote a visualizer that plots individuals as strands that traverse a long horizontal axis representing time (from the beginning of the discourse to the end). Sample output from this visualizer for Austen's *Pride and Prejudice* is shown in Figure 6.1. (The graph for *Little Dorrit* appears in Figure A.5.) Characters who are in conversation are drawn as adjacent strands. Because characters move from one social setting to another, strands must rise, fall and cross over one another along the vertical axis in order to remain as close as possible to their conversational partners. The visualizer attempts to arrange the strands to minimize the total amount of vertical movement required to keep conversational partners close to one another. The result is that different cliques appear above and below one another, giving the appearance of "plot threads" that weave through the discourse. In this example, for instance, we can see that *Pride and Prejudice* begins with a flurry of conversation as many characters introduce themselves to one another at Netherfield Park. The story then fragments somewhat, with Elizabeth speaking with different parties in turn; she grows close to Darcy in the second panel of Figure 6.1. Other social events in the novel are seen as sudden confluences of threads.

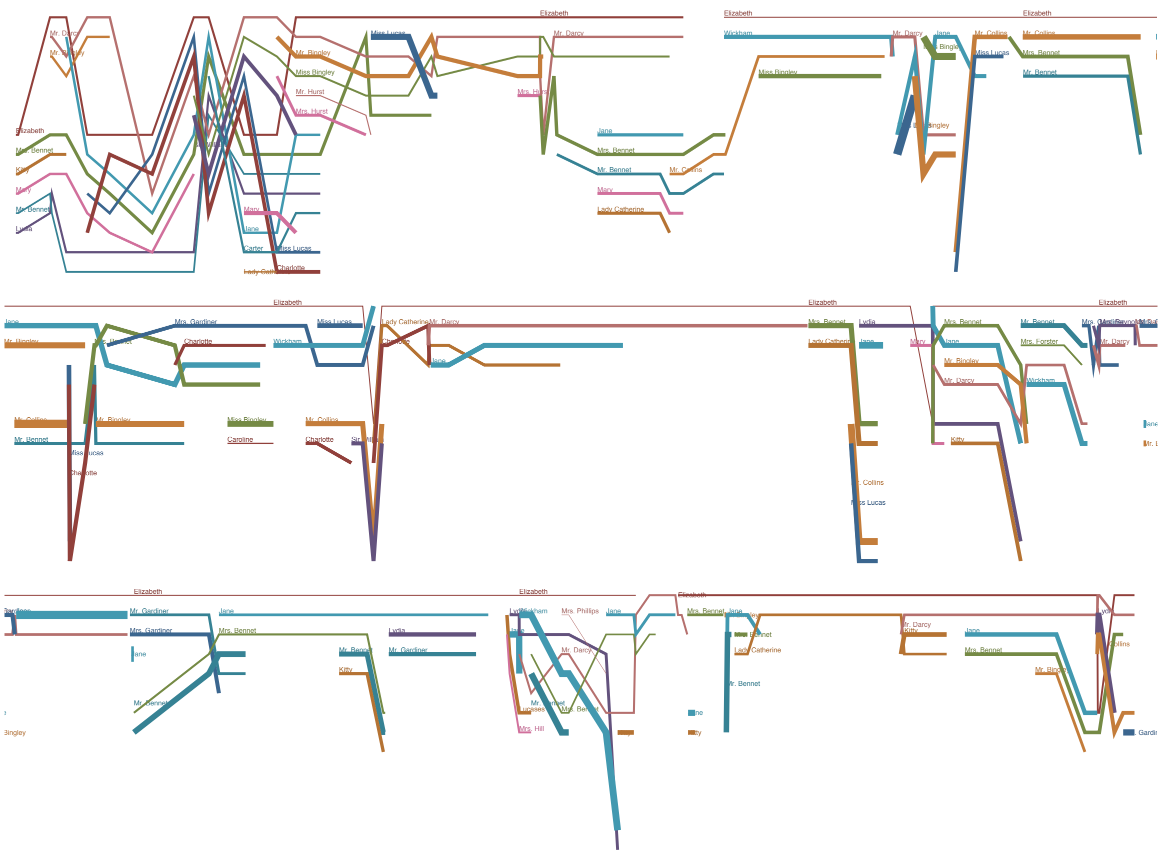


Figure 6.1: Longitudinal conversation network for Jane Austen's *Pride and Prejudice*.

We believe that this initial result suggests that with further development, we can more fully model a narrative as a dynamic social network. In addition to the potential for literary analysis, such results may find use in summarization and human-computer interaction, assisting readers with navigating and understanding novels.

6.2.2 Story Intention Graphs

There is always more than one way to read a text. From a design standpoint, we must remember that the Story Intention Graph schemata is designed for a particular mode of narrative and literary analysis. As we saw in Section 5.1, certain stories resist an agentive reading, and may be more about a particular time, place, or idea. Narrative is a vessel for communicating thoughts, images and emotions of all stripes, so we must be careful to not limit the definition of "narrative" when we claim that the SIG is a model of narrative discourse. The import of this issue depends on what one intends to do with the SIG. An agent- and goal-centric approach is the norm in certain areas of narrative intelligence such as interactive narrative, where the plans and goals of a generated story must be recomputed on the fly depending on player input [Riedl *et al.*, 2003]. Here, as elsewhere, the model may serve to organize a data bank of previously seen narrative scenarios for use in estimating player expectation of a certain outcome or development. In other contexts, however, it may be too general for the task at hand, or missing key elements that are locally considered to be crucial to narrative coherence. For example, the SIG does not consider the language of the source text, so certain linguistic clues that trigger reader reaction will find no formal mechanism to do so in this representation (as the node representing source text can only link to a timeline action or stative). One possible extension to the model would be to allow source-text nodes to interact with interpretative content directly, so that a word or phrase does not have to involve a *fabula* action in order to take part in the story's agentive coherence. In general, except for Collection C, the SIG has had limited exposure to corpora and learning contexts outside the fable realm; while we strove to make it domain-independent, more work is needed to show that this is the case.

The largest and most pressing caveat about the SIG, moreover, is the unknown quality of its practicality. Determining the potential for automatic extraction is the first of the next steps. For now, a review of prior work in various adjacent areas can illuminate the feasibility of constructing encodings from source texts.

One promising and recent analogue is work by Goyal et al. [2010a; 2010b] toward the automatic identification of plot units in Aesop's fables. Plot units, as we discussed in Section 3.2.3, are a model of affect state transitions devised for use in a knowledge-driven story understanding system [Lehnert, 1981; Lehnert *et al.*, 1983]. Goyal et al. bring a series of linguistic corpora to bear on the Aesop corpus and are able to classify clauses as belonging to the M (mental), + (positive) or − (negative) affect states with a combined F-measure of .45. On other fronts, Appling and Riedl [2009] use the plot unit as a representation for a machine learning model of story summarization; Chen [2009] uses WordNet, PropBank and other tools to ascribe mental states to agents in children's stories; Wiebe has long examined the issue of tracking point-of-view in narrative [1994], and, along with colleagues, has collected a large corpus of annotated news articles in which "private state frames" including opinions, emotions, sentiments, speculations, and evaluations are indicated for each agent [Wiebe *et al.*, 2005].

The notion of automatically interpreting goals and plans from a text is a well-known problem called *plan recognition* or *intention recognition* [Carberry, 2001; Sindlar *et al.*, 2008; Ramírez and Geffner, 2010; Sadri, 2010]. This has been studied in the context of question-answering systems such as trip planners, which must try to estimate what the user is ultimately asking for when he or she issues a query. For instance, if a tourist asks where the nearest post office is, a helpful QA system would infer that the tourist intends to travel to that post office and give directions from his or her current location. The inquiry about the post office's location was only one step in a multi-step plan that such a system would estimate to be the likely intention of the user. One recent implementation is Mott et al. [2006], who apply Markov chains to estimate which of 20 goals is probably being pursued by a user in an interactive narrative environment, based on observations of its actions.

As far as the timeline layer is concerned, the extraction of temporal and propositional metadata from surface text has been a longstanding challenge, and one that interacts with affect and plan recognition [Lapata and Lascarides, 2004; Mani and Wellner, 2006; Chambers and Jurafsky, 2008b; Harabagiu *et al.*, 1995]. Nonetheless, we believe that with a sustained effort to build upon the DramaBank corpus with hundreds, if not thousands of encodings, we can follow in the footsteps of the Penn Discourse Treebank, for which discourse parsers are currently being developed [Lin *et al.*, 2010].

A first step toward this goal would be to "flatten" the model. As the SIG combines several types of relations and involves inferred content, it may be impractical to attempt to extract all elements of an encoding at once. Instead, we can reduce the dimensionality of a SIG encoding and train models to address certain key aspects before examining their interrelationships. For instance, we can factor out the temporal information in the timeline layer and label nodes in the text layer (that is, spans of source text) as fulfilling one or more of six interpretative-layer functions: actualizing harm, actualizing aid, actualizing or ceasing a goal frame (intentional state), actualizing or ceasing a belief, portraying a goal-

Actualizes Harm	**Actualizes Aid**
So at length the Eagle consented to do the best he could for him, Gurov, who was sitting in the stalls, too, went up to her and said in a trembling voice, with a forced smile: "Good-evening." One day the Lion got entangled was a pistol shot from the woods, followed closely by another. Then silence. The old lady's head jerked around. She could hear the wind move through the tree tops like a long satisfied insuck of breath. he could not resist saying: "If only you knew what a fascinating woman I made the acquaintance of in Yalta!" But it was all in vain, dashed to pieces doomed to die he hid himself, then, bereft of pleasure, in his fen-refuge he laid down his life, his heathen soul; there Hel embraced him.	the beaten one went and hid himself and succeeded in freeing him from the Serpent and "I ain't a good man," The Misfit said after a second ah if he had considered her statement carefully, "but I ain't the worst in the world neither." into which she could get her bill with ease. "I thank Thee, Lord of all peoples For all those joys that I on earth have known. Now, my Maker mild - I have most need That thou to my ghost should grant good. That my soul to Thee may journey, Into thy kingdom - O lord of the Angels, May pass with peace - I do desire of Thee That the hell-fiends may not hurt it." when the Eagle knocked it out of his hand, hen once, however, he was thus disarmed, "There's something pathetic about her, anyway," he thought, While the victor flew up on to the roof of the stables
Intentional State Actualization/Cessation	**Goal-Directed Action**
Then I got it, ol' Dusty here is making sure that every bug smasher from Mount Whitney to the Mojave knows what true speed is. when the horror perceived him; she was in haste, wanted out of there, to protect her life, when she was discovered; quickly she a noble one had seized tightly, then she went to the fen; she made up her mind she would never take an alcoholic case again. but the people were so used to hearing him call that they took no notice of his cries for help. nearer he stepped forth, taking then with his hands a stout-hearted warrior from his rest, reached towards him the foe with his palm; quickly he grasped the malice thoughts and clamped down on the arm. I was married to him. I have been tormented by curiosity; I wanted something better. "I thank Thee, Lord of all peoples For all those joys that I on earth have known."	Opposition leaders said the government's violent crackdown had strengthened their demands for an immediate change of government. was not long to when that the battle-shirkers gave up the forest, cowardly troth-breakers, ten together, who had not dared before with javelins to fight in their liege-lord's great need but they, shamed, bore shields, war-clothing, to where the old man lay; On the landing above them two schoolboys were smoking and looking down, Al-Wafaq lawmakers, who hold 18 of 40 seats in parliament's lower house, said Thursday they would resigned en masse from parliament in solidarity with the protesters. Not at all him in a troop the hand-companions, nobles' sons, around him stood with valour in battle, but they sunk to the forest, to protect life; Then he looked at her intently, and all at once put his arm round her and kissed her on the lips,
Belief Actualization/Cessation	**Causality**
folk-chiefs arrived from far and near across wide regions to behold the wonder, the foe's foot-prints; his parting from life did not seem mournful to any man of those who the gloryless foe's track observed, The idea of so insignificant a creature ever being able to do anything for him amused the Lion but you may turn yourself into a bag of meal hanging there, if you like, he Cottager was afraid of him no longer, but the people were so used to hearing him call that always spoke ill of women, and when they were talked about in his presence, used to call them "the lower race." and envious of the birds he saw disporting themselves in the air, he laughed at them for their pains. but the people were so used to hearing him call that they took no notice of his cries for help. that he readily agreed that this should be done.	and set fire to it → It quickly caught fire he secretly considered her unintelligent, narrow, inelegant, was afraid of her, and did not like to be at home. → He had begun being unfaithful to her long ago–had been unfaithful to her often, I will come and see you in Moscow. → And only now when his head was grey he had fallen properly, really in love–for the first time in his life. The other was Della's hair. → "Twenty dollars," said Madame, lifting the mass with a practised hand. "Give it to me quick," said Della. She got up and went quickly to the door; → He went up to her and took her by the shoulders to say something affectionate and cheering, and at that moment he saw himself in the looking-glass. A fisherman was catching fish by the sea. → The monkey came and grabbed the net, thinking that he too would go fishing.

Table 6.1: Representative spans of text from the DramaBank corpus associated with six narrative functions as suggested by the SIG relations.

directed action, or having a causal relationship with another span. It is a simple matter to transmute an encoding into a list of sentences, one in each class, which may serve as training data for a more narrow classification effort. For purposes of example, a selection of the DramaBank corpus as flattened in this manner is shown in Table 6.1.

Beyond the challenge of practical extraction from surface text, there are several directions for future work with the SIG model. One is in regard to the author's telling of the story. We have mentioned that there is a process of selecting and ordering events inherent to a narrated discourse—we may well ask, who is the narrator, and what are his or her own goals in making these decisions [Barry and Elmes, 1997; Fayzullin *et al.*, 2007; Montfort, 2007]? Because the SIG preserves both the surface discourse ordering and the story-world timeline ordering of events, we can perform a corpus analysis that aggregates the "information release" strategies of narrators in different genres. Naturally, any event which the narrator never chooses to reveal will not be in a SIG encoding of that story, unless it is strongly implied. However, a model of story reception that tracks what is known and unknown at each point during the story's telling can reveal key moments when the revelation of a past event changes the reader's perception of a current or future situation (see Section B.6.2).

A model of the process of reception would also be a fascinating vehicle for the study of reader affect. Narrative is communicative experience in which the receiver is open to feelings of surprise, suspense and other emotions as part of his or her projection into the story-world and the minds of its agents. Understanding what emotions are triggered in a receiver, and why, would have both intrinsic value in understanding this cognitive process [Brewer and Ohtsuka, 1988; Gerrig and Bernardo, 1994] and extrinsic use in story generation systems [Bailey, 1999; Cheong and Young, 2006; Fitzgerald *et al.*, 2009]. In this case, the representation can be extended to include reader affect metadata, such as with a label classifying each source-text node as triggering one or more affectual responses. These annotations can then be matched against other thematic features to determine what story content may have provoked each reader response.

In general, we see the SIG as a framework which can be extended as needed with additional types of labels, or reduced as needed to a simpler reflection. Future work can explore its fitness for a wide range of applications that might leverage a corpus of descriptive models of narrative cohesion. For example:

1. **Literature humanities and cultural studies.** Examining the collected works within a single genre, by an author, or on a topic, for broad trends and comparisons to other works. Assessing the way that literary storytelling has advanced through the centuries. Extracting thematically rich features to aid in searching, clustering and visualization.

2. **News clustering and summarization.** Detecting news articles that are thematically similar to one another, and to reference stories, even if they take place in different social spheres or parts of the world. Selecting salient details for a news summary based on their significance to the overall story.

3. **Question answering.** Understanding how facts that are gathered from structured sources can be combined into clear and cohesive narratives. Separating relevant facts from those that have no bearing on the answer based on a model of narrative import.

4. **Online social network analysis.** Analyzing the narrative content in millions of posts of personal experiences published on social networking services such as Twitter. Discovering broad trends and local aberrations in users' everyday, self-reported personal experiences. Determining relevance (i.e., interesting stories) for use in searching through this high volume of real-time information.

While each of these genres has different properties at the textual level, the relations we have proposed would be relevant to all of them at the level of interpretation. This suggests cross-genre clustering as well, such as determining narrative trends that begin in a historical period of news text and later spread to the literary fiction of the day.

6.3 Contributions and General Conclusions

In this thesis, we have explored methods for modeling narrative discourse through relations relating to social connectedness, time and agency. The contributions have been:

1. A method for extracting social networks from literary novels based on patterns of quoted speech; a related method for attributing quotes to their speakers in the text; the interdisciplinary application of this method to questions in literary analysis. A training corpus for the quote attribution task.

2. A novel set of discourse relations relating to the narrative structure of the discourse beyond social connectedness. These relations organize a text into a structure called a Story Intention Graph, or SIG.

3. A publicly released software platform, SCHEHERAZADE, which facilitates the annotation of text according to the SIG model. An investigation into the relationship between a formal description of time and English tense and aspect.

4. A corpus of 110 SIG encodings, collectively known as DramaBank, annotated from a collection of stories. These encodings are mostly drawn from the fables attributed to Aesop, but include examples of other narrative discourse such as the epic poem *Beowulf*.

5. Methods for finding thematic similarities and analogies between stories, by means of per-story SIG encodings. An application of the algorithms to DramaBank that shows analogical connections between DramaBank stories. An evaluation of the analogy detection algorithms.

Our motivation for designing the SIG was to create a set of relations for modeling narrative that provided for certain types of logical inference while not crossing the line from *descriptive* to *prescriptive*. We showed how it bridges textual annotation to a model of agency by achieving four major goals:

- It is formal, so that we can apply transitive closure rules, extract rich narratological features, and procedurally find meaningful similarities across multiple encodings;

- It is expressive, capable of articulating a large and extendable array of narrative scenarios, from the common to the obscure, that deal with agentive characters and their desires;

- It is robust, supporting a complete propositional modeling of the surface text but not requiring it; and
- It is simple enough to be accessible to trained but non-professional annotators, especially domain experts in literature and creative writing.

While much progress remains to be made on the process of automatically extracting these relations from a text, our experiments in drawing insights from SIG encodings have shown that this is indeed a goal worth pursuing. The relations unlock thematic aspects of a narrative text which, to date, have not been exposed through lexical and syntactic methods.

In fact, we began our investigation with the assumption that narrative modeling had to begin at the propositional level with annotation in the style of the Penn Propbank [Kingsbury and Palmer, 2002]; our experimentation disavowed us of this notion and pointed the way first to a descriptive model of belief, intention and agency, and then to the near-elimination of propositions in the SIG timeline in favor of these relations. In an evaluation, we found that formal propositions do not contribute to a model of the similarity and analogy task as significantly as features drawn from agentive relations. More broadly, this suggests that even state-of-the-art methods for semantic analysis at the level of propositions (i.e., semantic role labeling) cannot identify the essence of narrative meaning, and that the set of relations we have proposed may be a viable alternative for pursuing this vision. The social network extraction experiment, as well, found results by looking for relations beyond the word level—together, both sides of the study show that relations between entities, rather than models of word use alone, are effective strategies for understanding and comparing narrative corpora.

Our society is moving ever more each year toward living, learning and socializing through online media. Perhaps in the near future, we will transition from telling stories *through* our computers, and instead tell our stories *to* them. As good listeners and mediators, such systems will be able to show compassion in their interactions with us, and help us understand how others are the heroes of their own stories.

Appendix A

Additional Sample Visualizations

This appendix provides five additional visualizations of narrative discourse. The first three are conversational networks extracted from Ainsworth, Austen and Trollope using the methods we discussed in Chapter 2. The fourth is an encoding of the analogical intersection between the two fables "The Lion In Love" and "The Dog and the Wolf", using the SIG formalism introduced in Chapter 3. This analogy has been procedurally generated from annotations of these fables, using the method discussed in Section 5.3.2. The full texts are provided in Appendix D. The last figure shows a longitudinal social network of Dickens's *Little Dorrit*, generated using the experimental technique discussed in Chapter 6.

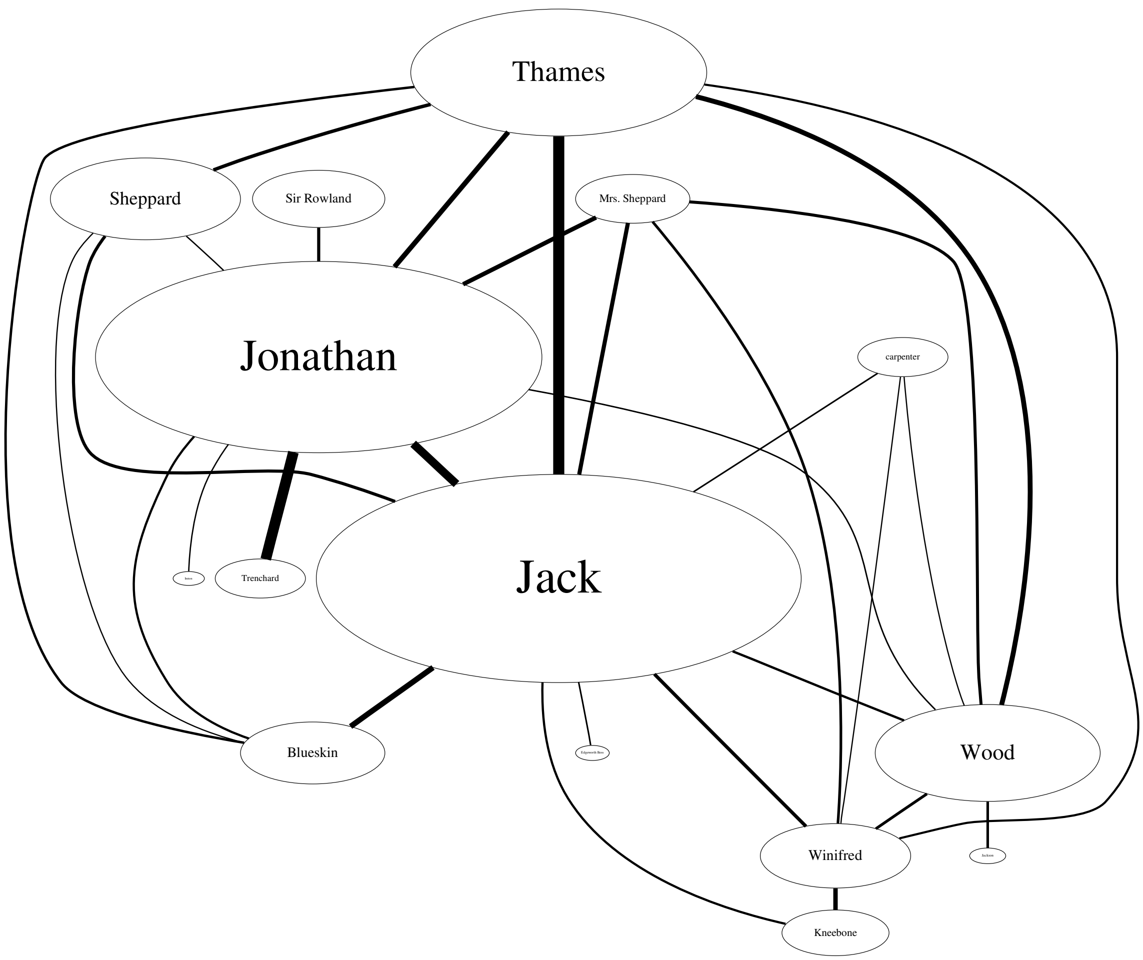


Figure A.1: Conversational network for Ainsworth's *Jack Sheppard*.

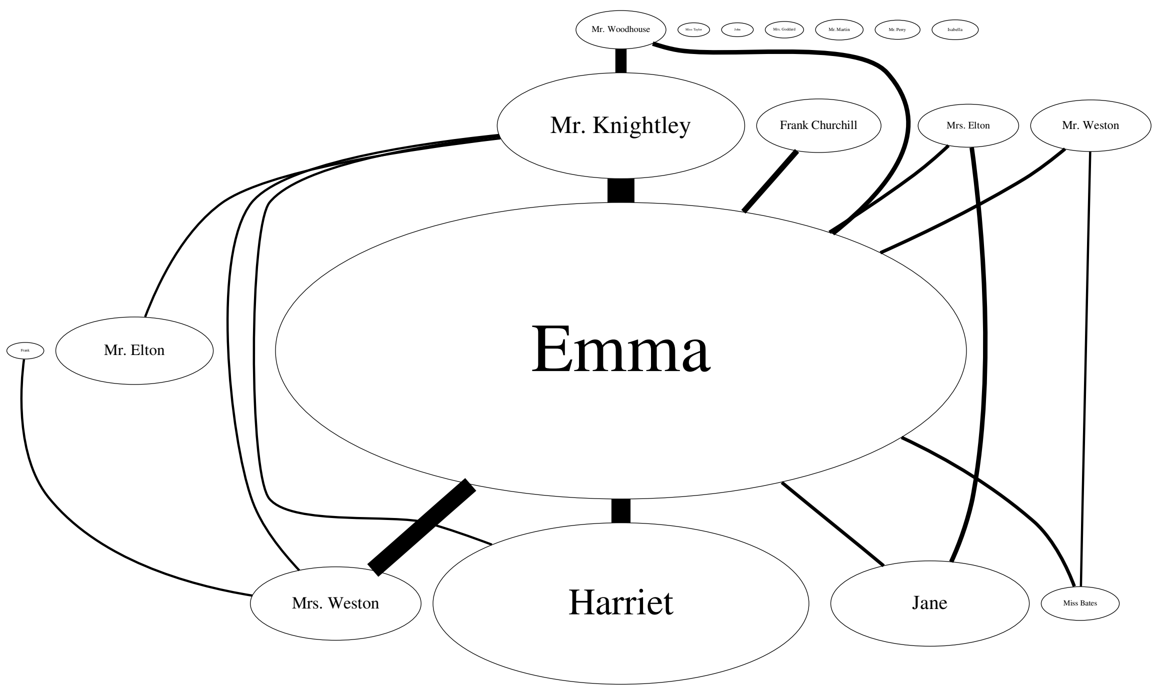


Figure A.2: Conversational network for Austen's *Emma*.

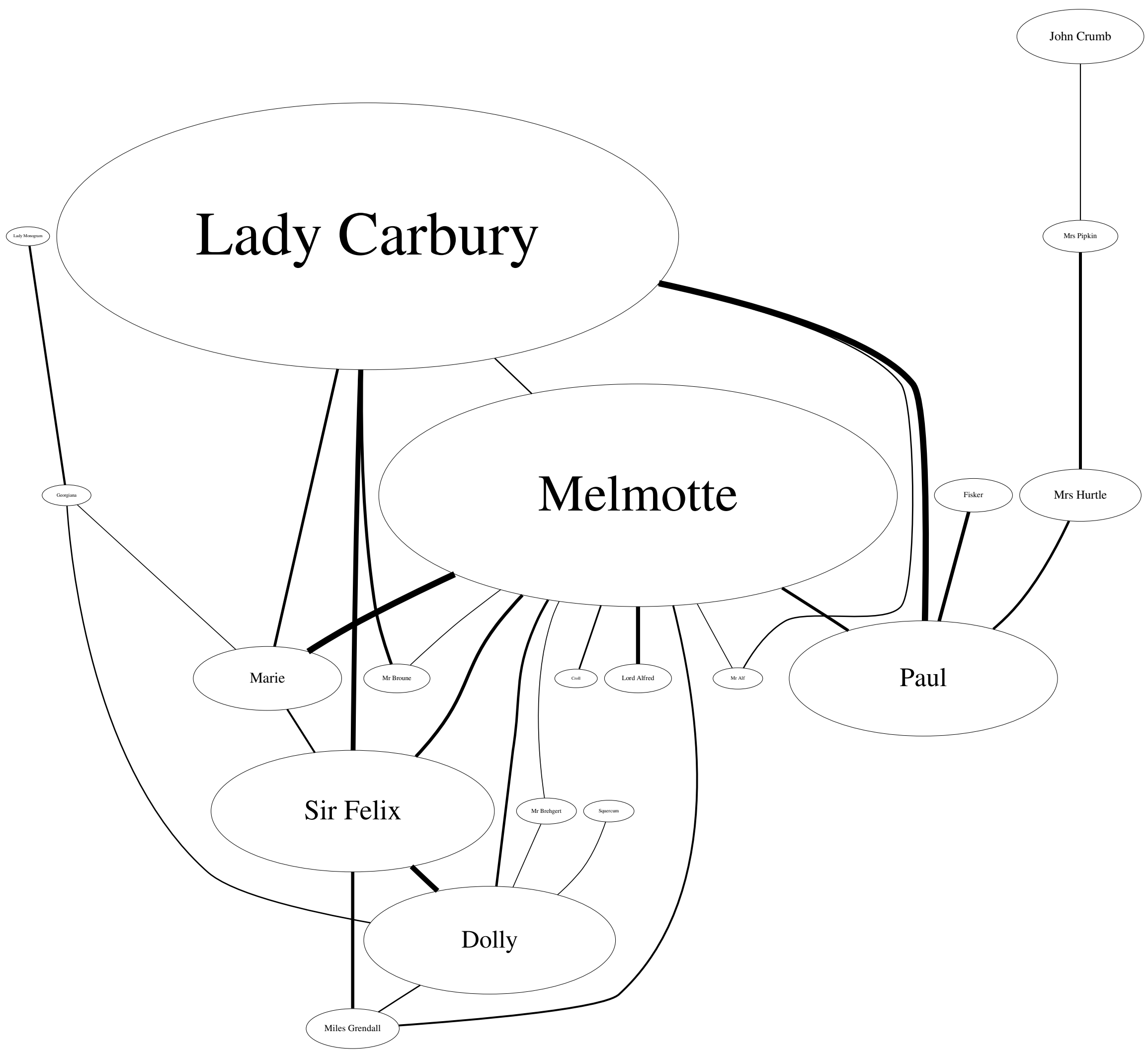


Figure A.3: Conversational network for Trollope's *The Way We Live Now*.

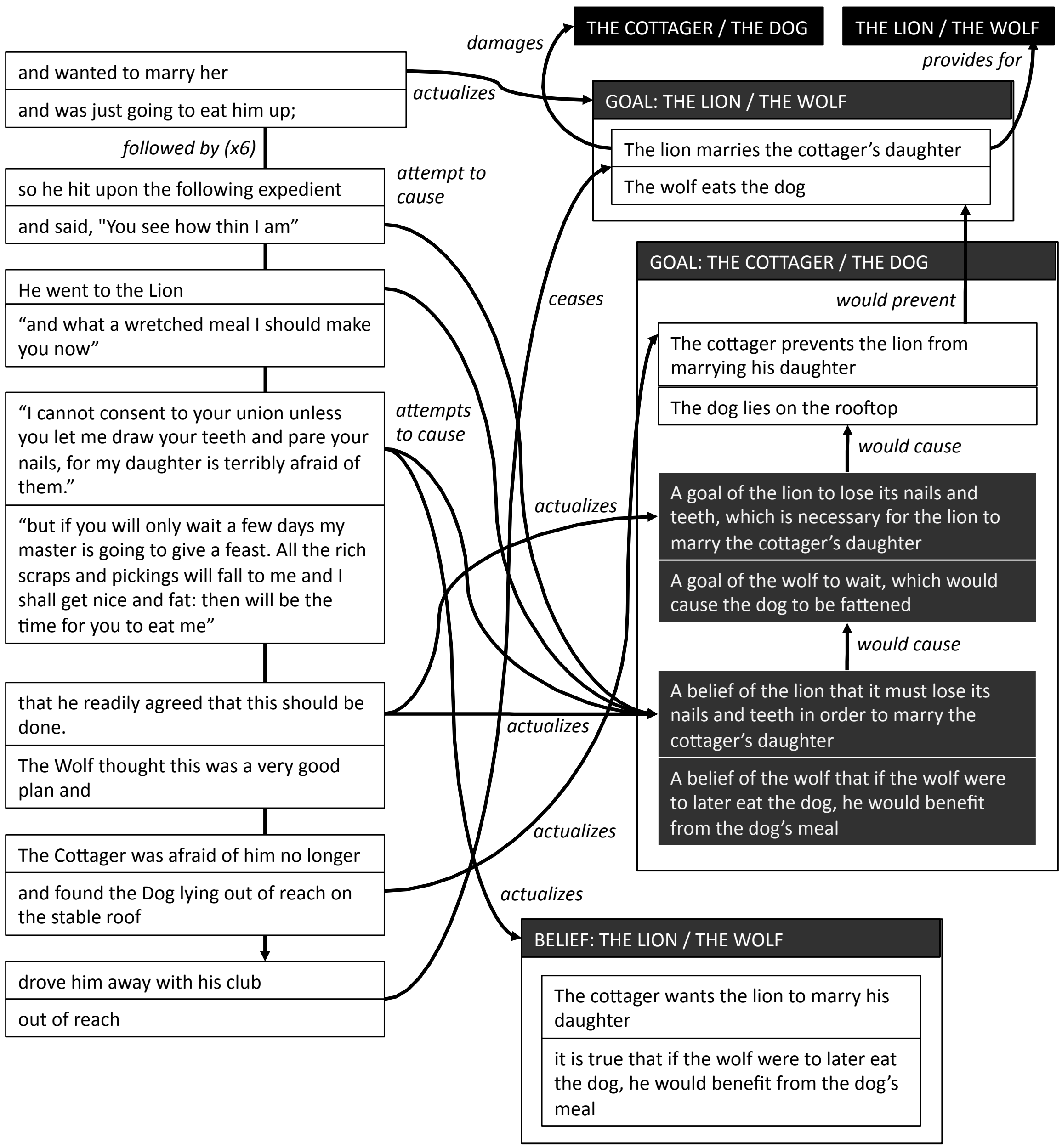


Figure A.4: Procedurally drawn analogy between collected SIG encodings of "The Lion In Love" and "The Dog and the Wolf".

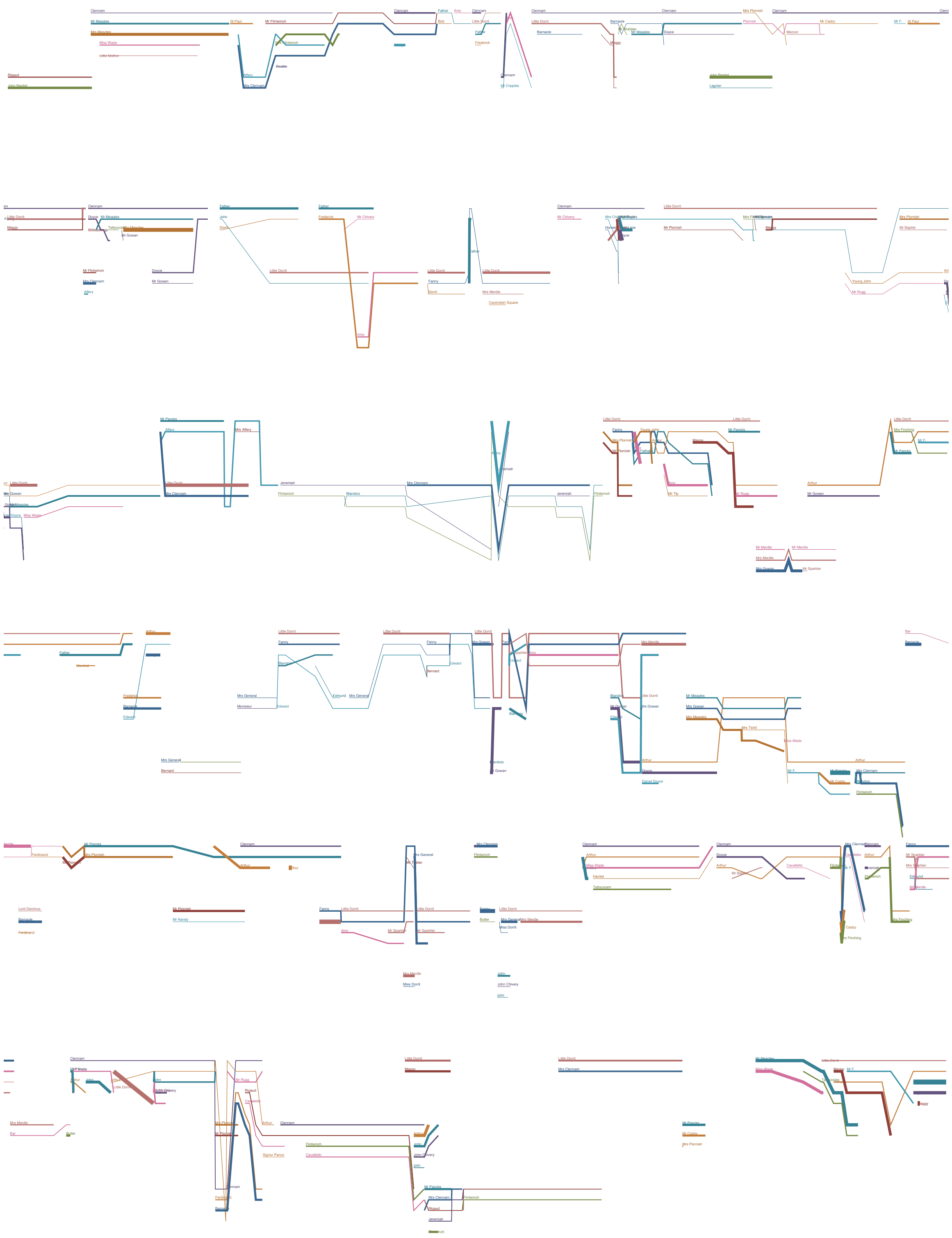


Figure A.5: Longitudinal conversation network for Dickens's *Little Dorrit* (in six segments).

Appendix B

Expressibility of SIGs

Section 3.3 defined a set of relations for representing a narrative by its agents, their problems and their strategies. In this appendix, we more fully explore the expressive range of the SIG—its generative power to cover the types of narrative scenarios we aim to identify when they occur in narrative discourse. Each scenario can be thought of as being minimally described by a **SIG pattern**, which is a fragment of an hypothetical encoding (one not associated with an actual discourse). By building notable patterns out of nodes and arcs, independent of any particular story, we can show the range of distinct narrative scenarios that our model can discriminate.

A SIG pattern is like a plot unit [Lehnert, 1981] in that both are small, connected graphs that conform to their respective schematas and represent narrative scenarios and tropes that are sometimes known by popular names (such as revenge or suspense). There are no knowledge structures inside Proposition (P) and Interpretative Proposition (I) nodes, giving SIG patterns domain independence. Like plot units, SIG patterns can be chained together to form complex stories involving multiple thematic turns.

We say that a particular SIG encoding *covers* a pattern if the abstract set of nodes and relations in the pattern is instantiated by the encoding. In general, this means the pattern is isomorphic to a subgraph of the encoding (respecting the types of nodes and arcs). We more fully explore the mapping of SIG patterns onto encodings in Chapter 5.

Figure B.1 gives a demonstration of this process, though we leave the technical details to Appendix C. B.1(i) is a strict subgraph of the encoding for "The Wily Lion" we gave

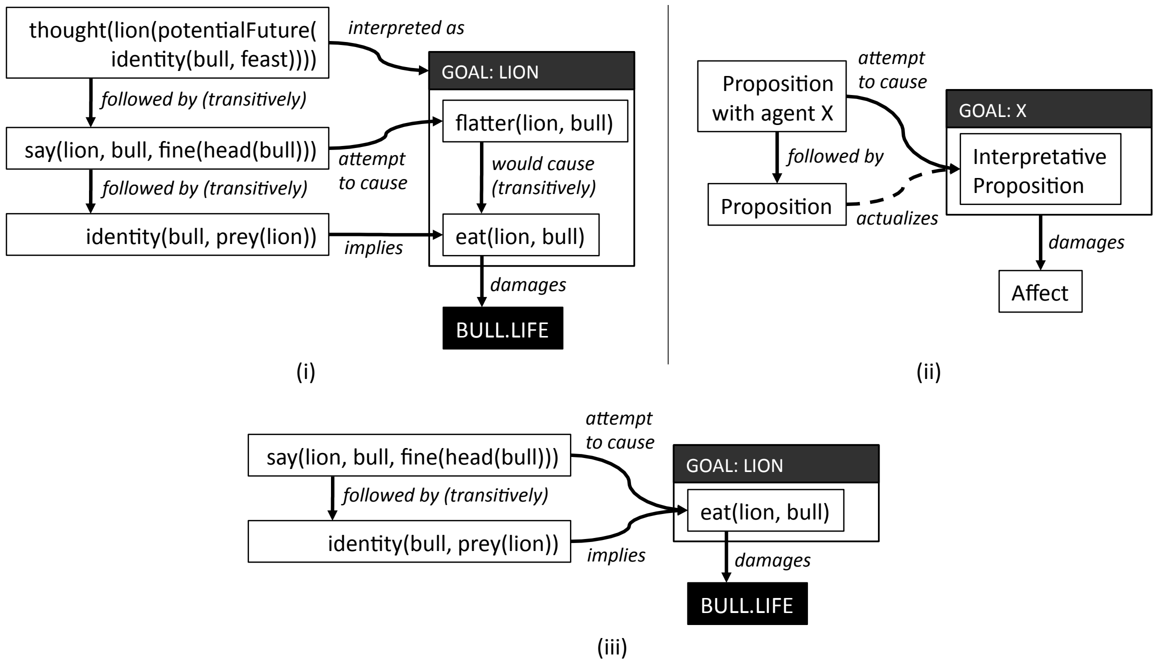


Figure B.1: Example of the coverage of a SIG pattern. Clockwise from top left: A subgraph of the encoding for "The Wily Lion;" a SIG pattern called *Deliberate Harm*; a transformation of the subgraph that is isomorphic to the pattern.

in Figure 3.18, with two exceptions: First, intermediate timeline events are replaced by a single, transitive *followed by* arc; and second, the intermediate steps of the lion's causally connected plan are collapsed into a single, transitive *would cause* arc between the first plan step (flatter) and final plan step (eat). B.1(ii) is a representation of a pattern called *Deliberate Harm*. It abstractly describes a situation in which an agent "X" acts with intention to harm some agent, and later the latter agent is harmed in the way that X envisioned. We use a dotted line to represent either of the three arcs that "trigger" an actualization (Section 3.3.2.1). While B.1(i) and B.1(ii) are not strictly isomorphic graphs, they become isomorphic when we apply our transitivity rules. B.1(iii) is a transformation of B.1(i) which infers that *attempt to cause* is transitively applied to all subsequent steps of the goal node to which it is incident. In other words, we infer that an attempt to cause an action to occur is also an attempt to cause the known and intended causal consequences of that action to occur. From here, we see that "The Wily Lion" covers *Deliberate Harm* from the isomorphic relationship to that pattern (as *implies* is an arc that triggers *actualizes* on its destination node). The lion speaks to the bull with an intention to eat the bull, and at a later time, the lion indeed does eat the bull.

There are several other distinguishing features of SIG patterns:

1. SIG patterns do not necessarily include all three layers of the SIG or follow the completeness rules we defined in our schemata. It is implied that an encoding has additional nodes and arcs that satisfy these constraints. For example, *Deliberate Harm* does not include a proposition that actualizes the goal frame, although we stipulated that a goal frame must be actualized before its content is actualized. This is why Figure B.1(iii) omits the timeline proposition that actualizes the lion's goal frame.

2. Similarly, goals in patterns do not need to be explicitly annotated with Affect nodes that specify their affectual impact. If no Affect node is provided in a pattern, then it is assumed that the goal has an impact on an agent, but the identity of that agent is unimportant.

3. Patterns may be compounded into larger structures, with some nodes and arcs forming an intersection common to both patterns. When two patterns are unified in this manner, it is a join. Three or more joined patterns make a chain. For instance, the encoding for "The Wily Lion" presented in Figure 3.18 covers both the *Hidden Agenda* and *Backfire* patterns we describe below, using some of the same P nodes to cover each pattern.

4. As a notational convenience, P nodes in patterns can connect to one another by *followed by*, even though this arc can only connect State nodes. Such a use of *followed by* indicates that the two P nodes are attached to temporally subsequent states.

5. For notational brevity, patterns use an abbreviated notation for certain symbols. A key to the symbols we will use in describing patterns is provided in Table B.1. Even though P and I nodes need not be indexed with propositional encodings of story content, we will refer to them as "propositions" and "interpretative propositions" for convenience.

Symbol	Meaning	Symbol	Meaning
TE	Text node	*ia*	Interpreted as
P	Timeline Proposition	P:X	Timeline Proposition with agent X
G	Goal frame	G:X	Goal frame of agent X
B	Belief frame	B:X	Belief frame of agent X
A	Affect node	A:X	Affectual impact on agent X
I	Interpretative Proposition	I:X	Interpretative Proposition with agent X
I:N	A particular interpretative-layer proposition	I:¬N	The negation of particular interpretative-layer proposition N
act	Actualizing arc (*implies*, *interpreted as*, *actualizes*)	*pc*	Preventing/ceasing arc (*prevents/ceases*)
f	Followed by	*in*	In
p	Provides for	*d*	Damages
wc	Would cause	*wp*	Would prevent
ac	Attempt to cause	*ap*	Attempt to prevent
pf	Precondition for	*pa*	Precondition against

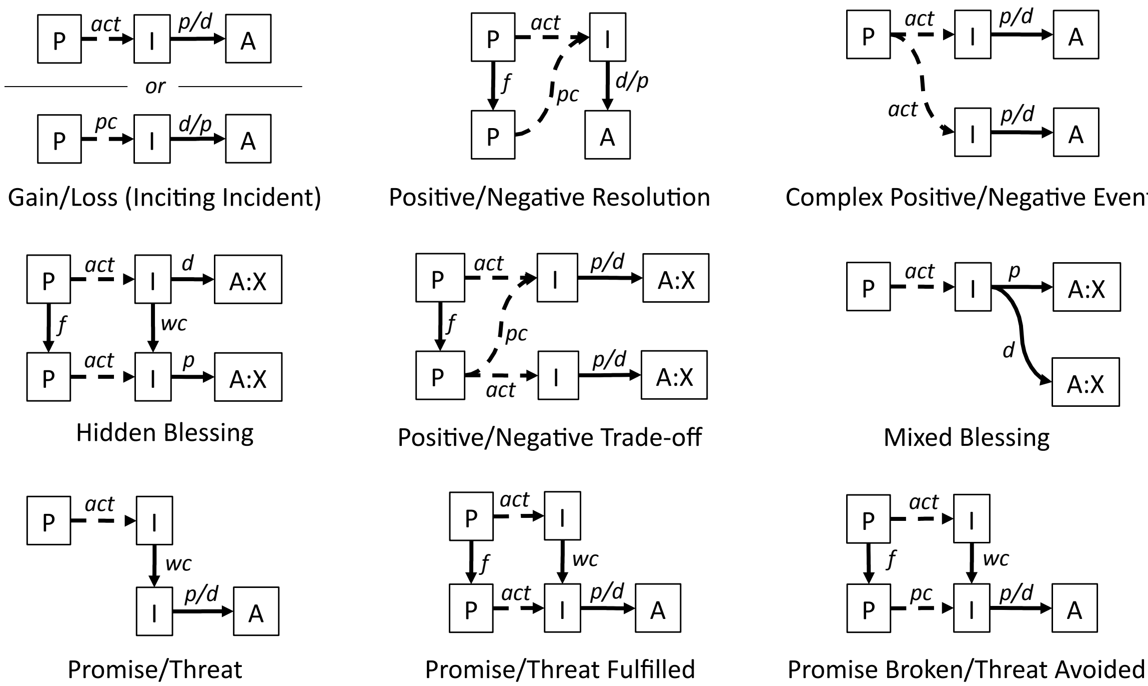
Table B.1: Key to the notation used to describe SIG patterns in this appendix.

Not only do patterns allow us to characterize a story in thematic terms like "hidden agenda," they also allow us to find similarities between stories. When multiple encodings cover the same pattern, we conclude that the stories are analogous to one another. However, not every encoding must cover a pattern from among those we will identify in this appendix; what follows is a top-down, non-exhaustive list. In Chapter 5, we describe an algorithm that finds patterns in a bottom-up manner by comparing encodings to one another directly.

Without further ado, we now describe a set of patterns that illustrate the model's ability to represent narrative scenarios and tropes. The patterns fall into four categories: transitions between affectual states, single-agent goals and plans, multiple-agent goals and plans, and formal storytelling devices as the textual level.

B.1 Affectual Status Transitions

We drew from Section 3.2 that affectual impact is a crucial aspect to a tellable narrative [Stein *et al.*, 2000]. The story's receiver is drawn into the narrative by identifying with one or several agents; when an event happens which impacts an agent's disposition, the effect is a narrative experience for the receiver. Affect nodes are the means by which the



Pattern	Example	Example
Gain/Loss	(Gain) John made a sale.	(Loss) Lou broke his ankle.
Resolution	(Positive) Lou broke his ankle, but it healed.	(Negative) John made a sale, but then the customer backed out.
Complex Event	(Positive) Valerie won a prestigious award with a monetary prize.	(Negative) Adam lost his house and custody of his child in the divorce.
Hidden Blessing	Ben missed his train, but met a girl on the next one.	Sue found a boyfriend, but he stole her money.
Trade-off	(Positive) Joe upgraded his computer to a faster model.	(Negative) Roger declared bankruptcy to get protection from his creditors.
Mixed Blessing	Rita became very wealthy, but lost her friends in the process.	Laura got a job, but the commute was terribly long.
Promise/Threat	The arriving train came around the bend.	Carrie missed her train to the airport.
Promise/Threat Fulfilled	The arriving train come around the bend and approached the station. It stopped as scheduled.	Carrie missed her train to the airport, causing her to miss her flight.
Promise/Threat Avoided	The arriving train come around the bend, but it passed the station without stopping.	Carrie missed her train to the airport, but managed to make her flight anyway.

Figure B.2: Nine patterns for affectual status transitions.

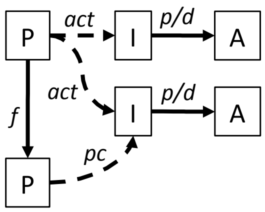
SIG expresses this basic unit of storytelling. We saw in Table 3.6 how *providing for* or *damaging* an Affect node is the mechanism for expressing a positive or negative affectual impact. Figure B.2 applies this mechanism in nine patterns that represent basic affectual status transitions.

As Lehnert's plot unit theory is centered around what we call affect status transitions, these patterns (as others we will describe) cover all of Lehnert's "basic plot units" (Figure 3.4). Further, since both models build complex derivatives by joining and chaining these simple units, the expessive range of the SIG is a superset of that of the plot unit. To show the correspondence, we have adopted some of Lehnert's labels (such as "complex positive") for the nine patterns in Figure B.2.

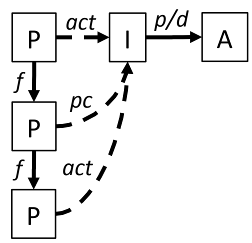
The simplest patterns are *Gain* and *Loss*, in which a single proposition is interpreted to provide for an Affect node (in the case of *Gain*) or damage it (in the case of *Loss*). Alternatively, a timeline proposition can prevent/cease an interpretative proposition that damages an Affect node ("the pain finally ended" is a *Gain*) or provides for one ("his fame faded away" in the case of *Loss*). *Gain* and *Loss* simply restate Table 3.6: A timeline proposition that provides for an Affect node, either through actualization or double-cessation, has a positive affectual impact on the agent; similarly, a proposition that damages one, regardless of the path, has a negative affectual impact. (For purposes of brevity, we do not enumerate all such variations for every pattern in this section that involves an affectual impact.)

The term *inciting incident* comes from popular screenwriting literature. McKee [1997] defines it as the "first major event of the telling" that is the "primary cause for all that follows," including progressive complications, crisis, climax and resolution. It is an incident which prods the hero into action by instilling a "gap" between his current state and his desired state. The archetypal drama is one in which the main character loses something and strives to regain it. A SIG encoding expresses this most basic of storytelling tropes in the damaging of an Affect node. To actualize such damage is to hurt the agent and motivate it to desire to regain (that is, provide for) the damaged Affect node.

In a *Resolution*, the interpretative proposition responsible for inciting change to the Affect node is itself subsequently ceased. The impactful action or stative is removed from the situation model, and the affectual status represented by the Affect node reverts to its



Partial Resolution
(Complex Event + Resolution)



Compounded Transition
(Gain + Loss + Gain…)

Pattern	Example	Example
Partial Resolution	Joe crashed his car into a tree. While his injuries healed, he never got over the loss of his Corvette.	Andy rose to fame and fortune from his real estate business, but later lost the fortune.
Compounded Transition	Bettie lost her dog, but then found him. The dog ran away again the following week!	The conductor's intermittent back problems sent him in and out of the orchestra pit.

Figure B.3: Complex patterns are joins or chains of simpler patterns.

prior state. One example resolution would be the end of a successful business partnership (a negative resolution to a positively impactful stative). Resolution is not compulsory for every Gain or Loss. For instance, death is irreversible in most stories. A story's telling may end with perpetual gain (a happy ending for a particular agent), perpetual loss (a sad ending), or a return to the status quo through *Resolution*.

The next set of patterns demonstrates a few interactions between multiple Affect nodes regarding different agents (or multiple nodes regarding a single agent):

1. A *Complex Event* is one in which a single timeline event has multiple interpretative consequences impacting two or more Affect nodes in a concordant manner (both nodes provided for, or both damaged). Recall that two Affect nodes can refer to the same agent if they belong to distinct types. Winning a prize with a monetary award is a complex positive event, in that it provides for both ego and wealth (using the simple typing we introduced in Section 3.3.2.8). Conversely, being involved in an automobile collision is a complex negative event, damaging at least two Affect nodes—those representing one's health and one's wealth.

   To give an example of how patterns can be joined: If an agent is involved in a car crash, and then heals to full health, the *Resolution* pattern can be applied to the health branch of the complex negative. Assuming the victim was not insured, and the car is unrecoverable, the damage to the victim's wealth is not resolved. Such a join, drawn in Figure B.3, would be a *Partial Resolution* in that only a strict subset of the loss is resolved.

2. In a *Hidden Blessing*, a timeline proposition with a negative impact has a belated, positive impact on another Affect node. An event which at first seems like a total loss is later responsible for a separate gain. The affect polarities can also be reversed, such that an event which at first seems positive later has an unexpected negative consequence.

3. A *Trade-off* is similar to a *Complex Event*, except that one of the interpretative propositions is ceased rather than actualized. That is, one interpretative event or stative is traded off, or replaced by, another; the trade-off is positive or negative depending on the affectual impacts involved. For example, upgrading one's computer to a faster model is an act of substituting one good situation (having this year's model) for another (having last year's model).

4. In a *Mixed Blessing*, a single interpretative event impacts two Affect nodes of the same agent, one positively and one negatively. Where *Trade-off* involves ceasing one event in favor of an affectually similar one, this pattern involves an event which both positively and negatively impacts an agent. A subsequent decision by the agent to reverse the effects of the event would signal a value judgment that the overall impact of the event was negative.

These examples illustrate that small variations in the structure of an encoding can significantly change the thematic nature of the narrative. *Hidden Blessing* and *Mixed Blessing* only differ in terms of temporal structure: In the former, the two Affect nodes are impacted in sequence, where in the latter, they are impacted simultaneously. These tell slightly different stories. The former gives a sense of false resolution to the receiver, where a matter that seems to be resolved for better or for worse is later revealed to have an unforeseen secondary consequence. The gain or loss is thought to be stable, only to later trigger a reversal of fortune. In a *Mixed Blessing*, one event has two immediate, contradictory effects on an agent. Of course, through chaining, a mixed blessing may be altered. Consider the example: "Laura got a job, but the commute was terribly long. So she got permission to telecommute." The negative aspect of the mixed blessing of finding employment is later resolved.

The notion of **expectation**, which we introduced with the *would cause* and *would prevent* arcs in Section 3.3.2.3, enables us to define a pattern for *Promise/Threat* (Figure B.2). An action is actualized that *would cause* a second action which would have a positive or negative affectual impact, respectively. Who is doing the expecting? If the pattern is inside a frame (a Goal or Belief), the frame's agent holds the expectation; if the pattern is in ground truth, the narrator would have the receiver hold the expectation. We saw in the plan for "The Wily Lion" (Figure 3.13) that agents can be mistaken in their beliefs. If the expected event is later actualized, the promise or threat is *Fulfilled*; if the expected event is prevented, the promise or threat is *Avoided* (bottom right of Figure B.2).

These patterns show our basic mechanism for modeling possible futures, which are a distinguishing feature of SIGs compared to previous descriptive models. As receivers, we are led by the storyteller to anticipate or fear certain future events. The clouds darken menacingly, and we expect a harmful storm. We see a daylight from inside a cave, and are given hope that the trapped characters can escape. This approach is scalable, in that a storyteller can give many cues that seem to indicate an event will happen. The longer the event remains hypothetical, the greater the sense of suspense and anticipation. We will explore other ways in which the schemata can express suspense in Section B.6.

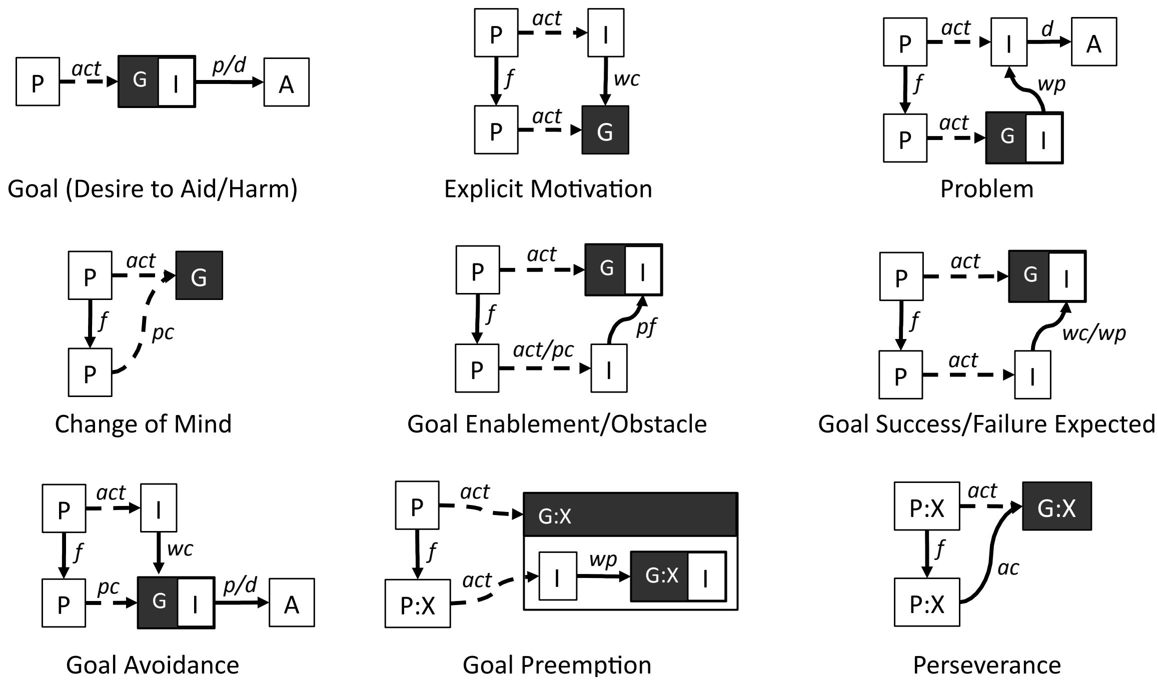
Through chaining, these patterns can express an arbitrarily long narrative arc that swings through multiple successive change of fortune. For instance, joining multiple Gains and Losses gives us a *Compounded Transition* (Figure B.3) in which a goal is obtained, lost, obtained again, and so on. In this case, multiple episodes point to the same goal, where in previous models (such as GRTNs and grammars), we are forced to consider goals as being wholly contained within their episodes. A SIG encoding can describe the global coherence of a lengthy discourse, as well as the local cohesion found between sections or sentences. Episodic structure is representable, but not prescribed.

B.2 Single-Agent Goals, Plans and Attempts

Let us progress to patterns that deal with the formation of goals and plans. We have defined a goal as a hypothetical proposition that a particular agent, or set of agents, desires to be actualized in the story-world (Section 3.3.2). A goal is situated in a "frame" that represents the mental state of the agent as having the desire (Figure 3.10).

The basic SIG pattern for a *Goal* is found in Figure B.4. A timeline proposition P actualizes the goal frame G containing goal content I, indicating that the frame itself is actualized. The agent in question (not specified in the pattern) actualizes a mental state of desiring the actualization of I. The *Goal* pattern says nothing about the outcome of the goal, i.e., whether I comes to pass. It does, though, assign an affective context to I. In a *Desire to Aid*, I would have a positive affectual impact on some agent; in *Desire to Harm*, the goal is to harm some agent's interests. Recall from Section 3.3.2.7 that all content inside a goal frame must "drain" to an Affect node, meaning there must be a path from each goal to an A node that can be traced by following forward arcs. This represents the "stakes" of the goal in an affective context: who, if anyone, would be helped or hurt by a goal's actualization.

Since a goal frame itself refers to a mental state of an agent, it can be treated as a special type of I node. For instance, the *Explicit Motivation* pattern features a causal arc traversing to a goal from an interpretative proposition which was actualized at a previous point in time. An event occurred, and as a consequence, the agent formed a goal. The goal content inside the frame is not specified by or important to the pattern. An example of *Explicit Motivation* would be: "Inspired by a van Gogh painting, Charlie decided to learn to paint." The act of seeing the van Gogh painting did not itself hurt or harm Charlie, but it was the causal antecedent of Charlie's desire to gain the ability to express himself creatively. The *Problem* pattern takes this a step further by explicating that the motivating event is harmful to the agent; therefore, the agent wishes to cease the offending I node and cease the damage. For example: "Charlie felt the rain starting, so he looked for shelter." In this case, the causal action (rain starting) itself causes a stative (Charlie is wet) that has a negative affectual impact on Charlie; this motivates Charlie to take action to try to counteract the effects of the rain. Charlie cannot stop the rain, but there are other actions



Pattern	Example	Example
Goal	(Desire to Aid) Mary dreamed of being a published author.	(Desire to Harm) The lion set out to kill the bull.
Explicit Motivation	Inspired by a van Gogh painting, Charlie decided to learn to paint.	Once Joel got a new bike, his brother had to have one too.
Problem	Rachel's roommate moved out, so Rachel had to look for a replacement.	Lou was desperate to find love after his wife left him.
Change of Mind	Tim suddenly wanted ice cream when he heard the truck go by, but he lost interest just as quickly.	Oscar had a brief interest in learning the violin.
Goal Enablement/ Obstacle	(Enablement) Gary has always wanted a speedboat. Last week he won the lottery!	(Obstacle) Evan's dream of being a fighter pilot was threatened when he failed the exam.
Goal Success/ Failure Expected	(Success) The opera tickets Paul wanted were mailed to him today.	(Failure) Helen slipped and fell as she ran to catch the departing train.
Goal Avoidance	When the bully insulted him, Tom simply ignored it.	Jerry felt the rain starting, but he didn't mind.
Goal Preemption	Frank's lifelong dedication to saving kept him from serious financial problems in retirement.	Jonathan always takes an umbrella so that rain does not threaten his perfect hair.
Perseverance	David courted Carly for years.	Jeff washed dishes at the diner, trying to pay off his bill.

Figure B.4: Patterns regarding the formation of simple single-agent goals and plans.

which might allow him to become dry. Note that in *Explicit Motivation* and other patterns, the desirer and the beneficiary can be different agents. One agent can have a goal to assist another with its problems.

The *Goal Enablement* and *Obstacle* patterns express changes to the actualization status of a goal's precondition. Suppose there is a stative I which must be actualized for a goal to succeed. If I is actualized, the goal is enabled; if I is ceased, I is an obstacle and must be actualized. For instance, the lion's plan in Figure 3.18 involved the successful removal of the obstacle blocking the path to the lion's goal to eat the bull, namely, the bull's dangerous horns. The situation is similar for propositions that are <u>sufficient</u> for a goal's success or failure—those that point to a goal's content with *would cause* or *would prevent* arcs. When the proposition is actualized, we as readers expect that there will be a positive or negative outcome after the normal flow of events. This is modeled with *Goal Success Expected* and *Goal Failure Expected*, patterns that are special cases of *Threat* and *Promise* in which the event that is expected is goal content.

If the goal frame is ceased, this does not necessarily mean that the agent fails to actualize the goal. It only means that the agent stops desiring the goal, as drawn in the *Change of Mind* pattern. A character desires something, and then no longer desires it. The reason for the change of mind, if any, can be attributed to a causal antecedant to the proposition that ceases the goal frame. The *Goal Avoidance* pattern is a special case of the *Promise Broken* pattern in which a goal frame is itself the broken promise: We expect an agent to develop a goal based on a prior event, but the agent does not carry through. For example, we expect the onset of rain to cause Charlie to want to go inside, but contrary to our expectations, he doesn't. This avoidance is itself thematically interesting because it communicates an unusual aspect about Charlie as an agent, namely, that he does not react to stimuli that would seem to damage him according to the affective model that we project onto him. *Goal Avoidance* is a pattern that represents a meaningful *lack* of narrative action.

Goal Preemption shows that the SIG can describe situations of complex goal management that Wilensky, in PAM [Wilensky, 1978b], called "goal subsumption." Specifically, an agent has a goal of *not* developing a goal in the future. We place the undesired goal, both frame and content, on the receiving end of *would prevent* inside a larger goal. The agent has a plan which culminates in the prevention of the unwanted goal. For instance, Frank may develop a long-term savings plan that prevents him from ever having serious financial problems in retirement. When he reaches retirement, having followed his plan for decades, the undesired problem goes from hypothetical to successfully prevented.

Finally, *Perseverance* is a simple pattern that illustrates the repeated use of the *attempt to cause* arc. It represents a sustained attempt, and is featured prominently in "The Wily Lion" during the lion's many attempts to flatter the bull.

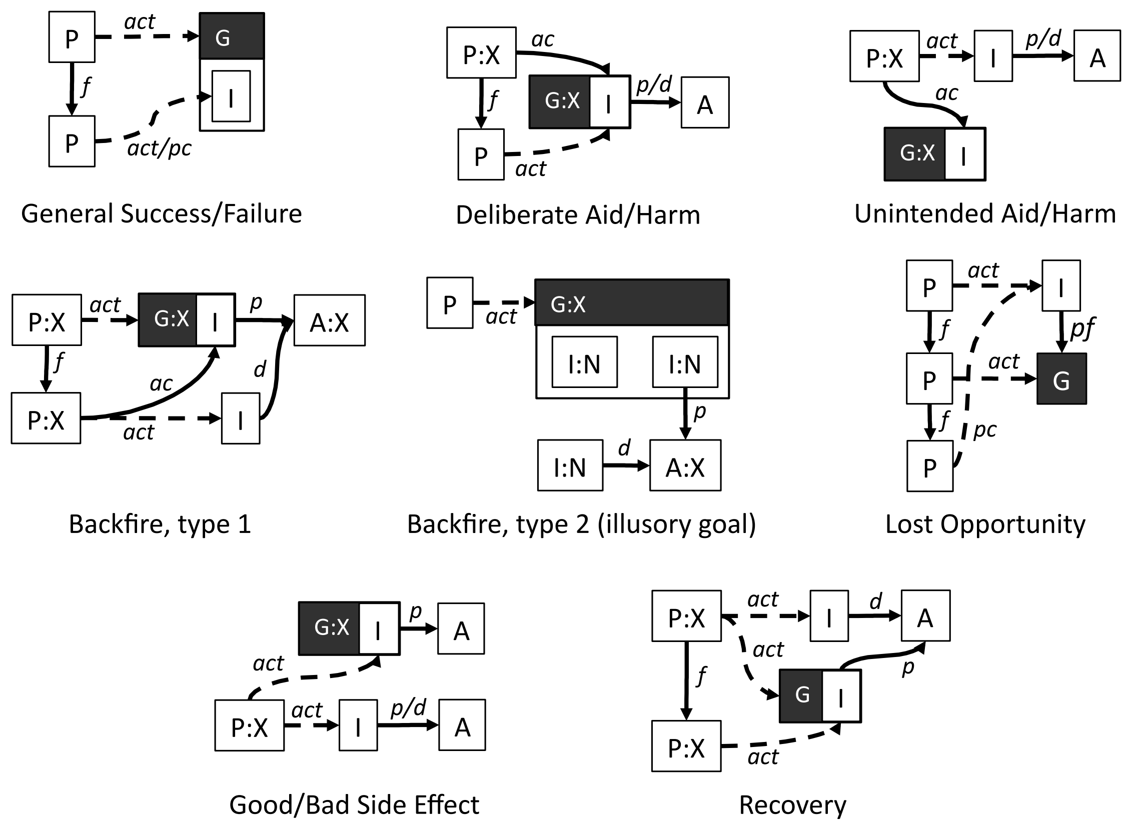
B.3 Single-Agent Goal Outcomes and Beliefs

We now move on to patterns that describe the outcomes of goals—whether the content inside a goal frame is actualized (a successful outcome) or prevented/ceased (a failure). In the simplest of these, *General Success* and *General Failure* as drawn in Figure B.5, the goal frame is actualized, and then the goal content is actualized or prevented (respectively). An agent either succeeds in reaching its goal, or it fails.

One important distinction that the schemata can express is that between deliberate and unintended outcomes. Simply put, if an agent's actions are *attempts to cause* an ultimately successful goal (Section 3.3.2.6), the outcome is deliberate. The agent succeeded in causing what it set out to cause. For example: A mountain climber attempts to reach a summit, and later does so. Figure B.5 illustrates this as *Deliberate Aid* and *Deliberate Harm*, depending on the affectual impact of the event. On the other hand, there are situations in which an agent is attempting to do one thing, but unintentionally accomplishes something quite different. The mountain climber attempts to reach the summit, but breaks his ankle. We represent this by drawing two outgoing arcs from the action:

1. An "attempt" arc (*attempt to cause* or *attempt to prevent*) to indicate the intended result, and
2. An actualization trigger (*interpreted as*, *implies*, *actualizes* or *prevents/ceases*) to indicate the unintended result.

Both results are causal consequences of the action, but only the goal that is linked with an "attempt" arc is considered to be intended. The *Unintended Aid/Harm* pattern



Pattern	Example	Example
General Success/Failure	(Success) Bill finally got the fancy bike he wanted.	(Failure) Travis lost his appeal and went to prison.
Deliberate Aid/Harm	(Aid) Warren succeeded in reaching the summit.	(Harm) Henry swatted and killed the fly.
Unintended Aid/Harm	(Aid) When Doug cut down the diseased tree, he greatly improved his neighbor's view of the lake.	(Harm) Lou's party, while fun, helped spread a nasty flu.
Backfire, type 1	Francis argued for a better grade, but annoyed his teacher into a deduction.	Carl thought that a golf game would help him close the deal, but the client hated his demeanor.
Backfire, type 2	Rick believed that answering the email from the Nigerian executor would make him a millionaire.	Anne thought that taking a knife to the Old Master's painting would raise her profile as a serious artist.
Lost Opportunity	The radio show offered free tickets, but they were gone before Jason could call in.	Scott became wealthy enough on paper to pay off his debts, but his stocks crashed before he sold them.
Side Effect	(Good) When Zoë bought her candy, she reminded the cashier to get his niece a birthday gift.	(Bad) Izzy threw the baseball back over the fence, but it knocked the Jones girl off her trike!
Recovery	Esther got sick, but eventually she healed.	Hillary lost her house, but the insurance company rebuilt it.

Figure B.5: Patterns regarding simple, single-agent goal outcomes.

illustrates an unintended consequence which has an affectual impact, either positive or negative: "Lou's party, while fun, helped spread a nasty flu." While this pattern makes no statement about whether the intended goal was achieved or not, the *Good/Bad Side Effect* pattern involves a situation in which an agent actualizes both an intended goal and a separate, unintended result.

If our goal were a complete semantic understanding of the story-world, we might feel compelled to add a proposition node for every conceivable consequence of every action. For instance, consider the sentence: "The climber broke his arm after falling into a crevice." Technically, the act of falling into a crevice has a untold number of causal consequences, both in the short term and as the climber goes through life: "The climber applied a certain amount of force to try to right himself as he fell," and so on. Naturally, we cannot expect an annotator to have the ability or desire to enumerate every such consequence that can be expressed. This intractable task is known as the *frame problem* in classic artificial intelligence [McCarthy and Hayes, 1969]. This SIG annotation process sidesteps the frame problem by selectively modeling only those consequences which are thematically relevant—those which are either selected for inclusion by the storyteller in the discourse, or have an affectual impact by lying on a path to an Affect node which is itself selected for inclusion. In other words, we rely on the storyteller to indicate which consequences, intended and unintended, are important enough to represent as nodes. The many unintended consequences of the earlier climber's summit attempt are too inconsequential for the telling of this example story, except for the breaking of the ankle. Since the unimportant consequences are not included in the discourse as Text (TE) nodes, they are not represented in the timeline and interpretative layers of the encoding.

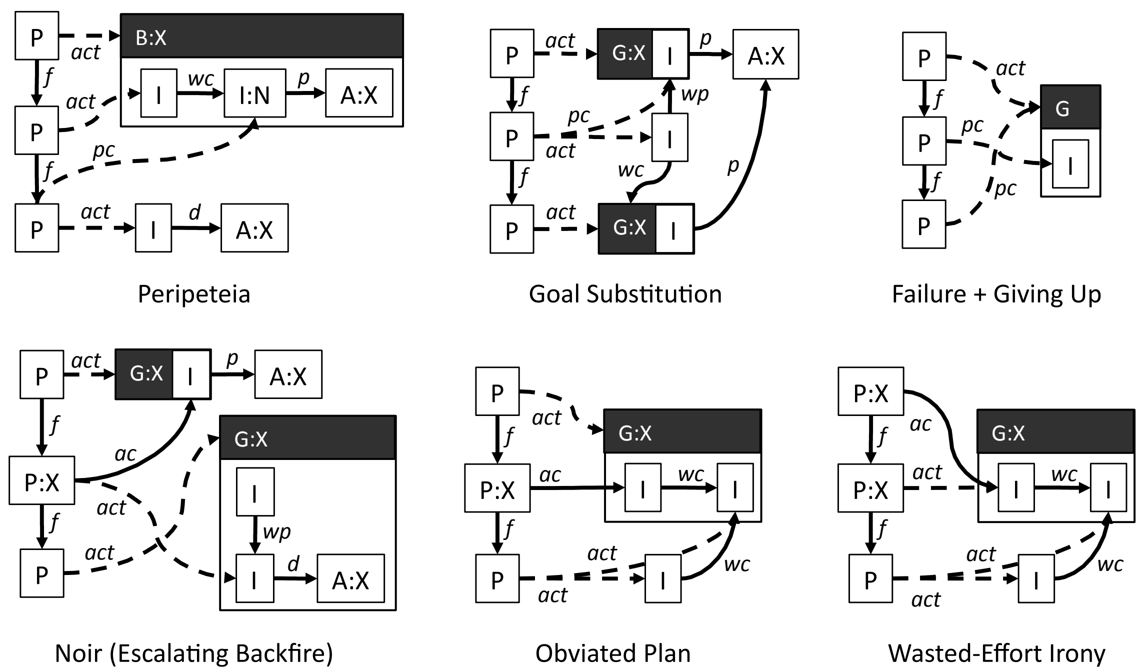
The *Backfire, type 1* pattern, also drawn in Figure B.5, is a special case of unintended harm where the harm is done to the same Affect node that was the intended beneficiary of the action. An agent acts in an attempt to provide for a Affect node; he not only fails, but unintentionally triggers damage to that same node. For example, "Francis argued for a better grade, but annoyed his teacher into a deduction." We also illustrate another type of backfire, which we call the *Illusory Goal*, in which the very premise of a goal is fundamentally flawed. The agent, regardless of the outcome of his goal, does not realize that achieving his goal would hurt his broader cause rather than help it.

The remaining two patterns in Figure B.5 describe other simple goal outcomes. In *Lost Opportunity*, an agent is inspired to develop a goal because of a new enablement that brings him closer to achieving it (a precondition is actualized). But the agent fails to achieve the goal when the precondition is ceased. The window of opportunity is lost. For instance: "The radio show offered free tickets, but they were gone before Jason could call in." The *Recovery* pattern is a fuller expression of the basic dramatic arc as espoused by McKee [1997]: An inciting incident damages an agent and motivates it to act; the agent eventually succeeds in reaching a goal which reverses the affectual impact of the inciting incident. An agent experiences a loss, but then recovers through goal-directed action.

The patterns in Figure B.6 include more complex outcomes:

1. The *Peripeteia* figure is an illustration of Aristotle's term for a sudden reversal of fortune, which we examined in Section 3.3.2.3 in the context of "The Wily Lion". An agent lays out a plan which he believes will provide for him in some way. At first, he succeeds in actualizing the first steps of the plan. But then, a step in the plan is ceased while an unintended consequence damages the agent. Similar to the *Backfire, type 1* pattern, *Peripeteia* involves a tragic (that is, self-damaging) violation of the agent's expectations. The bull believed that removing its horns would cause the lion to perceive him as more handsome, but instead, after removing his horns, he unintentionally enabled the lion to kill him.

2. *Goal Substitution* is the name we give to a dramatic device in which an agent fails in one strategy to achieve a certain affectual impact, but devises a plan to achieve the same impact by another means. One path to an Affect node is blocked, so as a result, the agent substitutes in a second goal that would have a similar type of affective result. Consider the sentence: "Mike never could afford to sail around the world, so instead, he read all the naval travelogues he could." Mike's desire for knowledge and adventure is his ultimate goal; when his initial plan for actualizing his goal is blocked by financial limits, he devises an alternate goal that still provides knowledge and a sense of adventure (albeit a weaker success than he originally envisioned).



Pattern	Example	Example
Peripeteia	The lion did not, in fact, think the bull was more handsome; rather, he attacked and killed the bull.	The teammates thought that they were home free, but their disloyal guide led them into custody.
Goal Substitution	When Valerie found out she couldn't have kids naturally, she decided to adopt.	Mike never could afford to sail around the world, so instead, he read all the naval travelogues he could.
Failure + Giving Up	After the fox failed to reach the grapes on the vine, he decided that they were probably sour anyway.	Ben swore off ever applying to the conference again after it rejected him.
Noir (Escalating Backfire)	Ivan thought it would be a petty theft, but after he accidentally killed the storekeeper, he was close to going to jail for life.	Nancy sped to get to work on time, but she crashed her new car into a neighbor's house.
Obviated Plan	Allie conjured up a fake cough to get out of school, but it turned out to be a snow day anyway.	Agustus ran to make the 8:00 curtain, but found out the show was delayed for half an hour.
Wasted-Effort Irony	Jennifer spent a fortune on a camera the week before everyone at the company got one for free.	Marie painstakingly scaled the fence, only to find an unlocked gate.

Figure B.6: Six patterns for complex single-agent goal outcomes.

3. *Failure + Giving Up* is a join between the simpler patterns *General Failure* and *Change of Mind*. Some goal content is prevented/ceased, and instead of somehow attempting to actualize it anew (which in some cases may be impossible), the agent ends its desire altogether (ceases the goal frame). For instance, "Ben swore off ever applying to the conference again after it rejected him."

4. *Noir* is a pattern that models the cinematic plot device by the same name. A story in the noir genre is typically a crime drama in which a small, selfish act on the part of the main character is exploited by "fate" into an ever-expanding crisis that ultimately results in greater transgressions and total downfall. A character is tempted into an adulterous affair, only to be roped into being the scapegoat for a murder. An everyman comes across a cache of money, and when he tries to keep it, gets involved in schemes larger and more evil than he expected. Noir usually combines the elements of *Backfire* and *Unintended Harm*, such that a "simple plan" goes awry and ends in disaster. As such, our pattern for *Noir* begins with a goal by an agent to help itself. In an attempt to reach that goal, the agent unintentionally triggers an event or stative which has the potential for damage. The agent devises a second goal to "put out the fire," removing the threat, but this attempt fails as well. The process can continue for further iterations, such that each attempt to resolve the crisis instead triggers an even bigger crisis.

5. An *Obviated Plan* is one in which the initial step is unexpectedly proven to be unnecessary. The agent tries to actualize A in order to cause B, but before it can succeed, B is actualized itself without A, due to circumstances that recently developed or were originally unknown to the agent. For instance, "Allie conjured up a fake cough to get out of school, but it turned out to be a snow day anyway." Allie's plan proved unnecessary to reach her goal, unbeknownst to her.

6. *Wasted-Effort Irony* is similar to *Obviated Plan*, except it is made explicit that the initial strategy succeeds before it is obviated. The agent reaches a goal through concerted effort, only to find that the goal could have been reached with less effort through unintended actions or the actions of others. For instance, "Marie painstakingly scaled

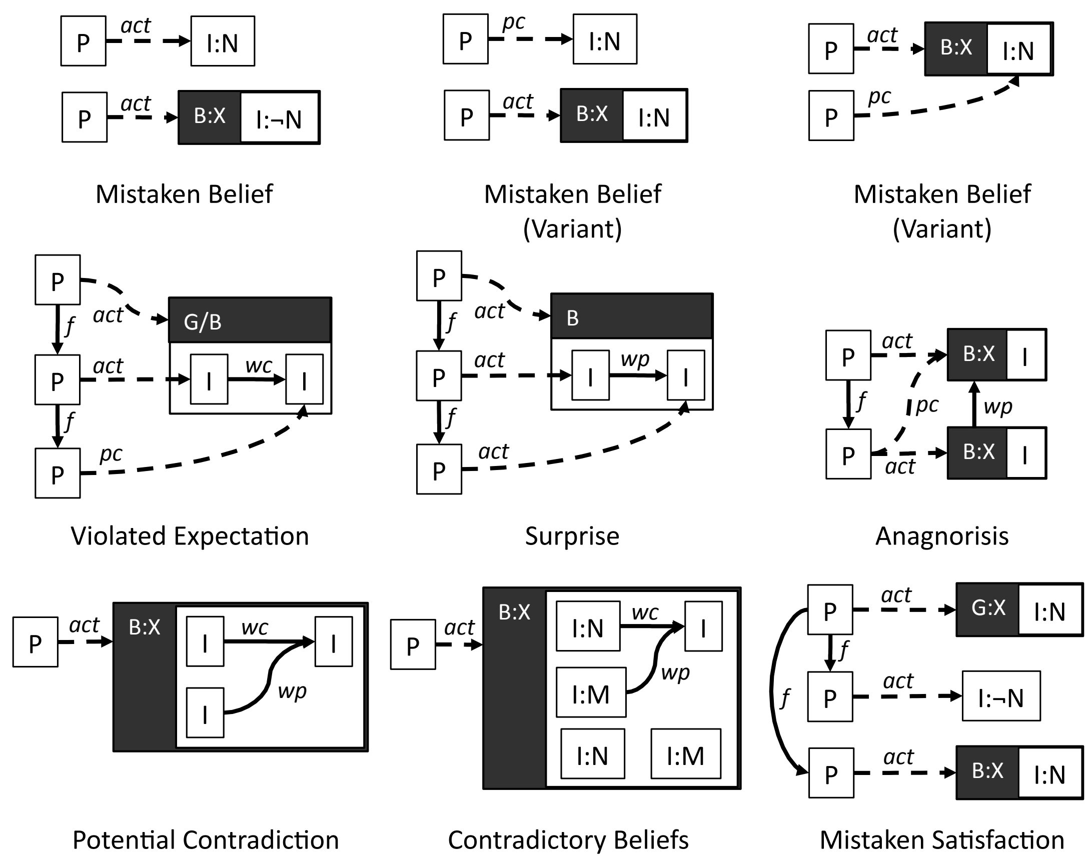
the fence, only to find an unlocked gate." Marie could have reached her ultimate goal (crossing the fence) had she only known about the gate.

B.4 Beliefs, Expectations and Dilemmas

The next set of patterns deals with the beliefs and expectations of agents, independent of their goals. We have declared that in either a goal frame or a belief frame, all arcs are in the belief context of the node from which they originate. For instance, *would cause* from A to B in a goal implies that the agent believes that A would cause B. If A would not, in fact, cause B, the agent is acting under mistaken assumptions, which is itself a thematically rich concept. In the Aesop fable "The Tortoise and the Eagle", the tortoise tries to convince an eagle to teach it to fly, believing that proper instruction is sufficient for him to gain this ability. The eagle reluctantly tries to teach the tortoise by releasing it in mid-air. The expectations of the tortoise are violated, of course, with fatal consequences. Belief and goal frames, by definition, are subjective to the agent and not necessarily consistent with ground truth.

We can draw several patterns that deal with beliefs and expectations such as those found in this fable. The most basic of these is the *Mistaken Belief*, in which we represent the fact that at some point in time, an agent believes a proposition which is known by "ground truth" (the belief context of the story's narrator) to be false. We listed in Section 3.3.2.5 three logically equivalent methods for expressing a false belief; all three are now illustrated in the top row of Figure B.7 as patterns. In the first, some proposition N is actualized, and separately, a belief frame containing a negated instance of the same proposition is actualized. We assert both that N is true, and that an agent believes $\neg$N. (There is no temporal relationship drawn with an *f* arc as the two assertions can be made in any textual or timeline order.) A more economical way to express a mistaken belief is to simply prevent/cease a node inside an actualized belief frame (top right of Figure B.7). The *belief* is asserted to be true, but the *fact* is asserted to be false.

When a belief is about the causal relationship expected between two events, we can express its mistaken nature through hard evidence to the contrary. For instance, as we have



Pattern	Example	Example
Mistaken Belief (any variant)	It was clear out, but Yaël thought it was still raining.	Tom could have sworn the 45-year-old was half her age.
Violated Expectation	The opera tickets Paul wanted were mailed to him last week, but they never arrived.	Steve pulled the trigger, but the gun jammed.
Surprise	Erin didn't expect a clap of thunder so long after the storm ended.	Sandra was startled when the sleeping man suddenly spoke.
Anagnorisis	Oedipus realized that Queen Jocasta was in fact his mother.	Andy later discovered that the class was laughing at him, not with him.
Potential contradiction	Veronica knew her husband could not be in two places at once.	Neil knew that the sun never set at 3 PM.
Contradictory beliefs	Veronica thought she saw her husband at the pastry shop even though she believed he was on a sales call out of town.	Neil was confused when he seemed to see the sun set at 3 PM.
Mistaken satisfaction	Nick thought he had reached LaGuardia airport, but the taxi driver had taken him to JFK.	Fred thought he won the race, but Tom had beaten him by a nose.

Figure B.7: Patterns regarding single-agent beliefs and expectations.

mentioned, an agent's plan may be predicated on the assumption that A will cause or allow for B. (The tortoise's plan incorporates an assumption that tutelage is sufficient for him to be able to fly.) When A is actualized but B is prevented/ceased, the assumption is proven false, as drawn in the *Violated Expectation* pattern in Figure B.7. The inverse situation, in which an action occurs even though an agent believes that the current circumstances would prevent it from occurring, is drawn as a *Surprise* pattern.

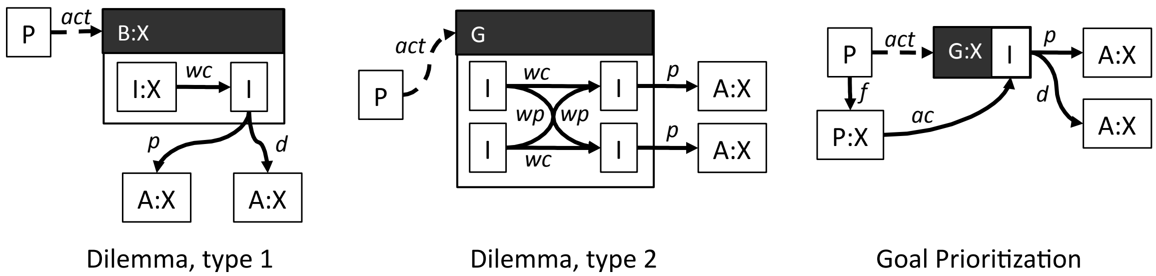
When events such as these cause one belief to supersede another, the agent has a revelation or sense of discovery. One belief frame is actualized as another is ceased. We call this pattern *Anagnorisis*, after Aristotle's term for a turning point in a work in which a character makes a critical discovery (often one that leads to tragic results). Anagnorisis frequently occurs alongside peripeteia. In a canonical example, Oedipus kills his father and marries his mother without realizing it (a mistaken belief). The climax of the play occurs at the moment of anagnorisis, when Oedipus realizes the truth of his identity and of the true nature of his past behavior.

Another thematic idea that this formalism can express is the contradiction of two beliefs. In a *Potential Contradiction*, shown on the bottom left of Figure B.7, an agent believes that two events would have the opposite effect on a hypothetical proposition (one would cause it, the other would prevent it). The agent does not assert belief in any of the three propositions, just that two of them are mutually exclusive due to their opposing causal effects on the third. If both of the two triggering events are actualized, the result is the *Contradictory Beliefs* pattern. The agent believes two mutually exclusive facts to be true at once. At least one of the beliefs in the frame is mistaken: Either one of the facts is not really true, or the facts are not truly mutually exclusive, or both. For instance, an agent who lives at a low latitude can believe that the sun cannot set at 3PM. If he sees a sunset at that time, either the sun is not truly setting, it is not truly that time, or the facts are not truly incompatible. Once the agent visits a very high latitude, he will see evidence that will cause him to no longer believe that the two facts are mutually exclusive (one can say that he will experience anagnorisis).

One final mistaken belief to consider is the belief that a goal has been satisfied when, in truth, it has not. We saw in Figure B.5 that success in general can be indicated by actualizing the content inside a goal frame. The same effect can be achieved by actualizing any node that contains the same proposition, even if it is not strictly inside the goal frame. This allows us to express situations in which the actualization status of a goal is itself a mistaken belief. The bottom right of Figure B.7 shows a pattern called *Mistaken Satisfaction* in which a proposition N is desired by agent X. Agent X then has a mistaken belief that N is actualized, when in fact it is ceased. This is a highly dramatic situation that cannot be expressed by the other descriptive formalisms we have considered in detail. Depending on how the story is told (that is, the *telling time* of the three actions), the receiver either knows that the agent has had an illusory victory, or can be later surprised along with the agent to discover that the success was in fact a failure (another episode of anagnorisis). One example is a cinematic trope, often found in horror films, that covers both *Mistaken Satisfaction* and *Surprise*. The hero strikes at the villain with what he believes is a fatal blow, but the villain is in fact still alive and able to terrorize anew. Sometimes the filmmaker reveals the villain's survival to the audience in advance; on other occasions, the audience only discovers it at the moment when the hero does the same.

Goals, like beliefs, can be contradictory (that is, mutually exclusive). An agent can believe that two propositions cannot both be true, and yet still desire them both to be true. This is the essence of the **dilemma**, in which the agent must choose between two potential paths. Dilemmas have been a key part of storytelling since its inception—an agent must often choose between duty and love, or between its own welfare and that of an ally. Achilles chooses to die a young hero rather than live longer but grow old; Faust chooses to give up his soul so that he can achieve unlimited knowledge; countless young heroines from 18th and 19th century literature have had to choose between marrying out of love or in accordance with their parents' wishes.

We identify two patterns that describe a dilemma, and one that describes acting on a dilemma (making a choice). They are shown in Figure B.8. The *Dilemma, type 1* pattern simply describes a belief that the same event would impact two of the agent's own Affect nodes. This presents a significant use for the typing of Affect nodes (Section 3.3.2.8). If the same action may provide for one essential need, but harm another, the agent faces a dilemma as to which need is "more important." The young heroine understands that



Pattern	Example	Example
Dilemma, type 1	Chuck felt pressured by his friends to take up smoking, but he knew it would hurt him.	Harris considered whether it was worth it to go for a run in the rain.
Dilemma, type 2	Colin knew that going to his sister's recital would make him miss the big game.	Marissa wanted to be both a full-time chef and a full-time mom.
Goal Prioritization	Colin decided to go to his sister's recital.	Marissa decided that her career was worth putting off a family.

Figure B.8: Patterns that demonstrate dilemmas.

if she marries out of love, she provides for her own happiness but damages her relations with her family. Achilles understands that to fight is to choose to damage his own life but provide for his honor and the welfare of his community. While using Affect nodes with such a simple typing scheme is a reductionist approach to modeling literary dilemmas, it is also domain-independent, and more expressive in absolute terms than a simple positive or negative (+/−) affect representation. The second pattern, *Dilemma, type 2*, models the choice between two goals that, though both desirable, are mutually exclusive. Both events are unequivocally good for one Affect node or another, but each would preclude the other from occurring (or so the agent believes). The agent must choose to pursue one or the other. A young man may wish to both travel the world and start his career, but he cannot do both.

The moment in which an agent chooses one alternative in a dilemma is the one in which it makes an attempt (either an *attempt to cause* or an *attempt to prevent*, depending on how the dilemma is structured). We call this the moment of *Goal Prioritization*. As drawn in Figure B.8, it describes the situation in which the agent attempts to cause the hypothetical event it considered in the *Dilemma, type 1* pattern. Note that the outcome of this decision is not a part of the pattern; we may join this pattern with any of the outcome patterns we have considered. For instance, the agent may unexpectedly find that the dilemma was a false choice, so that all the consequences of its action are positive or neutral: "Colin decided to go to his sister's recital rather than the big game, but the game was rained out anyway" (a mistaken dissatisfaction).

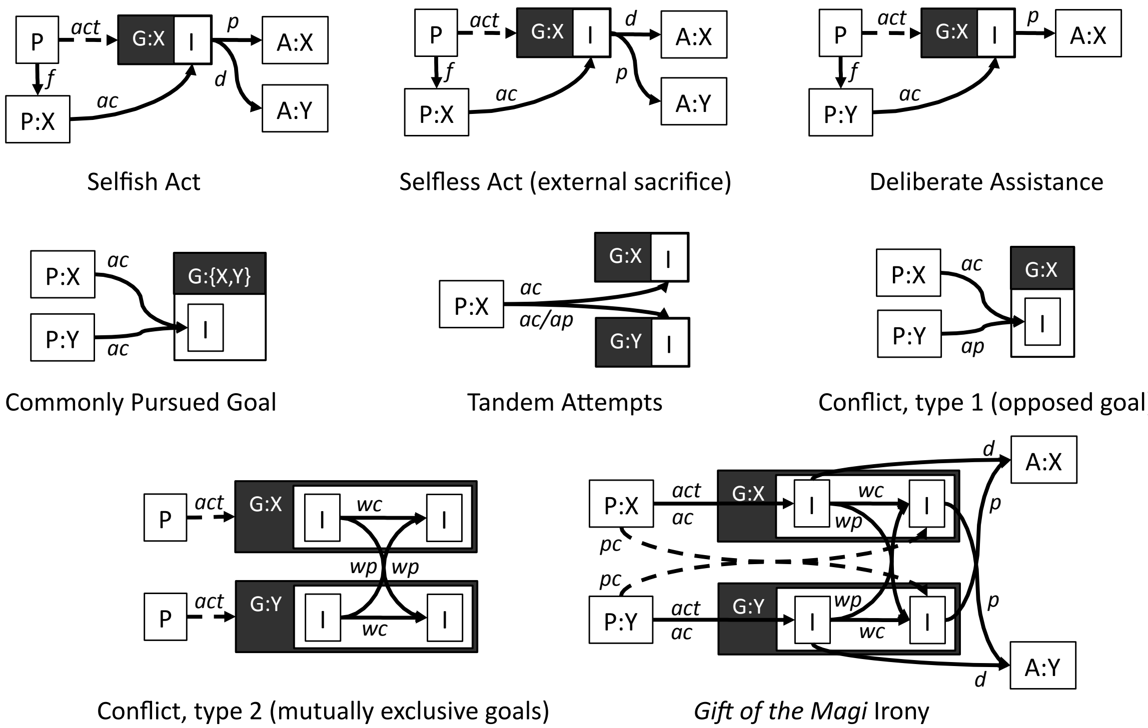
B.5 Multiple-Agent Interactions

We have thus far enumerated a set of patterns that represent the dynamics of single-character goals, plans, attempts, outcomes and beliefs. We now consider the interactions that occur between agents. Thematically interesting stories, including most fables and literature, describe an intricate interplay of intentions and desires between two or more distinct agents. For instance, the theory of mind (Section 3.2.2), which influenced the design of the SIG, is most often applied to scenarios with multiple agents rather than single ones. A plan such as the wily lion's may not only depend on having another agent act, but having another agent develop its own plan, with its own estimation of still others' beliefs and goals.

Figure B.9 shows patterns that depict basic two-agent interactions. The most fundamental interaction is that between one agent and another agent's goals and well-being. In *Selfish Act*, an agent X attempts to help himself at the expense of another agent Y. Conversely, a *Selfless Act* is one in which a character intentionally acts toward a goal that would help another but hurt himself. This is the essence of self-sacrifice: acting with the intention of assisting others, despite the damage the action causes to one's self. For instance, a girl may donate her only bicycle to the school's toy drive, to help children needier than herself.

Through chaining, we can describe more complex patterns built upon selfless and selfish actions. Combining *Selfish Act* with *Backfire*, for instance, depicts a situation in which an agent attempts to profit from others' misfortunes, but in the end, is defeated and punished. Such is the essence of the villain's journey in no small number of morality tales and fables.

If an agent acts with intention to help another, but there is no appreciable self-sacrifice, the result is *Deliberate Assistance*. As drawn in Figure B.9, Agent Y attempts to actualize Agent X's goal (e.g., a local giving directions to a tourist). But cross-agent attempts can be more complicated than simple assistance. A single action can represent two or more



Pattern	Example	Example
Selfish Act	Zach refused to give the old lady his seat on the bus.	Vicky sent copies of her sister's diary to Vicky's friends.
Selfless Act (external sacrifice)	Nate ran into traffic to try to fetch the stranger's wayward dog.	Barbara donated her bike to the school toy drive.
Deliberate Assistance	James gave the tourist directions to Times Square.	Mark bought his daughter the doll he knew she wanted.
Commonly Pursued Goal	Audrey and Aaron went to buy a new television.	The outfielder threw the ball to the second baseman, who tagged the runner out.
Tandem Attempts	Nick helped get the damaged car off the road so he and everyone else could proceed.	Alan sought a way to prove he was right without making his opponent lose face.
Conflict, type 1 (opposed goal)	Luís tried to ask Lupe out, but her sister blocked his message.	The candidate tried to dispel the image purported by his opponent that he was weak on defense.
Conflict, type 2 (mutually exclusive goals)	The two male bears fought each other for access to the female.	Alex and Tom were in a race to the patent office.
Gift of the Magi irony	Della sold her hair to buy a chain for Jim's watch, but in the meantime, Jim sold his watch to buy Della a set of beautiful combs.	

Figure B.9: Patterns that describe two-agent interactions.

intentions by having multiple *attempt to cause* and *attempt to prevent* arcs originate from its node. Thus, there is a many-to-many relationship between *who* is attempting and *what* is being attempted. The next two patterns are examples of this effect. In *Commonly Pursued Goal*, two or more agents each attempt to fulfill the same goal (as when a couple goes shopping for a new appliance together). Conversely, in *Tandem Attempts*, the same agent can perform an action that is a dual attempt to actualize two or more distinct goals ("killing two birds with one stone").

Of course, in many instances one agent's goal is at odds with another's. Interpersonal conflict—a bedrock of dramatic storytelling—can be expressed in several ways. Two conflict patterns are shown in Figure B.9:

1. In *Opposed Goal*, a single goal is the subject of opposing intentional actions. One agent strives to cause a hypothetical action while the other agent strives to prevent it. The affectual impact of the goal determines the stakes—which agent, if any, would be harmed or helped by the objectionable action. For instance, if a jealous ex-boyfriend attempts to stop the wedding of his ex-girlfriend to another man, there is a single hypothetical goal (to get married) whose actualization is pushed in different directions by two agents.

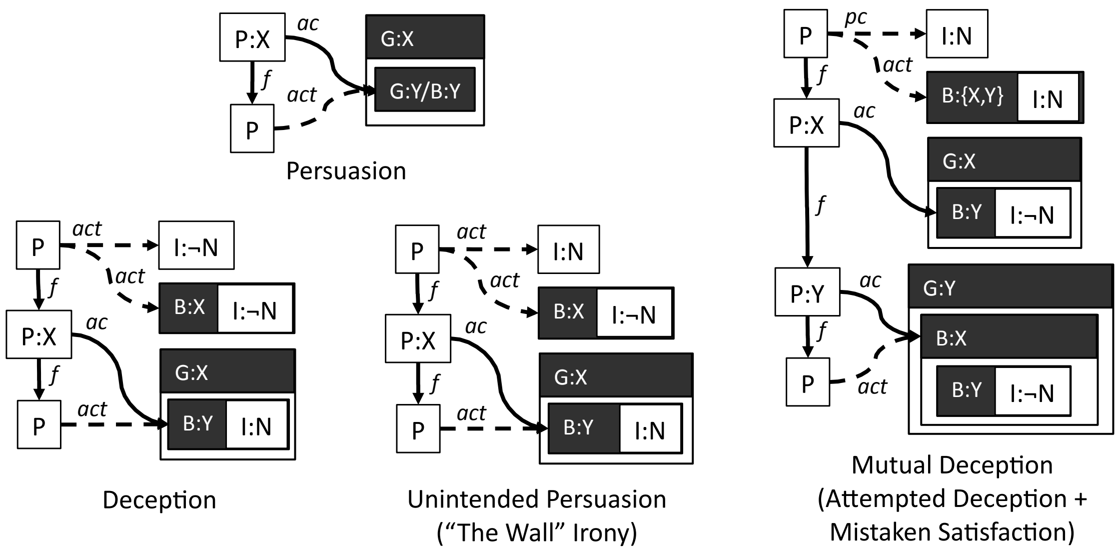
2. In *Mutually Exclusive Goals*, two agents pursue goals that cannot both be actualized at once (or so the agents believe). The success of each goal implies that the other goal has failed. Again, the affectual impact of each goal describes the stakes; in one common scenario, the two agents are competing for a limited resource: a mate (in the case of love triangles and multiple suitors), a natural resource (land, clean water, oil, fuel), and so on. In another common scenario, the fight, two agents become engaged in a conflict in which only one can achieve a mutually agreed upon status of victor. In a race, for instance, each agent attempts to win the race, a hypothetical action which entails that the other agent did not win the race.

Sometimes the interactions of desire, attempt, intention and outcome can become tangled in knots that form the bases of famous stories. In O. Henry's short story "The Gift of the Magi", Della and Jim are a young married couple too poor to buy one another gifts for Christmas. Della decides to cut off her beautiful hair and sell it to a wigmaker so that she can afford to buy a chain for Jim's pocket watch, which is itself a prized possession. (That is, Della intentionally damages her own interests in an attempt to provide for Jim's.) Unfortunately, she then finds out that Jim has sold his pocket watch in order to buy her a set of combs for her hair. (Jim had a symmetrical plan to make a sacrifice of his own to provide for his young wife.) The situational irony is that each agent unintentionally subverted a plan that would have positively impacted it—and in the process, made a sacrifice that turned out to be futile. The diagram for this scenario, at the bottom right of Figure B.9, combines the patterns for *Selfless Act*, *Unintentional Harm* and *Failure* (although one can easily add to the pattern to express the underlying message of the story, that each goal frame itself demonstrates the agent's love for its spouse—an ultimate emotional victory). The representation shows the literal symmetry of the situation, with the cross purposes seen as crossing lines.

B.5.1 Persuasion and Deception

The theory-of-mind aspect of our schemata is most evident in the patterns that deal with persuasion and deception. In these common narrative situations, one agent attempts to trigger a particular goal or belief in another agent. We show this by nesting one agency frame (a goal or belief) inside another frame: It is one agent's goal <u>content</u> to actualize another agent's goal <u>frame</u>. We first encountered this dynamic earlier, in Figure 3.10, as we introduced the concept of nested agency frames. We similarly define persuasion here as the act of causing another agent to believe in the truth of a proposition, or to cause another agent to desire a particular goal. The pattern for *Persuasion*, shown in Figure B.10, simply places a goal or belief frame of agent Y inside the goal frame of agent X; Y's belief is the subject on an intentional action (attempt), and is subsequently actualized. X has persuaded Y to take on a certain goal or belief.

If the proposition in question is known by the persuader to be false, the situation crosses into one of deception. From a philosophical standpoint, though, the exact definition of deception is a matter of debate. The most commonly accepted definition can be paraphrased formally as follows:



Pattern	Example	Example
Persuasion	Pete asked his neighbor if she'd help run his yard sale, and she agreed.	Debbie convinced her teacher that she was too sick to take the test.
Deception	Kris fooled everyone into thinking he had gone abroad for his birthday.	Paul gave a check to the jeweler that he knew would bounce.
Unintended Persuasion	Pablo gave his enemies what he thought was false information about his ally's whereabouts—which turned out to be true!	
Mutual Deception	Nathan lied to his wife that he was at work late, to keep his visit to the jewelry store a secret. Nancy had found out that he was at the store, but she pretended to believe him.	The attackers attempted to trick the defenders into drawing forces to the south rim, so the wily defenders attempted to trick the attackers into believing the ruse had worked.

Figure B.10: Patterns that describe interactions regarding persuasion and deception.

> To lie is to make an assertion that is believed to be false to some audience with the intention to deceive the audience about the content of that assertion. [Mahon, 2008; Williams, 2002]

In other words, the agent must know that the statement is false, but speak with intention to persuade another agent that the statement is true. Because this definition is based on an interplay of belief and intention, concepts which we represent, we can adapt it directly into a *Deception* pattern (Figure B.10). The only difference between this definition and the *Deception* pattern is that in the pattern, <u>any</u> action that is done with the intention to persuade the receiver—not just speaking, but communicating nonverbally, or even withholding information when it is called for—is considered an act of deception. This fulfills one philosophical objection to the above definition, that the statement condition omits non-verbal modes of deception [Mahon, 2008]. Other alternative definitions of deception vary the other requirements. One finds it sufficient for the deceiver to merely not believe the assertion is true, as opposed to positively believing that the assertion is false [Carson, 2006]. This variant, too, can be encoded by negating the belief frame in Figure B.10 (that is, asserting ¬B:X(I:N) rather than B:X(I:¬N)). In short, we make no claim as to the optimal logical definition of lying and deceptive behavior, but we do claim that the SIG formalism has the expressive power to generate several of the variants that philosophers have proposed.

To take one further example, consider the epistemic puzzle in Sartre's short story "The Wall". The main character is Pablo Ibbieta, a prisoner in the Spanish Civil War (1936-1939). Pablo's captors offer to spare his life if he reveals the location of an accomplice, Ramón Gris. Pablo tells them that Ramón is in the cemetery, but he knows Ramón to be hiding near the city. He expects to be shot once the guards discover his deception, but is left alive. Wondering why, he hears from a fellow prisoner that Ramón had left his safe house after an argument and gone to hide in the cemetery; the guards found him and shot him on the spot. Did Pablo lie? In this case, the statement was *true*, even though the agent *thought* it was false. In intending to convince his audience of a false statement, he unknowingly convinced them of a true statement in a manner that caused his entire plan to backfire. This particular example of situational irony, which we call *Unintended Persuasion*, is shown as a pattern in Figure B.10.

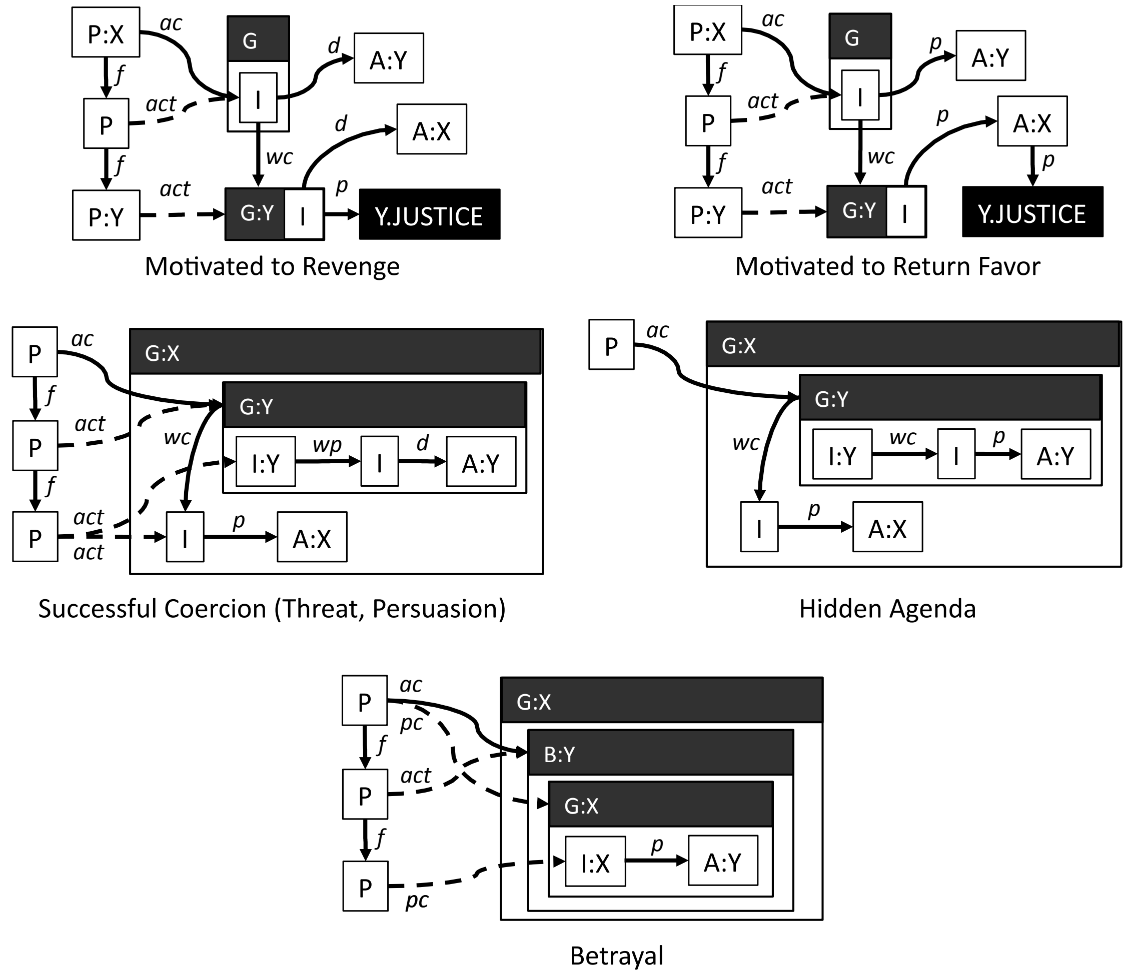
The final pattern in this figure is for *Mutual Deception*, a situation which shows how the concepts of *Deception*, *Failure* and *Mistaken Satisfaction* can be chained. It describes the scenario where one agent attempts to convince another of a false statement. Not only does the receiver understand the deception, it also turns the tables by falsely pretending to have fallen for the ruse. In other words, X tries and fails to deceive Y into believing N; Y tries and succeeds in deceiving X into believing that X succeeded in deceiving Y into believing N. For example: "Nathan lied to his wife that he was at work late, to keep his visit to the jewelry store a secret. Nancy had found out that he was at the store, but she pretended to believe him." Such a scenario shows how chaining multiple patterns leads to a narrative whole that is greater than the sum of its parts.

B.5.2 Complex Two-Agent Interactions

There are many other complex two-agent interactions we can enumerate that occur repeatedly in narratives across genres. Most are permutations of the basic pieces we have already visited. Figure B.11 illustrates five such patterns, all dealing with the complex interminglings of plan, counterplan and outcome.

Motivated to Revenge is a pattern that describes a revenge story in the most abstract of terms. In our definition, a revenge situation is one in which agent X acts with deliberate intent to harm agent Y, and succeeds; subsequently, Y is motivated to harm X, an act motivated by her sense of personal justice (a particularly typed Affect node). For example, a robbery motivates the victim to seek justice against the perpetrator, in the form of an harmful action (perhaps imprisonment by way of the criminal justice system, perhaps by more direct means). In the converse situation, *Motivated to Return Favor*, "one good turn deserves another." A successful, intentional action to help another is motivation for the receiver to help the giver.

Successful Coercion is a join of the *Threat* and *Persuasion* patterns. A coercing agent, X, persuades a victim, Y, that doing some action would prevent harm from coming to Y. That action, directly or indirectly, helps X. In other words, X convinces Y that Y is threatened and uses this threat to motivate Y into acting in such a way that it otherwise would not. For example, "Lisa blackmailed Robert into doing her homework so she wouldn't tell his



Pattern	Example	Example
Motivated to Revenge	When Moe hit Gina with a snowball, she became determined to somehow make him cry.	Lindsay vowed that she would find and sue the man who broke into her apartment and robbed her.
Motivated to Return Favor	Amy baked Mark a cake because he had helped her with her computer.	Jerome wanted to somehow find and thank the stranger who donated him a kidney.
Successful Coercion (Threat, Persuasion)	The kidnapper made the family pay thousands to avoid seeing their son hurt.	Lisa blackmailed Robert into doing her homework so she wouldn't tell his parents where he went on Saturday.
Hidden Agenda	The fox challenged the crow to demonstrate her singing ability, so that she would drop a piece cheese that the fox desired.	Dolores, hoping to win the bake sale, sabotaged her friend's entry by suggesting she use castor oil.
Betrayal	Mark made Andy believe he was on Andy's team, only to defect to Andy's opponents.	Lucy told Charlie she would hold the football, only to pull it away as Charlie tried to kick it.

Figure B.11: Five patterns for complex two-agent interactions.

parents where he went on Saturday." The *Hidden Agenda* pattern is quite similar, except that the potential impact on Y is positive. X instills a goal in Y to do something which X promises will help Y; the same event (or the intention itself) would actually help X, though X does not reveal this to Y. "The Wily Lion" was a clear example of a hidden agenda. Recall that the same action, the bull removing its horns, appeared twice in our encoding (Figure 3.18). Within the bull's plan, which was instilled by the lion, the removal of the horns was the first step on a causal path toward improving bull's handsomeness. The lion did not inform the bull of its belief that the same action was part of a separate causal chain designed to lead to the lion killing the bull. The dual instances of the same proposition allow the encoding to express such "two-faced" behavior.

Finally, let us consider a three-degree "stack" of agency frames in the form of a *Betrayal* pattern, seen at the bottom of Figure B.11. In our definition, betrayal is a deliberate and successful misrepresentation of one's intention, followed by action which is contrary to that intention. An offending agent, X, convinces a victim, Y, that X has a certain goal or desire. Once the deception succeeds, X belies the supposed goal or desire, actualizing the betrayal. The Trojan horse of lore is such a situation: The Greeks attempt to convince the Trojans that the Greeks have decided to sail away and abandon the siege of Troy. Only after the Trojans fall for the ruse and accept the horse as a prize do the Greeks reveal that they have not, in fact, sailed away in defeat. (This particular example also employs the *Hidden Agenda* pattern, in that the same action—the acceptance of the horse by Troy—appears in two distinct plans, one for each army.) The level of nesting can get far deeper than this in narratives about complex multi-agent plans. In *Götterdämmerung*, the fourth and final opera in Wagner's *Ring* cycle, the ultimate villain Alberich sets a plan in motion to reclaim a magical ring that he had forged to rule the world. The ring had subsequently been stolen by his arch-nemesis, the god Wotan, eventually passing to the heroine Brünnhilde as a wedding ring from the fearless hero Siegfried (a descendant of Wotan). Alberich conscripts his son Hagen to the task of manipulating the local human monarch, Gunther, into drugging Siegfried so that Siegfried is amenable to the suggestion that he forcibly betray his own wife Brünnhilde and relieve her of the ring that (in a sense) is controlling Alberich through the intoxicating effects of the power that the ring bestows on

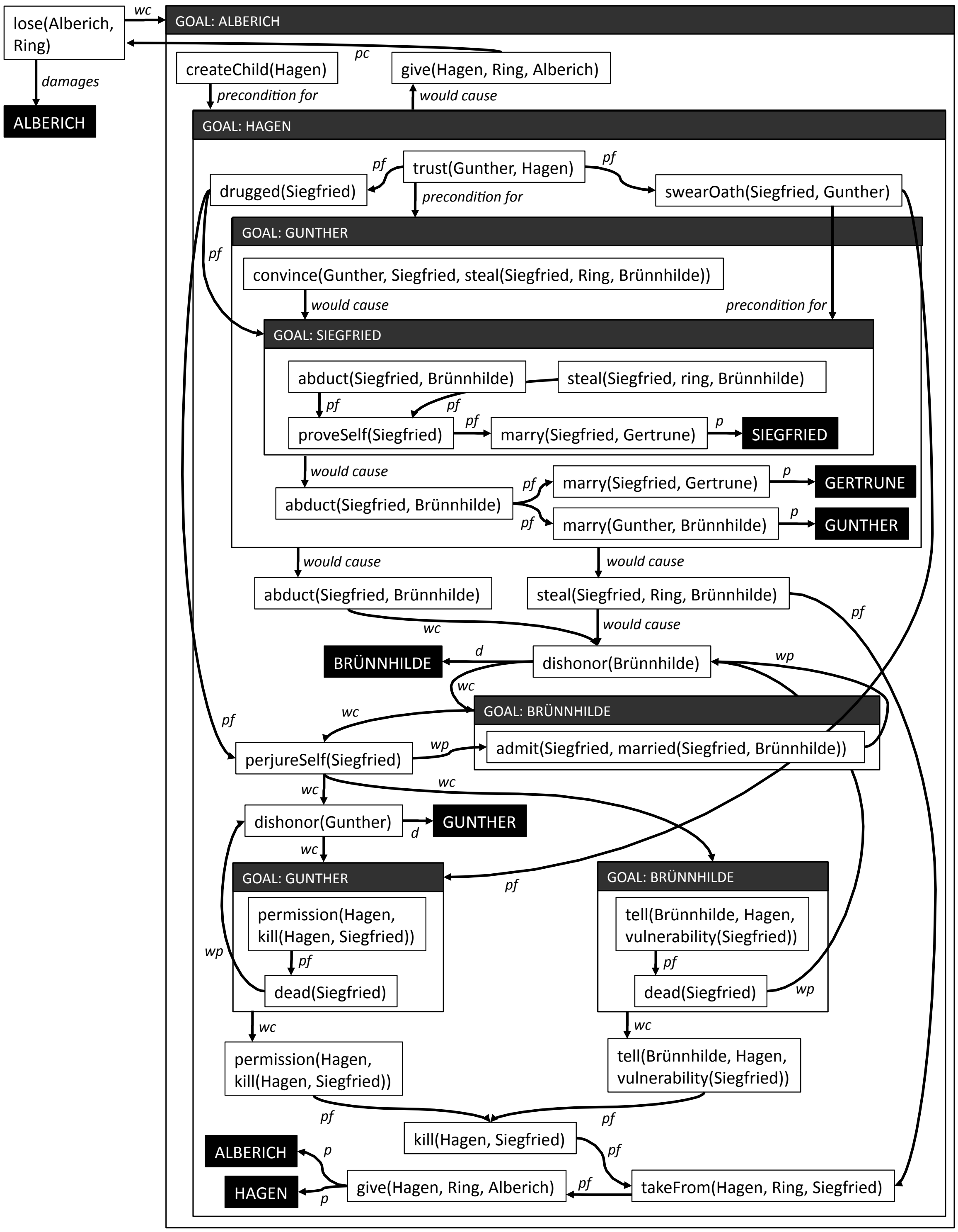


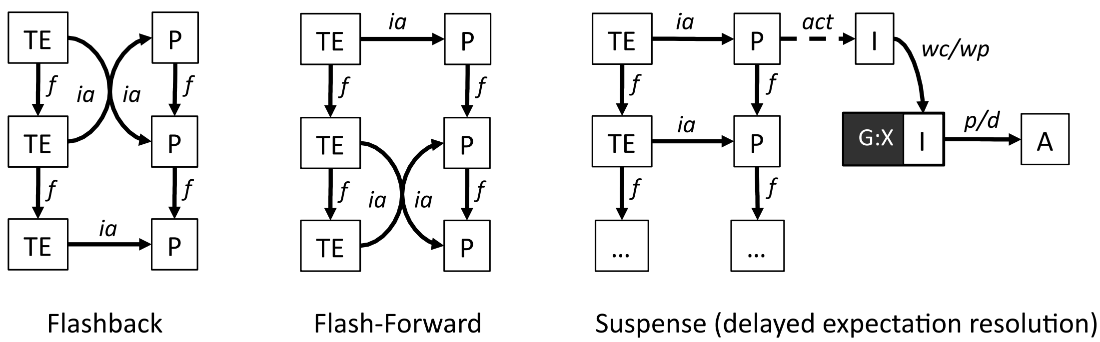
Figure B.12: An encoding of Alberich's highly manipulative plan in *Götterdämmerung*.

its former owners. Hagen successfully tricks Gunther into planning to manipulate Siegfried into abducting Brünnhilde by promising that Gunther could marry her; in fact, Hagen only wants Brünnhilde abducted so he can reclaim the ring on her finger. Siegfried, in turn, is persuaded with the promise that he would marry Gunther's daughter Gertrune (as the drug has caused him to forget that he is already married). Through a combination of hidden agendas, deceptions, manipulations of others' will, and physical force, Alberich nearly succeeds in executing a plan that requires many additional self-serving plans to be formed and executed. Our schemata formally represents complex interpersonal relationships such as these (Figure B.12).

B.6 Textual Devices

For the latter section of this appendix, we turn our attention away from the story itself and toward the telling of the story. In the language of the Russian formalists, we move from the *fabula* to the *sjužhet*. In terms of SIG structure, where all of the patterns we have seen concern the timeline and interpretative layers alone, the following patterns involve the textual layer. As introduced in Section 3.3.1, this layer contains Text (TE) nodes that correspond to snippets of surface discourse. This is the only layer that encodes the story as a sequence of utterances in a surface medium, from the beginning of the discourse to the end. In contrast, the timeline and interpretative layers are annotated with a full retrospective understanding of the entire text.

This distinction allows us to represent certain strategies employed by what Bal [1997] called the "narrating agent" in charge of the telling of the story, in particular the selection and ordering of story content into a narrated discourse. The need for content selection is clear: The essence of storytelling is deciding what to convey about a story-world and what to leave out as unimportant or implied. For instance, it would be inappropriate for the narrating agent to reveal the identity of a killer at the beginning of a crime story, when the crime first occurs. The information exists in the story-world but must be withheld at least until the detective completes his or her investigative process, in order for the receiver to experience a sense of ambiguous causal closure. Stories are received structures, and the



Pattern	Example	Example
Flashback	He came to America at the age of nine; he was born in Hungary. I met him in college.	Gary and Jamie are talking to a cake decorator. They got engaged just two weeks ago!
Flash-Forward	Yaël won at the gymnastics meet in 2008. She then broke her ankle and quit—but not before winning a second meet the next year.	The new Navy class showed promise early on. They graduated on Sunday. One Cadet dropped out after failing an exam.
Suspense	The bomb was set to go off at midday. The tower clock was just striking noon. An eerie quiet had settled over the normally busy street. A dog barked in the distance.	Joe had waited months to hear if she had intentions to move back East. He opened the envelope and read the note from Lisa. His eyes ravenously consumed the handwritten script. He put the note back on the table and felt drawn to the window. The neighbors' kids were splashing around in a yellow plastic pool.

Figure B.13: Patterns regarding textual devices that involve the manipulation of time.

temporal telling of the discourse represents the "valve" through which information flows from the storyteller's model to the receiver's model.

We addressed the question of content ordering in Section 3.3.1, and in particular in Figure 3.8. These plots illustrate the capacity of the narrating agent to manage the flow of time and introduce temporal disfluencies (flashbacks and flash-forwards). Let us now describe patterns that more formally demonstrate how these effects be represented. Figure B.13 shows patterns for *Flashback*, *Flash-Forward* and *Suspense*. In the first two cases, a time disfluency appears as a crossing of *interpreted as* arcs between the "telling time" vector of TE nodes and the "story time" vector of P nodes in the timeline layer. In a flashback, a text node connects to a timeline node attached to a state preceding the state associated with the prior text node. A flash-forward is similarly structured, except that the text node reaches farther ahead in timeline time than would be expected by the text nodes before and after in the discourse.

Suspense is a powerful storytelling device that evinces a heightened interest in the story's telling. In a suspenseful telling, a significant open question is not yet answered. As receivers, we want to know: What will be the outcome of the agent's goal? Will the factor threatening to cause a positive or negative affectual impact come to pass? As long as questions are unanswered, we anticipate an outcome of some kind. A pattern for the strategic withholding of outcome is shown on the right of Figure B.13. *Suspense* is structured as a delay in the resolution of an expected event. Recall from Section 3.3.2.3 that an event is "expected" to be actualized if a preceding event in a causal chain is actualized. If A would cause B, and A occurs, then an agent expects B. The receiver, identifying with that agent, may expect B as well. Does B happen? Note that the second TE node in the sequence is interpreted as a timeline proposition which has no bearing on the expectation at hand. The answer to the question is not yet given. As long as the answer is postponed in the flow of the discourse, the reader (assisted by Barthes's "hermeneutic code") may feel a sense of suspense, especially if the expected event is important to a plan or an affectual impact (that is, if the stakes are non-trivial). The nodes with ellipses indicate that the suspense can be prolonged indefinitely. In some stories, suspense is never resolved.

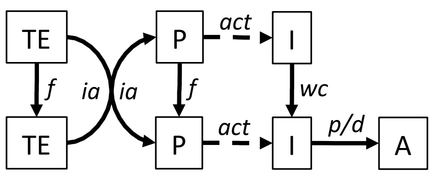
B.6.1 Mystery

The generation of a sense of mystery is also a potent storytelling device. In a mystery, certain information is withheld from the discourse, but that information is key to completing the receiver's understanding of the story. For instance, we are told in Forster's citation (Section 3.1) that "the queen died." This is an event without an explicit cause. As receivers, we yearn for causal closure, such that each event has a rationale and has been communicated by the storyteller to serve a clear pragmatic purpose. Literary critics have written about this yearning; in particular, Shklovsky [1990] found that mystery is a primary driver in Dickens novels. At the beginning of *Little Dorrit*, we are told that Arthur Clennam's father gave Arthur a watch with a mysterious message; the meaning of the message is withheld from us, as is the reason for Arthur's mother's strange behavior upon being told about the message. Both factors imply agentive goals without stating them. The full truth about Arthur's heritage, which belatedly provides causal closure by revealing the underlying goals and plans motivating these actions, is not revealed until the last act of the novel.

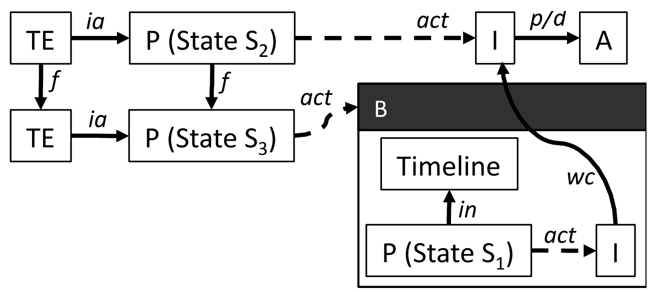
We identify three specific types of mystery, and depict each as a pattern in Figure B.14. The three are:

1. *Ambiguous causal antecedent, revealed in narration.* An event is described that has an affectual impact on an agent, but is not a part of a plan or any known causal chain. The reason for the event's occurrence is a mystery. Later in the discourse, a subsequent text node describes an event that occurred prior to the mysterious event, and served as its causal antecedent. (Perhaps the mysterious event was part of an intentional plan, or perhaps it was an unintended occurrence triggered by a tangentially related event. The reader can generate a set of possible interpretations—tentative additions to her situation model—that provide such closure, but will not know for sure until the true cause is later revealed.) For instance, we may modify Forster's citation to read: "The queen died. She had been ill for years."

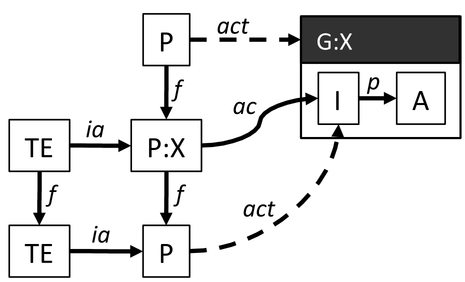
2. *Ambiguous causal antecedent, revealed in revelation.* Similar to the first type, mystery here stems from the description in the discourse of an event that has an impact but does not have a known cause. The difference lies in the manner of the mystery's reveal. Where in the first part, the narrating agent used a flashback to give a definitive answer to the mystery, in this pattern an agent comes to a *belief* about the answer. The belief frame contains an alternate timeline (Section 3.3.1) which draws from the main timeline and contains an event fixed to a time state prior to the one where the mysterious event occurred. The agent then expresses a belief in a causal relationship between the alternate-timeline event and the mysterious event. Naturally, since this is only a belief frame, the pattern makes no definitive claim that the agent is right or wrong in its interpretation of story-world history. The revelations of *Little Dorrit* are an example, as are crime stories in which the detective arrives at a hypothetical timeline representing the version of events she has deduced. The timeline provides causal closure to the agent believing in its veracity.



Mystery, type 1 (ambiguous causal antecedent, revealed in narration)



Mystery, type 2 (ambiguous causal antecedent, revealed with agent's revelation)



Mystery, type 3 (plan enacted without an explication of its frame)

Pattern	Example	Example
Mystery, type 1	The queen died. She had been ill for years.	Patrick stepped out of the plane. It was his first tandem skydive.
Mystery, type 2	Emily came to believe that the hamster escaped because her brother had left the cage open.	Ian realized that his fear of water stemmed from a childhood trauma he had forgotten.
Mystery, type 3	Jeremy inspected the outside of his mailbox every night for months. The neighbors eventually decided he was crazy.	Gertrude bought ten packs of toothpicks a day for years. At the age of eighty she revealed a six-foot model of the Eiffel Tower.

Figure B.14: Patterns regarding the creation of mystery.

3. *Plan enacted without an explication of its frame.* This third type of mystery involves an event for which, again, no known causal antecedent can be interpreted from the discourse itself. In this case, though, there is no explicit reveal of the answer to the particular mystery of why the agent takes a particular action: The event is part of a plan whose frame is actualized in a timeline event that is suppressed from the receiver. In other words, an agent acts with intention to reach a goal, but the receiver has not been told that there is a goal at stake, let alone what the goal is or how the agent developed its plan. Instead, the reader is left to infer the goal frame as the most likely interpretation based on other story events. (Because the actualization of a goal frame is necessary for the pursuit of the content found within the frame, this type of mystery can be seen as a special case of the first type.) For instance, in "The Wily Lion" the reader is not briefed on the lion's elaborate plan before the lion starts flattering the bull. Instead, the nature of the plan remains mysterious until the end of the story, when the lion kills and eats the bull for reasons that have been given (namely, that the bull was very hungry). The plan becomes apparent in retrospect. If the narrator had stopped to identify the plan in full, the suspense of the story would not have originated from the mystery of "What is the lion's strategy?", but instead from the more prosaic "Will the lion succeed?".

In sum, the creation of mystery is a device to increase the "tellability" of the story by prompting the receiver to have its own goal—to achieve causal closure in its cognitive model of the story's meaning. As receivers, we are given pieces of a large puzzle, one at a time and in a non-random order, and we must find the most likely (or most satisfying) assemblage of all the facts into a coherent whole.

B.6.2 Selective Inclusion and Point of View

The power that the narrative agent derives from selecting and ordering events in the story-world goes far beyond the ability to craft surprise and suspense. The careful omission of information can also alter the sympathies of the receiver. While we do not address reader affect and morality in detail in this thesis, let us note a pattern in which the moral orientation of the story can hinge on the inclusion or omission of key facts. Figure B.15(i)

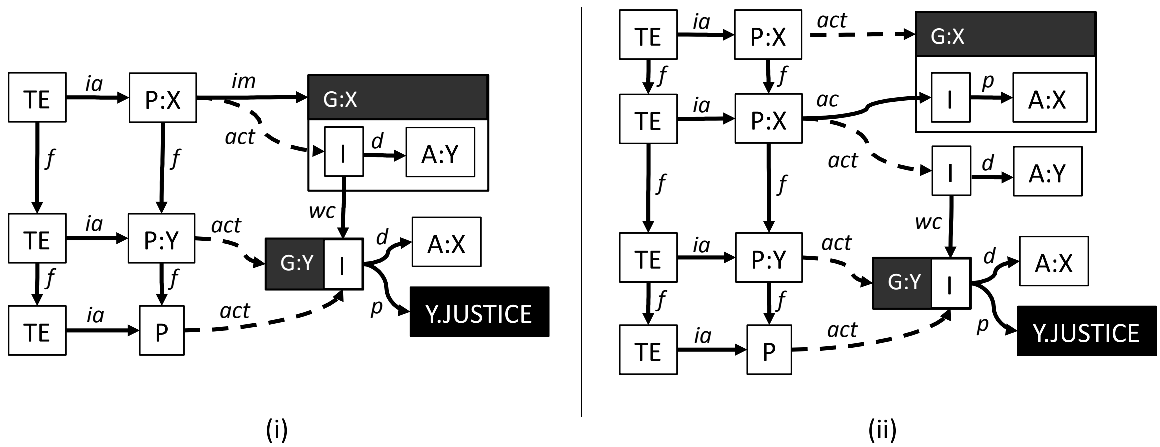


Figure B.15: Interpretative content is influenced by the selective inclusion of *fabula* information in a manner representing point of view.

Pattern	Example	Example
(i)	When Moe hit Gina with a snowball, she became determined to somehow make him cry. The next day at school, she told everyone about how she had seen Moe desperately call for his mother after slipping on some ice.	Lindsay vowed that she would find and sue the man who broke into her apartment and robbed her. It took a year, but she eventually had the teen sent to juvenile detention.
(ii)	Moe was trying to clear snow from his front yard when he accidentally threw snow in Gina's face. Gina was enraged and became determined to somehow make him cry. The next day at school, she told everyone about how she had seen Moe desperately call for his mother after slipping on some ice.	13-year-old Tim got a home run, but accidentally hit the baseball over the fence and through Lindsay's nearby window. Not finding her at home, he opened the window by reaching through the broken pane. After retrieving the ball, he used some first aid supplies to clean up the cuts on his hand. Lindsay, returning home, vowed that she would find and sue the man who broke into her apartment and robbed her. It took a year, but she eventually had the teen sent to juvenile detention.

shows a pattern similar to *Motivated to Revenge*, which we explored previously: Agent X acts in such a way to harm Agent Y, and because of this, Agent Y acts to harm Agent X out of a sense of personal justice. What this pattern does not include is the causal antecedent for Agent X's actions. Agent X's transgression is seen by the receiver as the "first mover"—X's action disrupts the initial equilibrium of the story-world by triggering damage to an Affect node; Y's action against X would not have happened unless X had acted against Y. For lack of any contrary information, the receiver assumes that X's action was deliberate, inferring an *implies* arc to the goal frame. (Strictly speaking, the motivation is mysterious.) The receiver sees X as a provoker, and Y as a victim.

B.15(ii) shows the same set of story events with a new prologue. A new timeline proposition, situated at the first state in the timeline, triggers an interpretation that X's actions were an unintentional side effect of a different plan (one that did not involve Y at all). With this new information, both X and Y are cast as victims of circumstance. Perhaps X could have been more careful to not hurt Y, but the moral absolutism of B.15(i) is replaced by a more nuanced response. X did not mean to hurt Y, but he did hurt Y, and so Y's revenge is less palatable.

These examples highlight the primacy of the receiver's interpretative process. As its name implies, the interpretative layer is a representation of a subjective impression of the causal, motivational and strategic structure of the story based on the incomplete set of cues present in the discourse. While in Chapter 5 we elicit encodings from trained annotators, from a formal standpoint we leave open the question of *how* a receiver arrives at a SIG representation from a text. Even small changes in the discourse may lead to major changes in the moral or causal shape of a story, and while the our schemata can formally depict alternative or plural readings, it does not dictate any particular inferential process.

Appendix C

SIG Closure Rules and Pattern Definitions

In Appendix B we described a series of SIG patterns (compounded relations) that represent narrative scenarios and tropes such as success in reaching a goal, motivation to return a favor, and the hidden agenda. We used these patterns in Chapter 5 to find similarities and analogies within the DramaBank corpus.

In order to identify patterns when they occur in an encoding, we must first perform a "closure" routine to find transitive and inferential relationships. An action that prevents a negative affectual impact, for instance, has an indirectly positive affectual impact. What follows is a precise description of the closure rules we use to analyze encodings, as well as formal descriptions of the patterns we introduced in Appendix B. The definitions are in Prolog format.

C.1 Closure Rules

```
% FollowedByTransitive allows for transitive temporal relationships.
followedByTransitive(A,B) :-
        followedBy(A,B).


followedByTransitive(A,B) :-
        setof((A,B), (
        followedBy(A,C),      % "consume" a base hop
        followedByTransitive(C,B)), Result),
```

```
        member((A,B),Result).


followedByOrSimultaneous(A,B) :-
        followedByTransitive(A,B).

followedByOrSimultaneous(A,B) :-
        A=B.


wouldCauseTransitive(A,B,_) :-
        wouldCause(A,B).

wouldCauseTransitive(A,B,_) :-
        providesFor(A,B).

wouldCauseTransitive(A,B,IMPLIED) :-
        wouldCause(A,IMPLIED),
        wouldCauseTransitive(IMPLIED,B,_).

wouldCauseTransitive(A,B,IMPLIED) :-
        wouldPrevent(A,IMPLIED),
        wouldPreventTransitive(IMPLIED,B,_).

wouldCauseTransitive(A,B) :-
        setof((A,B), (
        wouldCauseTransitive(A,B,_)), Answers),
        member((A,B), Answers).


% Flatten the various arcs for positive and
% negative actualization status transitions.
actualizesFlat(A,B) :-
        actualizes(A,B).

actualizesFlat(A,B) :-
        interpretedAs(A,B).

actualizesFlat(A,B) :-
        implies(A,B).

ceasesFlat(A,B) :-
        ceases(A,B).
```

```
% Causes a beneficial factor.
actualizesTransitive(A,B,_) :-
        actualizesFlat(A,B).

actualizesTransitive(A,B,IMPLIED) :-
        actualizesFlat(A,IMPLIED),
        providesFor(IMPLIED,B).

% Stops a damaging factor.
actualizesTransitive(A,B,IMPLIED) :-
        ceasesFlat(A,IMPLIED),
        damages(IMPLIED,B).

actualizesTransitive(A,B,IMPLIED) :-
        actualizesFlat(A, IMPLIED),
        equivalentOf(IMPLIED, B),
        A\==B.

actualizesTransitive(A,B,IMPLIED) :-
        ceasesFlat(A, IMPLIED),
        inverseOf(IMPLIED, B).


% Excludes frames themselves from actualization arcs;
% content is what matters.
actualizesTransitiveContent(A,B) :-
        actualizesTransitive(A,B,_),
        \+ outermostFrame(B).
actualizesTransitive(A,B) :-
        setof((A,B), (
        actualizesTransitive(A,B,_)), Answers),
        member((A,B), Answers).


% Ceases a damaging factor.
ceasesTransitive(A,B,_) :-
        ceasesFlat(A,B).

ceasesTransitive(A,B,IMPLIED) :-
        actualizesFlat(A,IMPLIED),
        damages(IMPLIED,B).

% Ceases a beneficial factor.
ceasesTransitive(A,B,IMPLIED) :-
        ceasesFlat(A,IMPLIED),
        providesFor(IMPLIED,B).
```

```
ceasesTransitive(A,B,IMPLIED) :-
        ceasesFlat(A, IMPLIED),
        equivalentOf(IMPLIED, B).

ceasesTransitive(A,B,IMPLIED) :-
        actualizesFlat(A, IMPLIED),
        inverseOf(IMPLIED, B).

ceasesTransitiveContent(A,B) :-
        ceasesTransitive(A,B,_),
        \+ outermostFrame(B).
ceasesTransitive(A,B) :-
        ceasesTransitive(A,B,_).


% These are used to go "through" frames to content.

% An agent declares an intention to achieve some goal content by
% actualizing its goal frame.
declaresIntention(ACTION, GOAL, AGENT) :-
        setof((ACTION,GOAL,AGENT), (
        actualizesTransitive(ACTION,GOALBOX,_),
        interpNodeIn(GOAL,GOALBOX),
        goalBox(GOALBOX),
        agent(GOALBOX, AGENT)), Result),
        member((ACTION, GOAL, AGENT), Result).

% An agent declares an intention to achieve some goal content
% by attempting to cause it.
declaresIntention(ACTION, GOAL, AGENT) :-
        setof((ACTION,GOAL,AGENT), (
        attemptToCauseTransitive(ACTION,GOAL,_,_),
        agent(ACTION, AGENT)), Result),
        member((ACTION, GOAL, AGENT), Result).


% An agent declares belief in some content.
declaresBelief(ACTION, CONTENT, AGENT) :-
        actualizesTransitive(ACTION, BELIEFBOX,_),
        interpNodeIn(CONTENT, BELIEFBOX),
        beliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT).

declaresBelief(ACTION, CONTENT, AGENT, BELIEFBOX) :-
        actualizesTransitive(ACTION, BELIEFBOX,_),
        interpNodeIn(CONTENT, BELIEFBOX),
        beliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT).
```

```
% An agent ceases its intention to achieve some goal content by
% ceasing its goal frame.
ceasesIntention(ACTION,GOAL) :-
        ceasesTransitive(ACTION, GOALBOX,_),
        interpNodeIn(GOAL, GOALBOX),
        goalBox(GOALBOX).

% An agent ceases its intention to achieve some goal content
% by attempting to prevent it.
ceasesIntention(ACTION,GOAL) :-
        attemptToPreventTransitive(ACTION,GOAL,_,_).

% An agent ceases belief in some content.
ceasesBelief(ACTION, CONTENT, AGENT, BELIEFBOX) :-
        ceasesTransitive(ACTION, BELIEFBOX,_),
        interpNodeIn(CONTENT, BELIEFBOX),
        beliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT).

preconditionForTransitive(A,B,_) :-
        providesFor(A,B).
preconditionForTransitive(A,B,_) :-
        preconditionForFlat(A,B,_).

preconditionForTransitive(A,B,IMPLIED) :-
        preconditionForFlat(A,B,IMPLIED).
preconditionForTransitive(A,B,IMPLIED) :-
        preconditionForFlat(A,IMPLIED,_),
        preconditionForTransitive(IMPLIED,B,_).

preconditionForTransitive(A,B,IMPLIED) :-
        wouldCauseTransitive(A,IMPLIED,_),
        preconditionForTransitive(IMPLIED,B,_).
preconditionForTransitive(A,B,IMPLIED) :-
        preconditionForFlat(A,IMPLIED,_),
        wouldCauseTransitive(IMPLIED,B,_).

% If X would prevent Y which is preventing Z, X is a precondition for Z
preconditionForTransitive(A,B,IMPLIED) :-
        wouldPreventTransitive(A,IMPLIED,_),
        wouldPreventTransitive(IMPLIED,B,_).

% If X is an agency box surrounding a node which is a precondition for Y,
% X is itself a precondition for Y
preconditionForTransitive(A,B,IMPLIED) :-
        interpNodeInTransitive(IMPLIED,A,_),
        preconditionForTransitive(IMPLIED,B,_).
```

```
preconditionForTransitive(A,B) :-
        setof((A,B), (
        preconditionForTransitive(A,B,_)), Answers),
        member((A,B), Answers).

preconditionAgainstTransitive(A,B,IMPLIED) :-
        wouldPreventTransitive(A,B,IMPLIED).

preconditionAgainstTransitive(A,B,IMPLIED) :-
        wouldPreventTransitive(A,IMPLIED,_),
        preconditionForTransitive(IMPLIED,B,_).


wouldPreventTransitive(A,B,_) :-
        wouldPrevent(A,B).

wouldPreventTransitive(A,B,_) :-
        damages(A,B).

% Preventing an X that would cause a Y would effectively prevent Y.
wouldPreventTransitive(A,B,IMPLIED) :-
        wouldPrevent(A,IMPLIED),
        wouldCauseTransitive(IMPLIED,B,_).

% Causing an X that would prevent a Y would effectively prevent Y.
wouldPreventTransitive(A,B,IMPLIED) :-
        wouldCauseTransitive(A,IMPLIED,_),
        wouldPreventTransitive(IMPLIED,B,_).

wouldPreventTransitive(A,B) :-
        wouldPreventTransitive(A,B,_).


attemptToCauseTransitive(ACTION,GOAL,_,_) :-
        attemptToCause(ACTION,GOAL).

% If an agent does something that they believe would eventually lead
% to a goal, they are attempting to achieve that goal.
attemptToCauseTransitive(ACTION,GOAL,IMPLIED_SUBGOAL,IMPLIED_PRECONDITION) :-
        attemptToCause(ACTION,IMPLIED_SUBGOAL),
        agent(ACTION,AGENT),
        wouldCauseTransitive(IMPLIED_SUBGOAL,GOAL,IMPLIED_PRECONDITION),
        partOfAgentBelief(GOAL,AGENT).

attemptToCauseTransitive(ACTION,GOAL,IMPLIED_SUBGOAL,IMPLIED_PRECONDITION) :-
        attemptToCause(ACTION,IMPLIED_SUBGOAL),
        agent(ACTION,AGENT),
        preconditionForTransitive(IMPLIED_SUBGOAL,GOAL,IMPLIED_PRECONDITION),
        partOfAgentBelief(GOAL,AGENT).
```

```
attemptToCauseTransitive(ACTION, GOAL) :-
        setof((ACTION, GOAL), (
        attemptToCauseTransitive(ACTION, GOAL,_,_)), Result),
        member((ACTION, GOAL), Result).


attemptToPreventTransitive(ACTION,GOAL,_,_) :-
        attemptToPrevent(ACTION,GOAL).

attemptToPreventTransitive(ACTION,GOAL,IMPLIED_SUBGOAL,IMPLIED_PRECONDITION) :-
        attemptToPrevent(ACTION,IMPLIED_SUBGOAL),
        agent(ACTION,AGENT),
        preconditionForTransitive(IMPLIED_SUBGOAL,GOAL,IMPLIED_PRECONDITION),
        groundTruthOrOfAgent(GOAL,AGENT).

attemptToPreventTransitive(ACTION,GOAL,IMPLIED_SUBGOAL,IMPLIED_PRECONDITION) :-
        attemptToCause(ACTION,IMPLIED_SUBGOAL),
        agent(ACTION,AGENT),
        preconditionAgainstTransitive(IMPLIED_SUBGOAL,GOAL,IMPLIED_PRECONDITION),
        groundTruthOrOfAgent(GOAL,AGENT).

attemptToPreventTransitive(A,B) :-
        attemptToPreventTransitive(A,B,_,_).


% preconditionForFlat integrates goals into their host agency boxes.
preconditionForFlat(X,Y,_) :-
        preconditionFor(X,Y).
preconditionForFlat(X,Y,_) :-
        wouldCause(X,Y).
preconditionForFlat(X,Y,IMPLIED_GOALBOX) :-
        interpNodeIn(Y,IMPLIED_GOALBOX),
        preconditionFor(X,IMPLIED_GOALBOX).


% A node exists in any agency box by some agent (goal or belief).
partOfAgentBelief(GOAL,AGENT) :-
        interpNodeInTransitive(GOAL, SUPERGOAL,_),
        agent(SUPERGOAL, AGENT).

interpNodeInTransitive(X,Y,_) :-
        interpNodeIn(X,Y).

interpNodeInTransitive(X,Z,IMPLIED) :-
        interpNodeIn(X,IMPLIED),
        interpNodeInTransitive(IMPLIED,Z,_).
```

```
declaresExpectationToCause(P, BELIEF, EXPECTATION, AGENT) :-
        actualizesTransitive(P, BELIEFBOX, _),
        interpNodeIn(BELIEF, BELIEFBOX),
        goalOrBeliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT),
        groundTruthOrOfAgent(EXPECTATION, AGENT),
        wouldCauseTransitive(BELIEF, EXPECTATION).

declaresExpectationToCause(P, BELIEF, EXPECTATION, AGENT) :-
        declaresIntention(P, BELIEF, AGENT),
        interpNodeIn(BELIEF, BELIEFBOX),
        goalOrBeliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT),
        groundTruthOrOfAgent(EXPECTATION, AGENT),
        wouldCauseTransitive(BELIEF, EXPECTATION).

declaresExpectationToPrevent(P, BELIEF, EXPECTATION, AGENT) :-
        actualizesTransitive(P, BELIEFBOX, _),
        interpNodeIn(BELIEF, BELIEFBOX),
        goalOrBeliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT),
        groundTruthOrOfAgent(EXPECTATION, AGENT),
        wouldPreventTransitive(BELIEF, EXPECTATION).

declaresExpectationToPrevent(P, BELIEF, EXPECTATION, AGENT) :-
        declaresIntention(P, BELIEF, AGENT),
        interpNodeIn(BELIEF, BELIEFBOX),
        goalOrBeliefBox(BELIEFBOX),
        agent(BELIEFBOX, AGENT),
        groundTruthOrOfAgent(EXPECTATION, AGENT),
        wouldPreventTransitive(BELIEF, EXPECTATION).


actualizesAid(P, I, AFFECT) :-
        actualizesTransitive(P, I),
        wouldCauseTransitive(I, AFFECT),
        affectNode(AFFECT).

actualizesAid(P, I, AFFECT) :-
        ceasesTransitive(P, I),
        wouldPreventTransitive(I, AFFECT),
        affectNode(AFFECT).

actualizesAid(P, AFFECT) :-
        actualizesAid(P, _, AFFECT),
        affectNode(AFFECT).

actualizesAid(P, AFFECT) :-
        actualizesTransitive(P, AFFECT),
        affectNode(AFFECT).
```

```
actualizesHarm(P, I, AFFECT) :-
        actualizesTransitive(P, I),
        wouldPreventTransitive(I, AFFECT),
        affectNode(AFFECT).

actualizesHarm(P, I, AFFECT) :-
        ceasesTransitive(P, I),
        preconditionForTransitive(I, AFFECT),
        affectNode(AFFECT).

actualizesHarm(P, AFFECT) :-
        actualizesHarm(P, _, AFFECT),
        affectNode(AFFECT).

actualizesHarm(P, AFFECT) :-
        ceasesTransitive(P, AFFECT),
        affectNode(AFFECT).

wouldAid(I, AFFECT, AGENT) :-
        preconditionForTransitive(I, AFFECT),
        affectNode(AFFECT),
        agent(AFFECT, AGENT).

wouldHarm(I, AFFECT, AGENT) :-
        wouldPreventTransitive(I, AFFECT),
        affectNode(AFFECT),
        agent(AFFECT, AGENT).


%%%% Supporting rules %%%%
goalBox(A) :-
        type(A,B),
        B==goalBox.
goalBox(A) :-
        type(A,B),
        B==obligationBox.

beliefBox(A) :-
        type(A,B),
        B==beliefBox.

affectNode(A) :-
        type(A,B),
        B==affectNode.

sameAgent(A, B) :-
        agent(A, A_AGENT),
        agent(B, B_AGENT),
        A_AGENT==B_AGENT.
```

```
sameAgentAffectNodes(A, B) :-
        affectNode(A),
        affectNode(B),
        agent(A, A_AGENT),
        agent(B, B_AGENT),
        A_AGENT==B_AGENT.

frame(A) :-
        goalBox(A).

frame(A) :-
        beliefBox(A).

goalOrBeliefBox(A) :-
        goalBox(A).
goalOrBeliefBox(A) :-
        beliefBox(A).

outermostFrame(A) :-
        frame(A),
        \+ interpNodeIn(A,_).


goalOfAgent(GOAL, AGENT) :-
        interpNodeIn(GOAL, GOALBOX),
        agent(GOALBOX, AGENT).

intention(P, I) :-
        attemptToPreventTransitive(P, I).
intention(P, I) :-
        attemptToCauseTransitive(P, I).


actualizesOrImplies(A,B) :-
        actualizesFlat(A,B).

actualizesOrImplies(A,B) :-
        implies(A,B).


groundTruth(A) :-
        \+ interpNodeIn(A,_).

groundTruthOrOfAgent(I, AGENT) :-
        interpNodeIn(I, BOX),
        agent(BOX, AGENT).

groundTruthOrOfAgent(I, AGENT) :-
        groundTruth(I),
        AGENT==AGENT.
```

C.2 Causality

These patterns identify cases where two timeline propositions have a causal relationship (A causes B). See Section 3.3.2.4.

```
causalTimelinePropositions(A,B,IMPLIED1,IMPLIED2) :-
        followedByOrSimultaneous(A,B),
        actualizesFlat(A,IMPLIED1),
        wouldCause(IMPLIED1,IMPLIED2,_),
        actualizesFlat(B,IMPLIED2).

causalTimelinePropositions(A,B,IMPLIED1,IMPLIED2) :-
        followedByOrSimultaneous(A,B),
        actualizesFlat(A,IMPLIED1),
        wouldPrevent(IMPLIED1,IMPLIED2),
        ceasesFlat(B,IMPLIED2).

causalTimelinePropositions(A,B,IMPLIED1,IMPLIED2) :-
        followedByOrSimultaneous(A,B),
        ceasesFlat(A,IMPLIED1),
        preconditionAgainstFlat(IMPLIED1,IMPLIED2),
        actualizesFlat(B,IMPLIED2).

causalTimelinePropositions(A,B,IMPLIED1,IMPLIED2) :-
        followedByOrSimultaneous(A,B),
        ceasesFlat(A,IMPLIED1),
        preconditionForFlat(IMPLIED1,IMPLIED2,_),
        ceasesFlat(B,IMPLIED2).

causalTimelinePropositions(A,B) :-
        causalTimelinePropositions(A,B,_,_).

causalTimelinePropositionsTransitive(A,B) :-
        causalTimelinePropositions(A,IMPLIED,_,_),
        causalTimelinePropositions(IMPLIED,B,_,_).
```

C.3 SIG Pattern Definitions

These definitions cover the SIG patterns described in Appendix B.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.2: Affectual Status Transitions %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

gain(P,AFFECT) :-
        affectNode(AFFECT),
        actualizesTransitive(P,AFFECT).

gain(P,AFFECT) :-
        gain(P, _, AFFECT).

gain(P,I,AFFECT) :-
        setof((P,I,AFFECT),(
        affectNode(AFFECT),
        actualizesTransitive(P,I),
        \+ outermostFrame(I),
        preconditionForTransitive(I, AFFECT)), Result),
        member((P,I,AFFECT),Result).

gain(P,I,AFFECT) :-
        setof((P,I,AFFECT), (
        affectNode(AFFECT),
        ceasesTransitive(P,I),
        \+ outermostFrame(I),
        wouldPreventTransitive(I, AFFECT)), Result),
        member((P,I,AFFECT),Result).

loss(P,AFFECT) :-
        affectNode(AFFECT),
        ceasesTransitive(P,AFFECT).

loss(P,AFFECT) :-
        loss(P, _, AFFECT).

loss(P,I,AFFECT) :-
        setof((P,I,AFFECT),(
        affectNode(AFFECT),
        ceasesTransitive(P,I),
        \+ outermostFrame(I),
        preconditionForTransitive(I, AFFECT)), Result),
        member((P,I,AFFECT), Result).
```

```
loss(P,I,AFFECT) :-
        setof((P,I,AFFECT), (
        affectNode(AFFECT),
        actualizesTransitive(P,I),
        \+ outermostFrame(I),
        wouldPreventTransitive(I, AFFECT)), Result),
        member((P,I,AFFECT), Result).

negativeResolution(P1, P2, AFFECT) :-
        actualizesTransitive(P1, AFFECT),
        affectNode(AFFECT),
        ceasesTransitive(P2, AFFECT),
        followedByTransitive(P1, P2).

positiveResolution(P1, P2, AFFECT) :-
        ceasesTransitive(P1, AFFECT),
        affectNode(AFFECT),
        actualizesTransitive(P2, AFFECT),
        followedByTransitive(P1, P2).

complexPositive(P, AFFECT1, AFFECT2, ROUTE1, ROUTE2) :-
        affectNode(AFFECT1),
        affectNode(AFFECT2),
        actualizesTransitive(P, AFFECT1, ROUTE1),
        actualizesTransitive(P, AFFECT2, ROUTE2),
        ROUTE1 \== ROUTE2,
        \+ preconditionForTransitive(ROUTE1, ROUTE2),
        \+ preconditionForTransitive(ROUTE2, ROUTE1),
        sameAgentAffectNodes(AFFECT1,AFFECT2).

complexNegative(P, AFFECT1, AFFECT2, ROUTE1, ROUTE2) :-
        affectNode(AFFECT1),
        affectNode(AFFECT2),
        ceasesTransitive(P, AFFECT1, ROUTE1),
        ceasesTransitive(P, AFFECT2, ROUTE2),
        ROUTE1 \== ROUTE2,
        \+ preconditionForTransitive(ROUTE1, ROUTE2),
        \+ preconditionForTransitive(ROUTE2, ROUTE1),
        sameAgentAffectNodes(AFFECT1,AFFECT2).

hiddenBlessing(P1, P2, AFFECT1, AFFECT2) :-
        loss(P1, AFFECT1),
        causalTimelinePropositions(P1, P2),
        gain(P2, AFFECT2),
        sameAgentAffectNodes(AFFECT1, AFFECT2).
```

```
positiveTradeoff(P1, P2, AFFECT1, AFFECT2) :-
        loss(P2, AFFECT1),
        sameAgentAffectNodes(AFFECT1, AFFECT2),
        gain(P2, AFFECT2),
        gain(P1, AFFECT1),
        followedByTransitive(P1, P2).

negativeTradeoff(P1, P2, AFFECT1, AFFECT2) :-
        gain(P2, AFFECT1),
        sameAgentAffectNodes(AFFECT1, AFFECT2),
        loss(P2, AFFECT2),
        loss(P1, AFFECT1),
        followedByTransitive(P1, P2).

mixedBlessing(P1, P2, AFFECT1, AFFECT2) :-
        gain(P1, AFFECT1),
        sameAgentAffectNodes(AFFECT1, AFFECT2),
        loss(P2, AFFECT2).

promise(P, POTENTIAL, PROMISE, AFFECT) :-
        affectNode(AFFECT),
        actualizesTransitive(P, PROMISE),
        wouldCauseTransitive(PROMISE, AFFECT, POTENTIAL),
        POTENTIAL\==AFFECT.

threat(P, POTENTIAL, THREAT, AFFECT) :-
        affectNode(AFFECT),
        actualizesTransitive(P, THREAT),
        wouldPreventTransitive(THREAT, AFFECT, POTENTIAL),
        POTENTIAL\==AFFECT.

promiseFulfilled(P1, P2, POTENTIAL, PROMISE, AFFECT) :-
        promise(P1, POTENTIAL, PROMISE, AFFECT),
        actualizesTransitive(P2, POTENTIAL),
        followedByTransitive(P1, P2).

threatFulfilled(P1, P2, POTENTIAL, THREAT, AFFECT) :-
        threat(P1, POTENTIAL, THREAT, AFFECT),
        actualizesTransitive(P2, POTENTIAL),
        followedByTransitive(P1, P2).

promiseBroken(P1, P2, POTENTIAL, PROMISE, AFFECT) :-
        promise(P1, POTENTIAL, PROMISE, AFFECT),
        ceasesTransitive(P2, POTENTIAL),
        followedByTransitive(P1, P2).

threatAvoided(P1, P2, POTENTIAL, THREAT, AFFECT) :-
        threat(P1, POTENTIAL, THREAT, AFFECT),
        ceasesTransitive(P2, POTENTIAL),
        followedByTransitive(P1, P2).
```

```
promiseOrThreat(P1, POTENTIAL, PROMISE, AFFECT) :-
        promise(P1, POTENTIAL, PROMISE, AFFECT).

promiseOrThreat(P1, POTENTIAL, PROMISE, AFFECT) :-
        threat(P1, POTENTIAL, PROMISE, AFFECT).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.3: Examples of Chaining %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

partialResolution(P1, P2, AFFECT1, AFFECT2) :-
        complexNegative(P1, AFFECT1, AFFECT2,_,_),
        positiveResolution(P1, P2, AFFECT1).

compoundedTransition(P1, P2, P3, I, AFFECT) :-
        loss(P1, I, AFFECT),
        followedByTransitive(P1, P2),
        gain(P2, I, AFFECT),
        followedByTransitive(P2, P3),
        loss(P3, I, AFFECT).

compoundedTransition(P1, P2, P3, I, AFFECT) :-
        gain(P1, I, AFFECT),
        followedByTransitive(P1, P2),
        loss(P2, I, AFFECT),
        followedByTransitive(P2, P3),
        gain(P3, I, AFFECT).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.4: Single-Agent Goals and Plans %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

goalDeclared(P, GOAL, GOALBOX) :-
        setof((P, GOAL, GOALBOX), (
        actualizesTransitive(P,GOALBOX,_),
        interpNodeIn(GOAL,GOALBOX),
        goalBox(GOALBOX)), Answers),
        member((P, GOAL, GOALBOX),Answers).

% We can also assume that an agent has a goal if it attempts to cause
% its content directly without having explicitly actualized the goal
% frame. (In the SIG schemata, frames must be actualized before their
% content can be actualized or ceased.)
goalDeclared(P, GOAL, GOALBOX) :-
        setof((P,GOAL,GOALBOX), (
        attemptToCauseTransitive(P,GOAL,_,_),
        interpNodeIn(GOAL, GOALBOX)), Answers),
        member((P,GOAL,GOALBOX), Answers).
```

```
desireToAid(P, GOAL, GOALBOX, AFFECT) :-
        affectNode(AFFECT),
        actualizesTransitive(P,GOALBOX,_),
        interpNodeIn(GOAL,GOALBOX),
        goalBox(GOALBOX),
        wouldCauseTransitive(GOAL, AFFECT).

desireToAid(P, GOAL, GOALBOX, AFFECT) :-
        affectNode(AFFECT),
        attemptToCauseTransitive(P,AFFECT,GOAL,_),
        interpNodeIn(GOAL, GOALBOX).

desireToHarm(P, GOAL, GOALBOX, AFFECT) :-
        affectNode(AFFECT),
        actualizesTransitive(P,GOALBOX,_),
        interpNodeIn(GOAL,GOALBOX),
        goalBox(GOALBOX),
        wouldPreventTransitive(GOAL, AFFECT).

desireToHarm(P, GOAL, GOALBOX, AFFECT) :-
        affectNode(AFFECT),
        attemptToPreventTransitive(P,AFFECT,GOAL,_),
        interpNodeIn(GOAL, GOALBOX).

explicitMotivation(P1, P2, GOALBOX) :-
        causalTimelinePropositions(P1, P2),
        actualizesTransitive(P2, GOALBOX),
        goalBox(GOALBOX).

problem(P1, P2, PROBLEM, AFFECT, PLAN, GOALBOX) :-
        actualizesFlat(P1, PROBLEM),
        wouldPreventTransitive(PROBLEM, AFFECT),
        affectNode(AFFECT),
        followedByTransitive(P1, P2),
        actualizesTransitive(P2, GOALBOX),
        interpNodeIn(PLAN, GOALBOX),
        wouldPreventTransitive(PLAN, PROBLEM).

changeOfMind(P1, P2, GOAL, GOALBOX) :-
        goalDeclared(P1, GOAL, GOALBOX),
        ceasesTransitive(P2, GOALBOX),
        followedByTransitive(P1, P2).

goalEnablement(P1, P2, GOAL, GOALBOX, ENABLER) :-
        setof((P1, GOAL, GOALBOX), (
        goalDeclared(P1, GOAL, GOALBOX)), Answers),
        member((P1, GOAL, GOALBOX), Answers),
        preconditionForTransitive(ENABLER, GOAL),
        actualizesTransitive(P2, ENABLER),
        followedByTransitive(P1, P2).
```

```
goalObstacle(P1, P2, GOAL, GOALBOX, CEASED_PRECONDITION) :-
        setof((P1, GOAL, GOALBOX), (
        goalDeclared(P1, GOAL, GOALBOX)), Answers),
        member((P1, GOAL, GOALBOX), Answers),
        preconditionForTransitive(CEASED_PRECONDITION,GOAL),
        ceasesTransitive(P2, CEASED_PRECONDITION),
        followedByTransitive(P1, P2).

goalSuccessExpected(P1, P2, GOAL, GOALBOX, ENABLER) :-
        goalDeclared(P1, GOAL, GOALBOX),
        wouldCauseTransitive(ENABLER,GOAL),
        actualizesTransitive(P2, ENABLER),
        followedByTransitive(P1, P2).

goalFailureExpected(P1, P2, GOAL, GOALBOX, DISABLER) :-
        goalDeclared(P1, GOAL, GOALBOX),
        wouldPreventTransitive(DISABLER,GOAL),
        actualizesTransitive(P2, DISABLER),
        followedByTransitive(P1, P2).

goalFailureExpected(P1, P2, GOAL, GOALBOX, DISABLER) :-
        goalDeclared(P1, GOAL, GOALBOX),
        preconditionForTransitive(DISABLER, GOAL),
        ceasesTransitive(P2, DISABLER),
        followedByTransitive(P1, P2).

goalAvoidance(P1, P2, TRIGGER, GOAL, GOALBOX) :-
        followedByTransitive(P1, P2),
        actualizesTransitive(P1, TRIGGER),
        wouldCauseTransitive(TRIGGER, GOALBOX),
        ceasesTransitive(P2, GOALBOX),
        interpNodeIn(GOAL, GOALBOX).

goalPreemption(P1, P2, AGENT, PLAN, GOALBOX, PREEMPTED) :-
        goalDeclared(P1, PLAN, GOALBOX),
        wouldPreventTransitive(PLAN, PREEMPTED),
        goalBox(PREEMPTED),
        agent(PREEMPTED, AGENT),
        agent(GOALBOX, AGENT),
        agent(P2, AGENT),
        actualizesTransitive(P2, PLAN),
        followedByTransitive(P1, P2).

perseverance(P1, P2, GOAL, GOALBOX) :-
        setof((P1, GOAL, GOALBOX), (
        goalDeclared(P1, GOAL, GOALBOX)), Result),
        member((P1, GOAL, GOALBOX), Result),
        attemptToCauseTransitive(P2, GOAL),
        followedByTransitive(P1, P2).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.5: Single-Agent Goal Outcomes %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

success(P1, P2, GOAL, GOALBOX) :-
        followedByTransitive(P1, P2),
        goalDeclared(P1, GOAL, GOALBOX),
        actualizesTransitive(P2, GOAL).

failure(P1, P2, GOAL, GOALBOX) :-
        followedByTransitive(P1, P2),
        goalDeclared(P1, GOAL, GOALBOX),
        ceasesTransitive(P2, GOAL).

deliberateAid(P1, P2, GOAL, AFFECT) :-
        attemptToCauseTransitive(P1, GOAL),
        preconditionForTransitive(GOAL, AFFECT),
        affectNode(AFFECT),
        actualizesTransitive(P2, GOAL),
        followedByTransitive(P1, P2).

deliberateAid(P1, P2, GOAL, AFFECT) :-
        attemptToPreventTransitive(P1, GOAL),
        wouldPreventTransitive(GOAL, AFFECT),
        affectNode(AFFECT),
        ceasesTransitive(P2, GOAL),
        followedByTransitive(P1, P2).

deliberateHarm(P1, P2, GOAL, AFFECT) :-
        attemptToCauseTransitive(P1, GOAL),
        wouldPreventTransitive(GOAL, AFFECT),
        affectNode(AFFECT),
        actualizesTransitive(P2, GOAL),
        followedByTransitive(P1, P2).

deliberateHarm(P1, P2, GOAL, AFFECT) :-
        attemptToPreventTransitive(P1, GOAL),
        preconditionForTransitive(GOAL, AFFECT),
        affectNode(AFFECT),
        ceasesTransitive(P2, GOAL),
        followedByTransitive(P1, P2).

unintendedAid(P1, INTENDED, UNINTENDED, AFFECT) :-
        affectNode(AFFECT),
        intention(P1, INTENDED),
        actualizesAid(P1, UNINTENDED, AFFECT),
        INTENDED \== UNINTENDED,
        agent(P1, AGENT),
        \+ declaresIntention(_,INTENDED,AGENT).
```

```
unintendedHarm(P1, INTENDED, UNINTENDED, AFFECT) :-
        affectNode(AFFECT),
        intention(P1, INTENDED),
        actualizesHarm(P1, UNINTENDED, AFFECT),
        INTENDED \== UNINTENDED,
        agent(P1, AGENT),
        \+ declaresIntention(_,INTENDED,AGENT).

backfireType1(P1, AFFECT) :-
        affectNode(AFFECT),
        attemptToCauseTransitive(P1, AFFECT),
        actualizesHarm(P1, AFFECT).

backfireType2(P1, AGENT, PLAN, INTENDED_AFFECT, UNINTENDED_AFFECT) :-
        declaresIntention(P1, PLAN, AGENT),
        interpNodeIn(PLAN, GOALBOX_1),
        wouldCauseTransitive(PLAN, INTENDED_AFFECT),
        interpNodeIn(INTENDED_AFFECT, GOALBOX_2),
        agent(P1, AGENT),
        agent(GOALBOX_1, AGENT),
        agent(GOALBOX_2, AGENT),
        wouldPreventTransitive(PLAN, UNINTENDED_AFFECT),
        \+ declaresIntention(_,UNINTENDED_AFFECT,AGENT),
        affectNode(INTENDED_AFFECT),
        affectNode(UNINTENDED_AFFECT).

lostOpportunity(P1, P2, P3, PRECONDITION, GOAL, AGENT) :-
        actualizesTransitive(P1, PRECONDITION),
        preconditionForTransitive(PRECONDITION, GOAL),
        ceasesTransitive(P3, PRECONDITION),
        goalDeclared(P2, GOAL, GOALBOX),
        agent(GOALBOX, AGENT),
        followedByTransitive(P1, P2),
        followedByTransitive(P2, P3).

goodSideEffect(P1, AGENT, GOAL, INTENDED_AFFECT,
  UNINTENDED, UNINTENDED_AFFECT) :-
        actualizesAid(P1, GOAL, INTENDED_AFFECT),
        actualizesAid(P1, UNINTENDED, UNINTENDED_AFFECT),
        interpNodeIn(GOAL, GOALBOX),
        agent(P1, AGENT),
        agent(GOALBOX, AGENT),
        \+ declaresIntention(_,UNINTENDED_AFFECT,AGENT).
```

```
badSideEffect(P1, AGENT, GOAL, INTENDED_AFFECT,
  UNINTENDED, UNINTENDED_AFFECT) :-
        actualizesAid(P1, GOAL, INTENDED_AFFECT),
        actualizesHarm(P1, UNINTENDED, UNINTENDED_AFFECT),
        interpNodeIn(GOAL, GOALBOX),
        agent(P1, AGENT),
        agent(GOALBOX, AGENT),
        \+ declaresIntention(_,UNINTENDED_AFFECT,AGENT).

recovery(P1, P2, P3, AGENT, DAMAGER, FIXER_GOAL, AFFECT) :-
        actualizesHarm(P1, DAMAGER, AFFECT),
        goalDeclared(P2, FIXER_GOAL, _),
        wouldCauseTransitive(FIXER_GOAL, AFFECT),
        actualizesTransitive(P3, FIXER_GOAL),
        agent(P1, AGENT),
        agent(P2, AGENT),
        followedByOrSimultaneous(P1, P2),
        followedByTransitive(P2, P3).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.6: Complex Single-Agent Goal Outcomes %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

peripeteia(P1, P2, P3, BELIEF, EXPECTATION, AGENT) :-
        declaresExpectationToCause(P1, BELIEF, EXPECTATION, AGENT),
        wouldAid(EXPECTATION, AFFECT, AGENT),
        agent(AFFECT, AGENT),
        actualizesTransitive(P2, BELIEF),
        ceasesTransitive(P3, EXPECTATION),
        followedByTransitive(P1, P2),
        followedByTransitive(P2, P3),
        actualizesHarm(P3, AFFECT2),
        agent(AFFECT2, AGENT).

goalSubstitution(P1, P2, P3, GOAL, NEWGOAL, AGENT) :-
        failure(P1, P2, GOAL, GOALBOX),
        wouldAid(GOAL, AFFECT, AGENT),
        agent(GOALBOX, AGENT),
        causalTimelinePropositions(P2, P3),
        declaresIntention(P3, NEWGOAL, AGENT),
        wouldAid(NEWGOAL, AFFECT, AGENT),
        interpNodeIn(NEWGOAL, NEWGOALBOX),
        agent(NEWGOALBOX, AGENT).

failureGivingUp(P1, P2, P3, GOAL, GOALBOX) :-
        failure(P1, P2, GOAL, GOALBOX),
        ceasesTransitive(P3, GOALBOX),
        followedByOrSimultaneous(P2, P3).
```

```
noir(P1, P2, P3, GOAL, AFFECT, AFFECT2, AGENT) :-
        attemptToCauseTransitive(P1,GOAL),
        wouldAid(GOAL, AFFECT, AGENT),
        agent(P1, AGENT),
        actualizesHarm(P1, AFFECT2),
        agent(AFFECT2, AGENT),
        declaresIntention(P3, AFFECT2, AGENT),
        followedByOrSimultaneous(P2, P3).

obviatedPlan(P1, P2, AGENT, PLAN, OBJECTIVE) :-
        declaresExpectationToCause(P1, PLAN, OBJECTIVE, AGENT),
        interpNodeIn(PLAN, GOALBOX),
        goalBox(GOALBOX),
        agent(GOALBOX, AGENT),
        followedByTransitive(P1, P2),
        actualizesTransitive(P2, OBJECTIVE),
        \+ actualizesTransitive(_, PLAN).

wastedEffortIrony(P1, P2, P3, P4, P5, AGENT, PLAN, OBJECTIVE) :-
        declaresExpectationToCause(P1, PLAN, OBJECTIVE, AGENT),
        interpNodeIn(PLAN, GOALBOX),
        goalBox(GOALBOX),
        agent(GOALBOX, AGENT),
        followedByOrSimultaneous(P1, P2),

        % OBJECTIVE is actualized because PLAN is.
        actualizesTransitive(P2, PLAN),
        followedByOrSimultaneous(P2, P3),
        actualizesTransitive(P3, OBJECTIVE),

        % OBJECTIVE is later actualized at P5 because of what happened
        % at P4, which is not an actualization of PLAN.
        followedByTransitive(P3, P4),
        causalTimelinePropositions(P4, P5),
        actualizesTransitive(P5, OBJECTIVE),
        \+ actualizesTransitive(P4, PLAN).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.7: Complex Single-Agent Goal Outcomes %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mistakenBelief(P1, P2, TRUTH, BELIEF) :-
        actualizesTransitive(P1, TRUTH),
        declaresBelief(P2, BELIEF, _),
        inverseOf(TRUTH, BELIEF).
```

```
mistakenBelief(P1, P2, FALSEHOOD, BELIEF) :-
        ceasesTransitive(P1, FALSEHOOD),
        declaresBelief(P2, BELIEF, _),
        equivalentOf(FALSEHOOD, BELIEF).

mistakenBelief(P1, P2, BELIEF) :-
        declaresBelief(P1, BELIEF, _),
        ceasesTransitive(P2, BELIEF).

violatedExpectation(P1, P2, P3, BELIEF, EXPECTATION, AGENT) :-
        declaresExpectationToCause(P1, BELIEF, EXPECTATION, AGENT),
        actualizesTransitive(P2, BELIEF),
        ceasesTransitive(P3, EXPECTATION),
        followedByTransitive(P1, P2),
        followedByTransitive(P2, P3).

surprise(P1, P2, P3, BELIEF, EXPECTATION, AGENT) :-
        declaresExpectationToPrevent(P1, BELIEF, EXPECTATION, AGENT),
        actualizesTransitive(P2, BELIEF),
        actualizesTransitive(P3, BELIEF),
        followedByTransitive(P1, P2),
        followedByTransitive(P2, P3).

anagnorisis(P1, P2, P3, PRIOR_BELIEF, NEW_BELIEF, AGENT) :-
        declaresBelief(P1, _, AGENT, PRIOR_BELIEF),
        wouldPrevent(NEW_BELIEF, PRIOR_BELIEF),
        declaresBelief(P2, _, AGENT, NEW_BELIEF),
        followedByTransitive(P1, P2),
        ceasesTransitive(P3, PRIOR_BELIEF),
        followedByOrSimultaneous(P2, P3).

anagnorisis(P1, P2, P3, PRIOR_BELIEF, NEW_BELIEF, AGENT) :-
        declaresBelief(P1, PRIOR_CONTENT, AGENT, PRIOR_BELIEF),
        wouldPrevent(NEW_CONTENT, PRIOR_CONTENT),
        declaresBelief(P2, NEW_CONTENT, AGENT, NEW_BELIEF),
        followedByTransitive(P1, P2),
        ceasesTransitive(P3, PRIOR_BELIEF),
        followedByOrSimultaneous(P2, P3).

anagnorisis(P1, P2, _, BELIEF, _, AGENT) :-
        declaresBelief(P1, BELIEF, AGENT, BELIEFBOX),
        ceasesBelief(P2, BELIEF, AGENT, BELIEFBOX),
        followedByTransitive(P1, P2).

potentialContradiction(P1, P2, POSSIBILITY1, POSSIBILITY2, CONFLICT, AGENT) :-
        declaresExpectationToCause(P1, POSSIBILITY1, CONFLICT, AGENT),
        declaresExpectationToPrevent(P2, POSSIBILITY2, CONFLICT, AGENT),
        POSSIBILITY1\==POSSIBILITY2.
```

```
contradictoryBeliefs(P1, P2, P3, P4, POSSIBILITY1, POSSIBILITY2,
  BELIEF1, BELIEF2, CONFLICT, AGENT) :-
        equivalentOf(POSSIBILITY1, BELIEF1),
        equivalentOf(POSSIBILITY2, BELIEF2),
        POSSIBILITY1\==POSSIBILITY2,
        declaresExpectationToCause(P1, POSSIBILITY1, CONFLICT, AGENT),
        declaresExpectationToPrevent(P2, POSSIBILITY2, CONFLICT, AGENT),
        declaresBelief(P3, BELIEF1, AGENT),
        declaresBelief(P4, BELIEF2, AGENT).

mistakenSatisfaction(P1, P2, P3, GOAL, TRUTH, BELIEVED_TRUTH, AGENT) :-
        declaresIntention(P1, GOAL, AGENT),
        ceasesTransitive(P2, TRUTH),
        declaresBelief(P3, BELIEVED_TRUTH, AGENT, _),
        equivalentOf(BELIEVED_TRUTH, GOAL),
        inverseOf(BELIEVED_TRUTH, TRUTH),
        followedByTransitive(P1, P2),
        followedByTransitive(P1, P3).


%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.8: Dilemmas %
%%%%%%%%%%%%%%%%%%%%%%%%

dilemmaType1(P1, POSSIBILITY, TRIGGER, CONSEQUENCE1, CONSEQUENCE2, AGENT) :-
          declaresExpectationToCause(P1, POSSIBILITY, TRIGGER, AGENT),
        wouldAid(TRIGGER, CONSEQUENCE1, AGENT),
        wouldHarm(TRIGGER, CONSEQUENCE2, AGENT),
        agent(POSSIBILITY, AGENT).

dilemmaType2(P1, MUTEX1, MUTEX2, CONSEQUENCE1, CONSEQUENCE2, AGENT) :-
        declaresExpectationToCause(P1, MUTEX1, TRIGGER1, AGENT),
        declaresExpectationToPrevent(P1, MUTEX1, TRIGGER2, AGENT),
        declaresExpectationToCause(P1, MUTEX2, TRIGGER2, AGENT),
        declaresExpectationToPrevent(P1, MUTEX2, TRIGGER1, AGENT),
        \+ equivalentOf(MUTEX1, MUTEX2),
        MUTEX1\==MUTEX2,
        wouldAid(TRIGGER1, CONSEQUENCE1, AGENT),
        wouldAid(TRIGGER2, CONSEQUENCE2, AGENT).

goalPrioritization(P1, CONSEQUENCE1, CONSEQUENCE2, AGENT) :-
        attemptToCauseTransitive(P1, TRIGGER),
        wouldAid(TRIGGER, CONSEQUENCE1, AGENT),
        wouldHarm(TRIGGER, CONSEQUENCE2, AGENT),
        agent(P1, AGENT).
```

```
goalPrioritization(P1, CONSEQUENCE1, CONSEQUENCE2, AGENT) :-
        attemptToPreventTransitive(P1, TRIGGER),
        wouldAid(TRIGGER, CONSEQUENCE1, AGENT),
        wouldHarm(TRIGGER, CONSEQUENCE2, AGENT),
        agent(P1, AGENT).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.9: Two-Agent Interactions %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

selfishAct(P1, P2, TRIGGER, CONSEQUENCE1, CONSEQUENCE2, AGENT, EXPERIENCER) :-
        declaresIntention(P1, TRIGGER, AGENT),
        attemptToCauseTransitive(P2, TRIGGER),
        followedByOrSimultaneous(P1, P2),
        wouldAid(TRIGGER, CONSEQUENCE1, AGENT),
        wouldHarm(TRIGGER, CONSEQUENCE2, EXPERIENCER),
        AGENT\==EXPERIENCER,
        CONSEQUENCE1\==CONSEQUENCE2.

selflessAct(P1, P2, TRIGGER, CONSEQUENCE1, CONSEQUENCE2, AGENT, EXPERIENCER) :-
        declaresIntention(P1, TRIGGER, AGENT),
        attemptToCauseTransitive(P2, TRIGGER),
        followedByOrSimultaneous(P1, P2),
        wouldHarm(TRIGGER, CONSEQUENCE1, AGENT),
        wouldAid(TRIGGER, CONSEQUENCE2, EXPERIENCER),
        AGENT\==EXPERIENCER,
        CONSEQUENCE1\==CONSEQUENCE2.

deliberateAssistance(P1, P2, AGENT, TRIGGER, AFFECT, EXPERIENCER) :-
        declaresIntention(P1, TRIGGER, AGENT),
        attemptToCauseTransitive(P2, TRIGGER),
        followedByOrSimultaneous(P1, P2),
        wouldAid(TRIGGER, AFFECT, EXPERIENCER),
        AGENT\==EXPERIENCER.

commonlyPursuedGoal(P1, P2, AGENT1, AGENT2, GOAL) :-
        declaresIntention(P1, GOAL, AGENT1),
        declaresIntention(P2, GOAL, AGENT2),
        AGENT1\==AGENT2.
```

```
tandemAttempts(P1, P2, P3, GOAL1, GOAL2, AGENT, EXPERIENCER) :-
        setof((GOAL1, GOAL2, P3), (
        attemptToCauseTransitive(P3, GOAL1),
        attemptToCauseTransitive(P3, GOAL2),
        GOAL1\==GOAL2,
        \+ preconditionForTransitive(GOAL1, GOAL2),
        \+ preconditionForTransitive(GOAL2, GOAL1)), Result),
        member((GOAL1, GOAL2, P3), Result),

        declaresIntention(P1, GOAL1, AGENT),
        declaresIntention(P2, GOAL2, EXPERIENCER),
        followedByOrSimultaneous(P1, P3),
        followedByOrSimultaneous(P2, P3),
        AGENT\==EXPERIENCER.

tandemAttempts(P1, P2, P3, GOAL1, GOAL2, AGENT, EXPERIENCER) :-
        setof((GOAL1, GOAL2, P3), (
        attemptToCauseTransitive(P3, GOAL1),
        attemptToPreventTransitive(P3, GOAL2),
        GOAL1\==GOAL2,
        \+ wouldPreventTransitive(GOAL1, GOAL2),
        \+ wouldPreventTransitive(GOAL2, GOAL1)), Result),
        member((GOAL1, GOAL2, P3), Result),

        declaresIntention(P1, GOAL1, AGENT),
        declaresIntention(P2, GOAL2, EXPERIENCER),
        followedByOrSimultaneous(P1, P3),
        followedByOrSimultaneous(P2, P3),
        AGENT\==EXPERIENCER.

conflictType1(P1, P2, AGENT1, AGENT2, GOAL) :-
        attemptToCauseTransitive(P1, GOAL),
        attemptToPreventTransitive(P2, GOAL),
        agent(P1, AGENT1),
        agent(P2, AGENT2),
        AGENT1\==AGENT2.

conflictType2(P1, P2, AGENT1, AGENT2, MUTEX1, MUTEX2,
  CONSEQUENCE1, CONSEQUENCE2) :-
        declaresExpectationToCause(P1, MUTEX1, CONSEQUENCE1, AGENT1),
        wouldPreventTransitive(MUTEX1, CONSEQUENCE2),
        declaresExpectationToCause(P2, MUTEX2, CONSEQUENCE2, AGENT2),
        wouldPreventTransitive(MUTEX2, CONSEQUENCE1),
        CONSEQUENCE1\==CONSEQUENCE2,
        AGENT1\==AGENT2,
        MUTEX1\==MUTEX2,
        goalOfAgent(CONSEQUENCE1, AGENT1),
        goalOfAgent(CONSEQUENCE2, AGENT2).
```

```
giftOfTheMagiIrony(P1, P2, P3, P4, AGENT1, AGENT2, MUTEX1, MUTEX2,
  CONSEQUENCE1, CONSEQUENCE2) :-
        declaresExpectationToCause(P1, MUTEX1, CONSEQUENCE1, AGENT1),
        declaresExpectationToCause(P2, MUTEX2, CONSEQUENCE2, AGENT2),
        goalOfAgent(CONSEQUENCE1, AGENT1),
        goalOfAgent(CONSEQUENCE2, AGENT2),

        AGENT1\==AGENT2,
        MUTEX1\==MUTEX2,
        CONSEQUENCE1\==CONSEQUENCE2,

        wouldPreventTransitive(MUTEX1, CONSEQUENCE2),
        wouldPreventTransitive(MUTEX2, CONSEQUENCE1),
        wouldAid(CONSEQUENCE1, AFFECT2, AGENT2),
        wouldAid(CONSEQUENCE2, AFFECT1, AGENT1),
        wouldHarm(MUTEX1, AFFECT1, AGENT1),
        wouldHarm(MUTEX2, AFFECT2, AGENT2),

        followedByOrSimultaneous(P3, P1),
        followedByOrSimultaneous(P3, P2),
        followedByOrSimultaneous(P4, P1),
        followedByOrSimultaneous(P4, P2),

        agent(P3, AGENT1),
        agent(P4, AGENT2),
        attemptToCauseTransitive(P3, MUTEX1),
        attemptToCauseTransitive(P4, MUTEX2),
        ceasesTransitive(P3, CONSEQUENCE2),
        ceasesTransitive(P4, CONSEQUENCE1).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.10: Persuasion and Deception %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

persuasion(P1, P2, AGENT, EXPERIENCER, THEME) :-
        success(P1, P2, THEME, GOALBOX),
        agent(GOALBOX, AGENT),
        goalOrBeliefBox(THEME),
        agent(THEME, EXPERIENCER).

deception(P1, P2, P3, P4, AGENT, EXPERIENCER, THEME) :-
        actualizesTransitive(P1, THEME),
        declaresBelief(P2, THEME2, AGENT),
        inverseOf(THEME_INVERSE, THEME),
        inverseOf(THEME_INVERSE, THEME2),
        followedByOrSimultaneous(P3, P1),
        followedByOrSimultaneous(P3, P2),
        followedByOrSimultaneous(P4, P1),
        followedByOrSimultaneous(P4, P2),
        persuasion(P2, P3, AGENT, EXPERIENCER, BELIEFBOX),
        beliefBox(BELIEFBOX),
        interpNodeIn(THEME_INVERSE, BELIEFBOX).

unintendedPersuasion(P1, P2, P3, P4, AGENT, EXPERIENCER, THEME) :-
        actualizesTransitive(P1, THEME),
        declaresBelief(P2, THEME_INVERSE, AGENT),
        inverseOf(THEME_INVERSE, THEME),
        followedByOrSimultaneous(P3, P1),
        followedByOrSimultaneous(P3, P2),
        followedByOrSimultaneous(P4, P1),
        followedByOrSimultaneous(P4, P2),
        persuasion(P2, P3, AGENT, EXPERIENCER, BELIEFBOX),
        beliefBox(BELIEFBOX),
        interpNodeIn(THEME2, BELIEFBOX),
        equivalentOf(THEME2, THEME).
```

```
mutualDeception(P1, P2, P3, P4, AGENT, EXPERIENCER, THEME) :-

        actualizesTransitive(P1, THEME),
        declaresBelief(P2, THEME2, AGENT),
        declaresBelief(P3, THEME3, EXPERIENCER),
        AGENT\==EXPERIENCER,

        equivalentOf(THEME2, THEME),
        equivalentOf(THEME3, THEME),

        attemptToCauseTransitive(P4, DECEPTION_GOAL),
        agent(P4, AGENT),
        beliefBox(DECEPTION_GOAL),
        agent(DECEPTION_GOAL, EXPERIENCER),
        interpNodeIn(INVERSE_THEME, DECEPTION_GOAL),
        inverseOf(INVERSE_THEME, THEME),

        attemptToCauseTransitive(P5, COUNTER_DECEPTION_GOAL),
        agent(P5, EXPERIENCER),
        beliefBox(COUNTER_DECEPTION_GOAL),
        agent(COUNTER_DECEPTION_GOAL, AGENT),

        interpNodeIn(COUNTER_DECEPTION_GOAL_INNER, COUNTER_DECEPTION_GOAL),
        beliefBox(COUNTER_DECEPTION_GOAL_INNER),
        agent(COUNTER_DECEPTION_GOAL_INNER, EXPERIENCER),
        interpNodeIn(INVERSE_THEME_2, COUNTER_DECEPTION_GOAL_INNER),
        inverseOf(INVERSE_THEME_2, THEME),

        followedBy(P4, P1),
        followedBy(P4, P2),
        followedBy(P4, P3),
        followedBy(P5, P4).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.11: Complex Two-Agent Interactions %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

motivatedToRevenge(P1, P2, P3, AGENT, EXPERIENCER) :-
        deliberateHarm(P1, P2, _, AFFECT),
        agent(P1, AGENT),
        agent(AFFECT, EXPERIENCER),
        AGENT\==EXPERIENCER,

        followedByTransitive(P2, P3),
        causalTimelinePropositions(P2, P3),
        declaresIntention(P3, REVENGE_GOAL, EXPERIENCER),
        wouldHarm(REVENGE_GOAL, _, AGENT).
```

```
motivatedToReturnFavor(P1, P2, P3, AGENT, EXPERIENCER) :-
        deliberateAid(P1, P2, _, AFFECT),
        agent(P1, AGENT),
        agent(AFFECT, EXPERIENCER),
        AGENT\==EXPERIENCER,

        followedByTransitive(P2, P3),
        causalTimelinePropositions(P2, P3),
        declaresIntention(P3, REVENGE_GOAL, EXPERIENCER),
        wouldAid(REVENGE_GOAL, _, AGENT).

successfulCoercion(P1, P2, P3, AGENT, EXPERIENCER) :-
        declaresExpectationToCause(P1, THREAT, ULTIMATE_GOAL, AGENT),
        goalBox(THREAT),
        agent(THREAT, EXPERIENCER),
        interpNodeIn(COERCED_ACTION, THREAT),
        wouldAid(COERCED_ACTION, _, EXPERIENCER),
        wouldAid(ULTIMATE_GOAL, _, AGENT),
        actualizesTransitive(P2, THREAT),
        actualizesTransitive(P3, ULTIMATE_GOAL),
        followedByOrSimultaneous(P2, P1),
        followedByOrSimultaneous(P3, P2).

hiddenAgenda(P1, AGENT, PURPORTED, ULTIMATE_GOAL, EXPERIENCER) :-
        declaresExpectationToCause(P1, PURPORTED, ULTIMATE_GOAL, AGENT),
        goalBox(PURPORTED),
        agent(PURPORTED, EXPERIENCER),
        interpNodeIn(PROPOSED_ACTION, PURPORTED),
        wouldAid(PROPOSED_ACTION, _, EXPERIENCER),
        wouldAid(ULTIMATE_GOAL, _, AGENT).

betrayal(P1, P2, P3, P4, AGENT, EXPERIENCER, ACTION) :-
        success(P1, P3, BELIEF, _),
        beliefBox(BELIEF),
        agent(BELIEF, EXPERIENCER),

        interpNodeIn(PURPORTED_GOAL, BELIEF),
        goalBox(PURPORTED_GOAL),
        agent(PURPORTED_GOAL, AGENT),
        ceasesTransitive(P2, PURPORTED_GOAL),
        followedByTransitive(P2, P3),

        interpNodeIn(ACTION, PURPORTED_GOAL),
        wouldAid(ACTION, _, EXPERIENCER),
        followedByTransitive(P3, P4),
        ceasesTransitive(P4, ACTION).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.13: Manipulation of Time %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

flashback(P1, P2) :-
        sourceTextBeginOffset(P1, S1),

        followedBy(P1, P2),
        sourceTextBeginOffset(P2, S2),
        S2 @< S1,

        followedBy(P2, P3),
        sourceTextBeginOffset(P3, S3),
        S3 @< S1,
        S3 @> S2,

        followedByTransitive(P3, P4),
        sourceTextBeginOffset(P4, S4),
        S4 @> S1.

flashforward(P1, P2) :-
        sourceTextBeginOffset(P1, S1),

        followedBy(P1, P2),
        sourceTextBeginOffset(P2, S2),
        S2 @> S1,

        followedBy(P2, P3),
        sourceTextBeginOffset(P3, S3),
        S3 @> S2,

        followedByTransitive(P3, P4),
        sourceTextBeginOffset(P4, S4),
        S4 @> S1,
        S4 @< S2.

suspense(P1, P2, P3, PROMISE, POTENTIAL) :-
        followedBy(P1, P2),
        followedBy(P2, P3),
        promiseOrThreat(P1, POTENTIAL, PROMISE, _),
        \+ actualizesTransitive(P1, POTENTIAL),
        \+ ceasesTransitive(P1, POTENTIAL),
        \+ actualizesTransitive(P2, POTENTIAL),
        \+ ceasesTransitive(P2, POTENTIAL),
        \+ actualizesTransitive(P3, POTENTIAL),
        \+ ceasesTransitive(P3, POTENTIAL).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURE B.14: Mystery %
%%%%%%%%%%%%%%%%%%%%%%%%

mysteryType1(P1, P2, MYSTERIOUS_EVENT) :-
        causalTimelinePropositions(P1, P2, _, MYSTERIOUS_EVENT),
        sourceTextBeginOffset(P1, S1),
        sourceTextBeginOffset(P2, S2),
        S1 @> S2.

mysteryType2(P1, P2, MYSTERIOUS_EVENT) :-
        causalTimelinePropositions(P1, P2, _, MYSTERIOUS_EVENT),
        interpNodeIn(P1, BELIEFBOX), % Alternate timeline
        followedByTransitive(P2, P3),
        actualizesTransitive(P3, BELIEFBOX). % Revelation

mysteryType3(P1, P2, MYSTERIOUS_EVENT, AGENT) :-
        attemptToCauseTransitive(P1, MYSTERIOUS_EVENT),
        interpNodeIn(MYSTERIOUS_EVENT, GOALBOX),
        agent(GOALBOX, AGENT),
        agent(P1, AGENT),
        \+ declaredPrior(GOALBOX, P1),
        followedByTransitive(P1, P2),
        actualizesTransitive(P2, MYSTERIOUS_EVENT).

declaredPrior(GOALBOX, CUTOFF_TIME) :-
        followedByTransitive(P1, CUTOFF_TIME),
        actualizesFlat(P1, GOALBOX).

mystery(P1, P2, MYSTERIOUS_EVENT) :-
        mysteryType1(P1, P2, MYSTERIOUS_EVENT).

mystery(P1, P2, MYSTERIOUS_EVENT) :-
        mysteryType2(P1, P2, MYSTERIOUS_EVENT).

mystery(P1, P2, MYSTERIOUS_EVENT) :-
        mysteryType3(P1, P2, MYSTERIOUS_EVENT, _).
```

Appendix D

Selected Aesop Fables

The full texts of the fables that we used in the DramaBank collection, attributed to Aesop and translated by Jones [1912], are reproduced below with minor lexical translation changes.

The Ape and the Fisherman

A fisherman was catching fish by the sea. A monkey saw him, and wanted to imitate what he was doing. The man went away into a little cave to take a rest, leaving his net on the beach. The monkey came and grabbed the net, thinking that he too would go fishing. But since he didn't know anything about it and had not had any training, the monkey got tangled up in the net, fell into the sea, and was drowned. The fisherman seized the monkey when he was already done for and said, "You wretched creature! Your lack of judgment and stupid behaviour has cost you your life!"

The Cat and the Mice

There was once a house that was overrun with Mice. A Cat heard of this, and said to herself, "That's the place for me," and off she went and took up her quarters in the house, and caught the Mice one by one and ate them. At last the Mice could stand it no longer, and they determined to take to their holes and stay there. "That's awkward," said the Cat to herself: "the only thing to do is to coax them out by a trick." So she considered a while, and then climbed up the wall and let herself hang down by her hind legs from a peg, and pretended to be dead. By and by a Mouse peeped out and saw the Cat hanging there. "Aha!" it cried, "you're very clever, madam, no doubt: but you may turn yourself into a bag of meal hanging there, if you like, yet you won't catch us coming anywhere near you."

The Crow and the Pitcher

A thirsty Crow found a Pitcher with some water in it, but so little was there that, try as she might, she could not reach it with her beak, and it seemed as though she would die of thirst within sight of the remedy. At last she hit upon a clever plan. She began dropping pebbles into the Pitcher, and with each pebble the water rose a little higher until at last it reached the brim, and the knowing bird was enabled to quench her thirst.

The Dog and His Shadow

A Dog was crossing a plank bridge over a stream with a piece of meat in his mouth, when he happened to see his own reflection in the water. He thought it was another dog with a piece of meat twice as big; so he let go his own, and flew at the other dog to get the larger piece. But, of course, all that happened was that he got neither; for one was only a shadow, and the other was carried away by the current.

The Dog and the Wolf

A Dog was lying in the sun before a farmyard gate when a Wolf pounced upon him and was just going to eat him up; but he begged for his life and said, "You see how thin I am and what a wretched meal I should make you now: but if you will only wait a few days my master is going to give a feast. All the rich scraps and pickings will fall to me and I shall get nice and fat: then will be the time for you to eat me." The Wolf thought this was a very good plan and went away. Some time afterwards he came to the farmyard again, and found the Dog lying out of reach on the stable roof. "Come down," he called, "and be eaten: you remember our agreement?" But the Dog said coolly, "My friend, if ever you catch me lying down by the gate there again, don't you wait for any feast."

The Donkey and the Mule

A Muleteer set forth on a journey, driving before him a Donkey and a Mule, both well laden. The Donkey, as long as he traveled along the plain, carried his load with ease, but when he began to ascend the steep path of the mountain, felt his load to be more than he could bear. He entreated his companion to relieve him of a small portion, that he might carry home the rest; but the Mule paid no attention to the request. The Donkey shortly afterwards fell down dead under his burden. Not knowing what else to do in so wild a region, the Muleteer placed upon the Mule the load carried by the Donkey in addition to his own, and at the top of all placed the hide of the Donkey, after he had skinned him. The Mule, groaning beneath his heavy burden, said to himself: "I am treated according to my deserts. If I had only been willing to assist the Donkey a little in his need, I should not now be bearing, together with his burden, himself as well."

The Eagle and the Roosters

There were two roosters in the same farmyard, and they fought to decide who should be master. When the fight was over, the beaten one went and hid himself in a dark corner; while the victor flew up on to the roof of the stables and crowed lustily. But an Eagle espied him from high up in the sky, and swooped down and carried him off. Forthwith the other rooster came out of his corner and ruled the roost without a rival.

The Farmer and the Fox

A Farmer was greatly annoyed by a Fox, which came prowling about his yard at night and carried off his fowls. So he set a trap for him and caught him; and in order to be revenged upon him, he tied a bunch of tow to his tail and set fire to it and let him go. As ill-luck would have it, however, the Fox made straight for the fields where the corn was standing ripe and ready for cutting. It quickly caught fire and was all burnt up, and the Farmer lost all his harvest.

The Farmer and the Viper

One winter a Farmer found a Viper frozen and numb with cold, and out of pity picked it up and placed it in his bosom. The Viper was no sooner revived by the warmth than it turned upon its benefactor and inflicted a fatal bite upon him; and as the poor man lay dying, he cried, "I have only got what I deserved, for taking compassion on so villainous a creature."

The Fox and the Crow

A Crow was sitting on a branch of a tree with a piece of cheese in her beak when a Fox observed her and set his wits to work to discover some way of getting the cheese. Coming and standing under the tree he looked up and said, "What a noble bird I see above me! Her beauty is without equal, the hue of her plumage exquisite. If only her voice is as sweet as her looks are fair, she ought without doubt to be Queen of the Birds." The Crow was hugely flattered by this, and just to show the Fox that she could sing she gave a loud caw. Down came the cheese, of course, and the Fox, snatching it up, said, "You have a voice, madam, I see: what you want is wits."

The Fox and the Grapes

A hungry Fox saw some fine bunches of Grapes hanging from a vine that was trained along a high trellis, and did his best to reach them by jumping as high as he could into the air. But it was all in vain, for they were just out of reach: so he gave up trying, and walked away with an air of dignity and unconcern, remarking, "I thought those Grapes were ripe, but I see now they are quite sour."

The Fox and the Stork

A Fox invited a Stork to dinner, at which the only fare provided was a large flat dish of soup. The Fox lapped it up with great relish, but the Stork with her long bill tried in vain to partake of the savoury broth. Her evident distress caused the sly Fox much amusement. But not long after the Stork invited him in turn, and set before him a pitcher with a long and narrow neck, into which she could get her bill with ease. Thus, while she enjoyed her dinner, the Fox sat by hungry and helpless, for it was impossible for him to reach the tempting contents of the vessel.

The Goose that Laid the Golden Eggs

A Man and his Wife had the good fortune to possess a Goose which laid a Golden Egg every day. Lucky though they were, they soon began to think they were not getting rich fast enough, and, imagining the bird must be made of gold inside, they decided to kill it in order to secure the whole store of precious metal at once. But when they cut it open they found it was just like any other goose. Thus, they neither got rich all at once, as they had hoped, nor enjoyed any longer the daily addition to their wealth.

The Lion and the Boar

One hot and thirsty day in the height of summer a Lion and a Boar came down to a little spring at the same moment to drink. In a trice they were quarrelling as to who should drink first. The quarrel soon became a fight and they attacked one another with the utmost fury. Presently, stopping for a moment to take breath, they saw some vultures seated on a rock above evidently waiting for one of them to be killed, when they would fly down and feed upon the carcase. The sight sobered them at once, and they made up their quarrel, saying, “We had much better be friends than fight and be eaten by vultures.”

The Lion and the Hare

A Lion found a Hare sleeping in her form, and was just going to devour her when he caught sight of a passing stag. Dropping the Hare, he at once made for the bigger game; but finding, after a long chase, that he could not overtake the stag, he abandoned the attempt and came back for the Hare. When he reached the spot, however, he found she was nowhere to be seen, and he had to go without his dinner. “It serves me right,” he said; “I should have been content with what I had got, instead of hankering after a better prize.”

The Lion and the Mouse

A Lion asleep in his lair was waked up by a Mouse running over his face. Losing his temper he seized it with his paw and was about to kill it. The Mouse, terrified, piteously entreated him to spare its life. “Please let me go,” it cried, “and one day I will repay you for your kindness.” The idea of so insignificant a creature ever being able to do anything for him amused the Lion so much that he laughed aloud, and good-humouredly let it go. But the Mouse’s chance came, after all. One day the Lion got entangled in a net which had been spread for game by some hunters, and the Mouse heard and recognised his roars of anger and ran to the spot. Without more ado it set to work to gnaw the ropes with its teeth, and succeeded before long in setting the Lion free. “There!” said the Mouse, “you laughed at me when I promised I would repay you: but now you see, even a Mouse can help a Lion.”

The Lion In Love

A Lion fell deeply in love with the daughter of a cottager and wanted to marry her; but her father was unwilling to give her to so fearsome a husband, and yet didn’t want to offend the Lion; so he hit upon the following expedient. He went to the Lion and said, “I think you will make a very good husband for my daughter: but I cannot consent to your union unless you let me draw your teeth and pare your nails, for my daughter is terribly afraid of them.” The Lion was so much in love that he readily agreed that this should be done. When once, however, he was thus disarmed, the Cottager was afraid of him no longer, but drove him away with his club.

The Milkmaid and Her Pail

A farmer’s daughter had been out to milk the cows, and was returning to the dairy carrying her pail of milk upon her head. As she walked along, she fell a-musing after this fashion: “The milk in this pail will provide me with cream, which I will make into butter and take to market to sell. With the money I will buy a number of eggs, and these, when hatched, will produce chickens, and by and by I shall have quite a large poultry-yard. Then I shall sell some of my fowls, and with the money which they will bring in I will buy myself a new gown, which I shall wear when I go to the fair; and all the young fellows will admire it, and come and make love to me, but I shall toss my head and have nothing to say to them.” Forgetting all about the pail, and suiting the action to the word, she tossed her head. Down went the pail, all the milk was spilled, and all her fine castles in the air vanished in a moment!

The Shepherd and the Eagle

One day a Jackdaw saw an Eagle swoop down on a lamb and carry it off in its talons. “My word,” said the Jackdaw, “I’ll do that myself.” So it flew high up into the air, and then came shooting down with a great whirring of wings on to the back of a big ram. It had no sooner alighted than its claws got caught fast in the wool, and nothing it could do was of any use: there it stuck, flapping away, and only making things worse instead of better. By and by up came the Shepherd. “Oho,” he said, “so that’s what you’d be doing, is it?” And he took the Jackdaw, and clipped its wings and carried it home to his children. It looked so odd that they didn't know what to make of it. "What sort of bird is it, father?" they asked. "It's a Jackdaw," he replied, "and nothing but a Jackdaw: but it wants to be taken for an Eagle."

The Shepherd's Boy and the Wolf

A Shepherd's Boy was tending his flock near a village, and thought it would be great fun to hoax the villagers by pretending that a Wolf was attacking the sheep: so he shouted out, "Wolf! wolf!" and when the people came running up he laughed at them for their pains. He did this more than once, and every time the villagers found they had been hoaxed, for there was no Wolf at all. At last a Wolf really did come, and the Boy cried, "Wolf! wolf!" as loud as he could: but the people were so used to hearing him call that they took no notice of his cries for help. And so the Wolf had it all his own way, and killed off sheep after sheep at his leisure.

The Serpent and the Eagle

An Eagle swooped down upon a Serpent and seized it in his talons with the intention of carrying it off and devouring it. But the Serpent was too quick for him and had its coils round him in a moment; and then there ensued a life-and-death struggle between the two. A countryman, who was a witness of the encounter, came to the assistance of the Eagle, and succeeded in freeing him from the Serpent and enabling him to escape. In revenge the Serpent spat some of his poison into the man's drinking-horn. Heated with his exertions, the man was about to slake his thirst with a draught from the horn, when the Eagle knocked it out of his hand, and spilled its contents upon the ground.

The Tortoise and the Eagle

A Tortoise, discontented with his lowly life, and envious of the birds he saw disporting themselves in the air, begged an Eagle to teach him to fly. The Eagle protested that it was idle for him to try, as nature had not provided him with wings; but the Tortoise pressed him with entreaties and promises of treasure, insisting that it could only be a question of learning the craft of the air. So at length the Eagle consented to do the best he could for him, and picked him up in his talons. Soaring with him to a great height in the sky he then let him go, and the wretched Tortoise fell headlong and was dashed to pieces on a rock.

The Wily Lion

A Lion watched a fat Bull feeding in a meadow, and his mouth watered when he thought of the royal feast he would make, but he did not dare to attack him, for he was afraid of his sharp horns. Hunger, however, presently compelled him to do something: and as the use of force did not promise success, he determined to resort to artifice. Going up to the Bull in friendly fashion, he said to him, “I cannot help saying how much I admire your magnificent figure. What a fine head! What powerful shoulders and thighs! But, my dear friend, what in the world makes you wear those ugly horns? You must find them as awkward as they are unsightly. Believe me, you would do much better without them.” The Bull was foolish enough to be persuaded by this flattery to have his horns cut off; and, having now lost his only means of defense, fell an easy prey to the Lion.

The Wolf and the Lamb

A Wolf came upon a Lamb straying from the flock, and felt some compunction about taking the life of so helpless a creature without some plausible excuse; so he cast about for a grievance and said at last, “Last year, sirrah, you grossly insulted me.” “That is impossible, sir,” bleated the Lamb, “for I wasn’t born then.” “Well,” retorted the Wolf, “you feed in my pastures.” “That cannot be,” replied the Lamb, “for I have never yet tasted grass.” “You drink from my spring, then,” continued the Wolf. “Indeed, sir,” said the poor Lamb, “I have never yet drunk anything but my mother’s milk.” “Well, anyhow,” said the Wolf, “I’m not going without my dinner”: and he sprang upon the Lamb and devoured it without more ado.

The Wolf and the Shepherd

A Wolf hung about near a flock of sheep for a long time, but made no attempt to molest them. The Shepherd at first kept a sharp eye on him, for he naturally thought he meant mischief: but as time went by and the Wolf showed no inclination to meddle with the flock, he began to look upon him more as a protector than as an enemy: and when one day some errand took him to the city, he felt no uneasiness at leaving the Wolf with the sheep. But as soon as his back was turned the Wolf attacked them and killed the greater number. When the Shepherd returned and saw the havoc he had wrought, he cried, “It serves me right for trusting my flock to a Wolf.”

The Wolf in Sheep’s Clothing

A Wolf resolved to disguise himself in order that he might prey upon a flock of sheep without fear of detection. So he clothed himself in a sheepskin, and slipped among the sheep when they were out at pasture. He completely deceived the shepherd, and when the flock was penned for the night he was shut in with the rest. But that very night as it happened, the shepherd, requiring a supply of mutton for the table, laid hands on the Wolf in mistake for a Sheep, and killed him with his knife on the spot.

Bibliography

[Aarts *et al.*, 2008] Henk Aarts, Ap Dijksterhuis, and Giel Dik. Goal contagion: Inferring goals from others' actions - and what it leads to. In James Y. Shah and Wendi L. Gardner, editors, *Handbook of motivation science*. Guilford, New York, 2008. 3.2.1

[Agarwal and Rambow, 2010] Apoorv Agarwal and Owen Rambow. Automatic detection and classification of social events. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, Cambridge, Massachusetts, 2010. 2.1, 6.2.1

[Albrecht *et al.*, 1995] Jason E. Albrecht, Edward J. O'Brien, Robert A. Mason, and Jerome L. Myers. The role of perspective in the accessibility of goals during reading. *Journal of Experimental Psychology*, 21(2):364–372, 1995. 3.2.1

[Allen and Ferguson, 1994] James F. Allen and George Ferguson. Actions and events in interval temporal logic. Technical Report 521, University of Rochester, Rochester, New York, 1994. 3.3.1

[Allen, 1983] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. 4.4.4

[Allen, 1984] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984. 4.4.2

[Allen, 1991] James F. Allen. Time and time again: The many ways to represent time. *International Journal of Intelligent Systems*, 6(4), 1991. 3.3.1

[Appling and Riedl, 2009] D. Scott Appling and Mark O. Riedl. Representations for learning to summarize plots. In *Proceedings of the AAAI Spring Symposium on Intelligent Narrative Technologies II*, Palo Alto, California, 2009. 3.2.3, 6.2.2

[Aristotle, 1961] Aristotle. Poetics. In Francis Fergusson and S. H. Butcher, editors, *Aristotle's Poetics*. Hill and Wang, 1961. Trans. S.H. Butcher. 3.1, 3.3.2.3

[Bailey, 1999] Paul Bailey. Searching for storiness: Story-generation from a reader's perspective. In Mateas and Sengers [1999]. 6.2.2

[Bakhtin, 1981] Mikhail Bakhtin. Forms of time and of the chronotope in the novel. In *The Dialogic Imagination: Four Essays*, pages 84–258. University of Texas Press, Austin, 1981. Trans. Michael Holquist and Caryl Emerson. 2.2

[Bal, 1981] Mieke Bal. Notes on narrative embedding. *Poetics Today*, 2(2):41–59, 1981. 3.3.1

[Bal, 1997] Mieke Bal. *Narratology: Introduction to the Theory of Narrative*. University of Toronto Press, Toronto, second edition, 1997. 3.1, 3.2.1, 5.2, B.6

[Balahur *et al.*, 2009] Alexandra Balahur, Ralf Steinberger, Erik van der Goot, Bruno Pouliquen, and Mijail Kabadjov. Opinion mining on newspaper quotations. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, Milano, Italy, 2009. 2.5

[Banfield, 1973] Ann Banfield. Narrative style and the grammar of direct and indirect speech. *Foundations of Language*, 10(1):1–39, 1973. 2.7.1

[Banfield, 1982] Ann Banfield. *Unspeakable sentences: narration and representation in the language of fiction*. Routledge, 1982. 2.5.1

[Barber and Kudenko, 2008] Heather Barber and Daniel Kudenko. Generation of dilemma-based interactive narratives with a changeable story goal. In *Proceedings of the 2nd international conference on intelligent technologies for interactive entertainment (INTETAIN '08)*, Cancún, Mexico, 2008. 3.2.3

[Baron-Cohen, 1989] Simon Baron-Cohen. The autistic child's theory of mind: a case of specific developmental delay. *Journal of Child Psychology and Psychiatry*, 30(2):285–297, 1989. 3.2.2

[Barry and Elmes, 1997] David Barry and Michael Elmes. Strategy retold: Toward a narrative view of strategic discourse. *The Academy of Management Review*, 22(2):429–452, 1997. 6.2.2

[Barthes, 1974] Roland Barthes. *S/Z: An Essay*. Hill and Wang, New York City, 1974. Trans. Richard Miller. 3.2.2

[Barthes, 1975] Roland Barthes. An introduction to the structural analysis of narrative. *New Literary History*, 6(2):237–272, 1975. Trans. Lionel Duisit. 1, 3.2.2

[Barthes, 1978] Roland Barthes. *The Death of The Author*, pages 142–154. Hill and Wang, 1978. 3.2.2

[Bartlett, 1932] Frederic C. Bartlett. *Remembering: a study in experimental and social psychology*. Cambridge University Press, Cambridge, 1932. 1

[Barzilay and Lee, 2003] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL/HLT 2003*, pages 16–23, Edmonton, Canada, 2003. 5.2

[Barzilay and McKeown, 2001] Regina Barzilay and Kathleen R McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL/EACL 2001)*, Toulouse, France, 2001. 5.2.1

[Bearman and Stovel, 2000] Peter S. Bearman and Katherine Stovel. Becoming a nazi: A model for narrative networks. *Poetics*, 27:69–90, 2000. 3.2.1

[Bell, 1999] Allan Bell. News stories as narratives. In Adam Jaworski and Nikolas Coupland, editors, *The Discourse Reader*, pages 236–251. Routledge, London & New York, 1999. 3.2.2

[Berman and Nir-Sagiv, 2007] Ruth A. Berman and Bracha Nir-Sagiv. Comparing narrative and expository text construction across adolescence: A developmental paradox. *Discourse Processes*, 43(2):79–120, 2007. 3.2.2

[Bers, 1999] Marina Bers. Narrative construction kits: "who am i? who are you? what are we?". In Mateas and Sengers [1999]. 4.3.1

[Bickmore and Cassell, 1999] Timothy Bickmore and Justine Cassell. Small talk and conversational storytelling in embodied conversational interface agents. In Mateas and Sengers [1999]. 4.3.1

[Biller *et al.*, 2005] Ofer Biller, Michael Elhadad, and Yael Netzer. Interactive authoring of logical forms for multilingual generation. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, pages 24–31, Aberdeen, Scotland, 2005. 4.3.1

[Bird *et al.*, 2006] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. Mining email social networks. In *Proceedings of the Third International Workshop on Mining Software Repositories (MSR 06)*, Shanghai, China, 2006. 2.1

[Black and Wilensky, 1979] John B. Black and Robert Wilensky. An evaluation of story grammars. *Cognitive Science*, 3:213–230, 1979. 3.2.2

[Blei and Lafferty, 2006] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120, Pittsburgh, Pennsylvania, 2006. 6.2.1

[Booth, 1961] Wayne C. Booth. *The Rhetoric of Fiction*. University of Chicago Press, Chicago, 1961. 4

[Booth, 1989] Wayne C. Booth. *The Company We Keep: An Ethics of Fiction*. University of California Press, 1989. 1

[Bratman, 1987] M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, Massachusetts, 1987. 3.2.3

[Bremond, 1970] Claude Bremond. Morphology of the french folktale. *Semiotica*, 2(3):247–276, 1970. 3.2.2, 5

[Bremond, 1980] Claude Bremond. The logic of narrative possibilities. *New Literary History*, 11(3):387–411, 1980. Trans. Elaine D. Cancalon. Originally published 1966. 3.2.2

[Brewer and Ohtsuka, 1988] William Brewer and Keisuke Ohtsuka. Story structure, characterization, just world organization, and reader affect in american and hungarian short stories. *Poetics*, 17:398–415, 1988. 6.2.2

[Bronzwaer, 1981] W. Bronzwaer. Mieke bal's concept of focalization: A critical note. *Poetics Today*, 2(2):193–201, 1981. 3.3.1

[Brown *et al.*, 1991] Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 47 Annual Meeting of the ACL*, pages 169–176, 1991. 5.2.1

[Bruner, 1986] Jerome Bruner. *Actual Minds, Possible Worlds*. Harvard University Press, Cambridge, MA, 1986. 1, 3.2.1

[Bruner, 1991] Jerome Bruner. The narrative construction of reality. *Critical Inquiry*, 18:1–21, 1991. 1

[Budanitsky and Hirst, 2001] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL 2001 Workshop on WordNet and other lexical resources*, pages 29–34, Pittsburgh, PA, 2001. 5.2.1

[Bundgaard, 2007] Peer F. Bundgaard. The cognitive import of the narrative schema. *Semiotica*, 165(1–4):247–261, 2007. 3.2.1

[Burrows, 2004] John Burrows. Textual analysis. In Susan Schreibman, Ray Siemens, and John Unsworth, editors, *A Companion to Digital Humanities*. Blackwell, Oxford, 2004. 2.1

[Busetta *et al.*, 2003] Paolo Busetta, James Bailey, and Ramamohanarao Kotagiri. A reliable computational model for bdi agents. In *Proceedings of Workshop on Safe Agents, International Conference on Autonomous Agents & Multi-Agent Systems (AAMAS '03), Melbourne, Australia*, 2003. 3.2.3

[Callaway and Lester, 2002] Charles Callaway and James Lester. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252, 2002. 3.2.3, 4.4.2

[Campbell, 1949] Joseph Campbell. *The Hero with a Thousand Faces*. Bollingen Foundation Inc., New York, 1949. 3.1

[Carberry, 2001] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11:31–48, 2001. 6.2.2

[Carlson *et al.*, 2001] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialog*, Aalborg, Denmark, 2001. ACL. 4.3.1

[Carlson *et al.*, 2003] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In Jan van Kuppevelt and Ronnie Smith, editors, *Current Directions in Discourse and Dialogue*, pages 85–112. Kluwer Academic Publishers, 2003. 3.1, 3.2.2

[Carson, 2006] Thomas L. Carson. The definition of lying. *Noûs*, 40:284–306, 2006. B.5.1

[Chambers and Jurafsky, 2008a] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics (ACL-08)*, pages 789–797, Columbus, Ohio, 2008. 2.1, 3.1, 3.2.3, 6

[Chambers and Jurafsky, 2008b] Nathanael Chambers and Daniel Jurafsky. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 08)*, pages 698–706, Honolulu, Hawaii, 2008. Association for Computational Linguistics. 6.2.2

[Chang *et al.*, 2009] Jonathan Chang, Jordan Boyd-Graber, and David M. Blei. Connections between the lines: Augmenting social networks with text. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009. 6.2.1

[Charniak, 1972] Eugene Charniak. Toward a model of children's story comprehension. Technical Report AITR-266, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1972. 3.2.3

[Chen and Fahlman, 2008] Wei Chen and Scott E. Fahlman. Modeling mental contexts and their interactions. In *Proceedings of the AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures*, Arlington, Virginia, 2008. 3.2.2, 3.2.3

[Chen *et al.*, 2002] John Chen, Srinivas Bangalore, Owen Rambow, and Marilyn Walker. Towards automatic generation of natural language generation systems. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002. 4.4.2

[Chen, 2009] Wei Chen. Understanding mental states in natural language. In *Proceedings of The Eighth International Workshop on Computational Semantics (IWCS-8)*, Tilburg, The Netherlands, 2009. 6.2.2

[Cheong and Young, 2006] Yun-Gyung Cheong and R. Michael Young. A computational model of narrative generation for suspense. In *Proceedings of the AAAI 2006 Computational Aesthetic Workshop*, Boston, MA, 2006. 6.2.2

[Chklovski and Pantel, 2004] Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain, 2004. 5.2.1

[Cho and Fowler, 2010] Wendy K. Tam Cho and James H. Fowler. Legislative success in a small world: Social network analysis and the dynamics of congressional legislation. *The Journal of Politics*, 72(1):124–135, 2010. 2.1

[Christian and Young, 2003] David Christian and R. Michael Young. Comparing cognitive and computational models of narrative structure. liquid narrative technical report tr03-001. Technical report, Liquid Narrative Group, Department of Computer Science, North Carolina State University, Raleigh, NC, 2003. 3.1

[Clark *et al.*, 2001] Peter Clark, John A. Thompson, Ken Barker, Bruce W. Porter, Vinay K. Chaudhri, Andrés Rodríguez, Jérôme Thoméré, Sunil Mishra, Yolanda Gil, Patrick J. Hayes, and Thomas Reichherzer. Knowledge entry as the graphical assembly of components. In *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001)*, pages 22–29, Victoria, British Columbia, Canada, 2001. 4.3.1

[Cohen and Levesque, 1990] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990. 3.2.3

[Cohen, 1960] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. 2.6

[Cohen, 1995] William W. Cohen. Fast effective rule induction. In *Machine Learning: Proceedings of the Twelfth International Conference*, pages 115–123, 1995. 2.4

[Comrie, 1976] Bernard Comrie. *Aspect*. Cambridge University Press, 1976. 3.3.1

[Comrie, 1985] Bernard Comrie. *Tense*. Cambridge University Press, 1985. 3.3.1, 4.4.5

[Cook *et al.*, 1984] Malcolm E. Cook, Wendy G. Lehnert, and David D. McDonald. Conveying implicit content in narrative summaries. In *Proceedings of the 10th International Conference on Computational Linguistics (COLING '84)*, pages 5–7, 1984. 3.2.3

[Crouch and Pulman, 1993] Richard Crouch and Stephen Pulman. Time and modality in a natural language interface to a planning system. *Artificial Intelligence*, pages 265–304, 1993. 4.4.6

[Csibra *et al.*, 1999] Gergely Csibra, György Gergely, Szilvia Bíró, Orsolya Koós, and Margaret Brockbank. Goal attribution without agency cues: the perception of 'pure reason' in infancy. *Cognition*, 72:237–267, 1999. 3.2.1

[Cullingford, 1981] M. Cullingford. Sam. In R. Schank and C. Riesbeck, editors, *Inside Computer Understanding: Five Programs Plus Miniatures*, pages 75–135. Erlbaum, Hillsdale, New Jersey, 1981. 3.2.3

[Damiano and Lombardo, 2009] Rossana Damiano and Vincenzo Lombardo. Value-driven characters for storytelling and drama. In *AI*IA 2009: Emergent Perspectives in Artificial Intelligence, Lecture Notes in Computer Science, Volume 5883*, pages 436–445. Springer, 2009. 3.2.3

[Danlos, 1999] Laurence Danlos. Event coreference between two sentences. In *Proceedings of the Third International Workshop on Computational Semantics (IWCS '99)*, Tilburg, The Netherlands, 1999. 5.2

[Davis *et al.*, 2003] Peter T. Davis, David K. Elson, and Judith L. Klavans. Methods for precise named entity matching in digital collections. In *Proceedings of the Third ACM/IEEE Joint Conference on Digital Libraries (JCDL '03)*, Houston, Texas, 2003. 2.4

[Dennett, 1991] Daniel C. Dennett. The origins of selves. In Daniel Kolak and R. Martin, editors, *Self & Identity: Contemporary Philosophical Issues*. Macmillan, 1991. 1

[Díaz-Agudo *et al.*, 2004] Belén Díaz-Agudo, Pablo Gervás, and Federico Peinado. A case based reasoning approach to story plot generation. In *Proceedings of the European Conference on Case-Based Reasoning (ECCBR '04)*, Madrid, Spain, 2004. Springer-Verlag LNCS/LNAI. 3.2.2

[Dik and Aarts, 2007] Giel Dik and Henk Aarts. Behavioral cues to others' motivation and goal pursuits: The perception of effort facilitates goal inference and contagion. *Journal of Experimental Social Psychology*, 43:727–737, 2007. 3.2.1

[Doddington *et al.*, 2004] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 837–840, Lisbon, 2004. 2.1, 2.4

[Dolan and Dyer, 1985] Charles Dolan and Michael Dyer. Learning planning heuristics through observation. In *Proceedings of IJCAI 85*, pages 600–602, 1985. 5.1

[Dorr and Gaasterland, 1995] Bonnie J. Dorr and Terry Gaasterland. Selecting tense, aspect, and connecting words in language generation. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, 1995. 4.4.2, 4.4.4

[Dorr and Gaasterland, 2002] Bonnie J. Dorr and Terry Gaasterland. Constraints on the generation of tense, aspect, and connecting words from temporal expressions. *Journal of Artificial Intelligence Research*, 1:1–47, 2002. 4.4.2

[Dowty, 1979] David R. Dowty. *Word Meaning and Montague Grammar*. D. Reidel, Dordrecht, 1979. 4.4.2, 4.4.3, 4.4.3, 4.4.4, 4.4.5

[Eagleton, 2005] Terry Eagleton. *The English Novel: An Introduction*. Blackwell, Oxford, 2005. 2.2

[Eco, 1995] Umberto Eco. *Six Walks in the Fictional Woods*. Harvard University Press, reprint edition edition, 1995. 2, 3.2.1, 3.3.1

[Egidi and Gerrig, 2006] Giovanna Egidi and Richard J. Gerrig. Readers' experiences of characters' goals and actions. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 32(6):1322–1329, 2006. 3.2.1

[Elhadad and Robin, 1996] Michael Elhadad and Jacques Robin. An overview of surge: a reusable comprehensive syntactic realization component. In *INLG '96 Demonstrations and Posters*, pages 1–4, Brighton, UK, 1996. Eighth International Natural Language Generation Workshop. 4.4.2

[Elson and McKeown, 2009] David K. Elson and Kathleen R. McKeown. Extending and evaluating a platform for story understanding. In *Proceedings of the AAAI 2009 Spring Symposium on Intelligent Narrative Technologies II*, Stanford, CA, 2009. 4.2

[Elson and McKeown, 2010] David K. Elson and Kathleen R. McKeown. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, Atlanta, Georgia, 2010. 1

[Elson *et al.*, 2010] David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, Uppsala, Sweden, 2010. 1

[Fayzullin *et al.*, 2007] Marat Fayzullin, V.S. Subrahmanian, Massimiliano Albanese, Carmine Cesarano, and Antonio Picariello. Story creation from heterogeneous data sources. *Multimedia Tools and Applications*, 33(3):351–377, 2007. 6.2.2

[Fellbaum, 1998] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998. 2.4, 4.2, 5.2, 6.1

[Fernando and Stevenson, 2008] Samuel Fernando and Mark Stevenson. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special-interest group for Computational Lingustics*, Oxford, England, 2008. 5.2

[Finkel *et al.*, 2005] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005. 2.4

[Finlayson, 2009] Mark Alan Finlayson. Deriving narrative morphologies via analogical story merging. In B. Kokinov, K. Holyoak, and D. Gentner, editors, *New Frontiers in Analogy Research*. NBU Press, Sofia, 2009. 5.3.2

[Fitzgerald *et al.*, 2009] Adam Fitzgerald, Gurlal Kahlon, and Mark O. Riedl. A computational model of emotional response to stories. In *Proceedings of the 2nd International Conference on Interactive Digital Storytelling*, Guimarães, Portugal, 2009. 6.2.2

[Forster, 1990] E. M. Forster. *Aspects of the Novel*. Penguin, Harmondsworth, UK, 1990. Originally published 1927. 3.1

[French, 2002] Robert M. French. The computational modeling of analogy-making. *Trends in Cognitive Sciences*, 6(5):200–205, 2002. 5.3.2

[Fulton *et al.*, 2005] Helen Fulton, Rosemary Huisman, Julian Murphet, and Anne Dunn, editors. *Narrative and Media*. Cambridge University Press, 2005. 1

[Fum *et al.*, 1991] Danilo Fum, Carlo Tasso, L. Tiepolo, and A. Tramontini. A computational model of tense selection and its experimentation within an intellegent tutor. In Edoardo Ardizzone, Salvatore Gaglio, and Filippo Sorbello, editors, *Proceedings of the 2nd Congress of the Italian Association for Artificial Intelligence (AI*IA)*, pages 261–270, Palermo, Italy, 1991. Springer. 4.4.2

[Gagnon and Lapalme, 1996] Michel Gagnon and Guy Lapalme. From conceptual time to linguistic time. *Computational Linguistics*, 22(1):91–127, 1996. 4.4.2

[Gagnon *et al.*, 2006] Michel Gagnon, Eliana de Mattos Pinto Coelho, and Roger Antonio Finger. Towards a formalization of tense and aspect for the generation of portuguese sentences. In *Proceedings of PROPOR '06*, pages 229–232, Itatiaia, Brazil,, 2006. 4.4.4

[Gale and Church, 1993] William A Gale and Kenneth W Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993. 5.2.1

[Gansner and North, 2000] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software - Practice and Experience (SPE)*, 30(11):1203–1233, 2000. 2.6

[Gedigian *et al.*, 2006] Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ciric. Catching metaphors. In *Proceedings of the 3rd Worskhop on Scalable Natural Language Understanding, HLT/NAACL '06*, pages 41–48, New York, NY, 2006. 4.3.4

[Genette, 1972] Gérard Genette. *Figures III*. Seuil, Paris, 1972. 3.2.1, 4

[Genette, 1983] Gérard Genette. *Nouveau discours du récit*. Seuil, Paris, 1983. 3.1, 3.3.1

[Gergely *et al.*, 1995] György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. *Cognition*, 56:165–193, 1995. 3.2.1

[Gerrig and Bernardo, 1994] Richard Gerrig and Allan B.I. Bernardo. Readers as problem-solvers in the experience of suspense. *Poetics*, 22:459–472, 1994. 6.2.2

[Gerrig and Egidi, 2010] Richard Gerrig and Giovanna Egidi. The bushwhacked piano and the bushwhacked reader: The willing construction of disbelief. *Style*, 44(1-2):189–206, 2010. 3.2.1

[Gervás *et al.*, 2005] Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. Story plot generation based on cbr. *Knowledge-Based Systems*, 4:235–242, 2005. 3.2.2

[Glass and Bangay, 2007] Kevin Glass and Shaun Bangay. A naïve, salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA '07)*, pages 1–6, Pietermaritzburg, South Africa, 2007. 2.5.1

[Goldenberg and Zheng, 2007] Anna Goldenberg and Alice Zheng. Exploratory study of a new model for evolving networks. In Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, Anna Goldenberg, Eric P. Xing, and Alice X. Zheng, editors, *Statistical Network Analysis: Models, Issues and New Directions*. Springer Berlin / Heidelberg, 2007. 6.2.1

[Gordon and Swanson, 2009] Andrew S. Gordon and Reid Swanson. Identifying personal stories in millions of weblog entries. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, San Jose, California, 2009. 3.1

[Goyal *et al.*, 2010a] Amit Goyal, Ellen Riloff, and Hal Daumé III. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, Cambridge, Massachusetts, 2010. 3.2.3, 5.1, 6.2.2

[Goyal *et al.*, 2010b] Amit Goyal, Ellen Riloff, Hal Daumé III, and Nathan Gilbert. Toward plot units: Automatic affect state analysis. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles, California, 2010. 6.2.2

[Graesser and Clark, 1985] Arthur C. Graesser and Leslie F. Clark. *Structures and Procedures of Implicit Knowledge*. Ablex, Norwood, New Jersey, 1985. 3.2.1

[Graesser *et al.*, 1991a] Arthur Graesser, Jonathan M. Golding, and Debra L. Long. Narrative representation and comprehension. In Rebecca Barr, Michael L. Kamil, and Peter B. Mosenthal, editors, *Handbook of reading research*, volume 2, pages 171–205. Longman, New York, 1991. 3.2.1

[Graesser *et al.*, 1991b] Arthur C. Graesser, Kathy Lang, and Richard Roberts. Question answering in the context of stories. *Journal of Experimental Psychology: General*, 120:254–277, 1991. 3.2.1

[Graesser *et al.*, 1994] Arthur C. Graesser, Murray Singer, and Tom Trabasso. Constructing inferences during narrative text comprehension. *Psychological Review*, 101(3):371–395, 1994. 3.1, 3.2.1

[Graesser *et al.*, 1997] Arthur C. Graesser, Keith K. Millis, and Rolf A. Zwaan. Discourse comprehension. *Annual Review of Psychology*, 48:163–189, 1997. 3.2.1

[Graesser *et al.*, 2001] Arthur C. Graesser, Peter Wiemer-Hastings, and Katja Wiemer-Hastings. Constructing inferences and relations during text comprehension. In Ted Sanders, Joost Schilperoord, and Wilbert Spooren, editors, *Text representation: linguistic and psycholinguistic aspects*, pages 249–272. John Benjamins, 2001. 3.2.1

[Green *et al.*, 2004] Melanie C. Green, Timothy C. Brock, and Geoff F. Kaufman. Understanding media enjoyment: The role of transportation into narrative worlds. *Communication Theory*, 14(4):311–327, 2004. 1

[Grishman *et al.*, 2005] Ralph Grishman, David Westbrook, and Adam Meyers. Nyu's english ace 2005 system description. In *ACE 05 Evaluation Workshop*, Gaithersburg, MD, 2005. 2.4, 3.2.2

[Grosz and Sidner, 1986] Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Linguistics*, 12(3):175–204, 1986. 3.2.2

[Grosz *et al.*, 1995] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226, 1995. 3.2.2

[Grote, 1998] Brigitte Grote. Representing temporal discourse markers for generation purposes. In *Proceedings of the Discourse Relations and Discourse Markers Workshop*, pages 22–28, Montreal, Canada, 1998. 4.4.2

[Gruzd and Haythornthwaite, 2008] Anatoliy Gruzd and Caroline Haythornthwaite. Automated discovery and analysis of social networks from threaded discussions. In *International Network of Social Network Analysis (INSNA) Conference*, St. Pete Beach, Florida, 2008. 2.1

[Habash and Dorr, 2003] Nizar Habash and Bonnie Dorr. A categorial variation database for english. In *Proceedings of the North American Association for Computational Linguistics (NAACL)*, pages 96–102, Edmonton, Canada, 2003. 5.2.1

[Hall *et al.*, 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorations*, 11(1), 2009. 2.5.6

[Halliday, 1976] M.A.K. Halliday. The english verbal group. In G. R. Kress, editor, *Halliday: System and Function in Language*. Oxford University Press, London, 1976. (document), 3.3.1, 4.4.6, 4.21

[Halpin *et al.*, 2004] Harry Halpin, Johanna D. Moore, and Judy Robertson. Automatic analysis of plot for story rewriting. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '04)*, Barcelona, 2004. 2.1, 3.2.2, 5.2

[Halpin, 2003] Harry Halpin. The plots of children and machines: The statistical and symbolic semantic analysis of narratives. Master's thesis, University of Edinburgh, Edinburgh, UK, 2003. 5.2

[Harabagiu *et al.*, 1995] Sanda Harabagiu, Cosmin Adrian Bejan, and Paul Morarescu. Shallow semantics for relation extraction. In *Proceedings of the Nineteenth International*

Joint Conference on Artificial Intelligence (IJCAI '05), pages 1061–1066, Edinburgh, UK, 1995. 6.2.2

[Harper and Charniak, 1986] Mary P. Harper and Eugene Charniak. Time and tense in english. In *Proceedings of ACL '86*, New York City, 1986. 4.4.2

[Hassin *et al.*, 2005] Ran R. Hassin, Henk Aarts, and Melissa J. Ferguson. Automatic goal inferences. *Journal of Experimental Social Psychology*, 41:129–140, 2005. 3.2.1

[Hatzivassiloglou *et al.*, 2001] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization*, Pittsburgh, Pennsylvania, 2001. 5

[Heider and Simmel, 1944] Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *The American Journal of Psychology*, 57(2):243–259, 1944. 3.2.1

[Hidi and Baird, 1986] Suzanne Hidi and William Baird. Interestingness— a neglected variable in discourse processing. *Cognitive Science*, 10:179–194, 1986. 3.1

[Hinrichs, 1987] Erhard W. Hinrichs. A compositional semantics of temporal expressions in english. In *Proceedings of the 25th Annual Conference of the Association for Computational Linguistics (ACL-87)*, Stanford, CA, 1987. 3.3.1, 4.4.2

[Hirschberg and Nakatani, 1996] Julia Hirschberg and Christine H. Nakatani. A prosodic analysis of discourse segments in direction-giving monologues. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL '96)*, pages 286–293, Santa Cruz, California, 1996. 3.2.2

[Hobbs and Gordon, 2005] Jerry R. Hobbs and Andrew S. Gordon. Encoding knowledge of commonsense psychology. In *Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning*, pages 107–114, Corfu, Greece, 2005. 3.1, 3.2.3

[Hobbs and Gordon, 2010] Jerry R. Hobbs and Andrew Gordon. Goals in a formal theory of commonsense psychology. In *Proceedings of the 6thInternational Conference on Formal Ontology in Information Systems (FOIS-2010)*, Toronto, Canada, 2010. 3.2.3

[Hobbs, 1985] Jerry R. Hobbs. On the coherence and structure of discourse. Technical Report Technical Report 85-37, Center for the Study of Language and Information (CSLI), Stanford, California, 1985. 3.2.2

[Holyoak and Thagard, 1989] Keith J. Holyoak and Paul Thagard. Analogical mapping by constraint satisfaction. *Cognitive Science*, 13:295–355, 1989. 5.3.2

[Hornstein, 1990] Norbert Hornstein. *As Time Goes By: Tense and Universal Grammar*. MIT Press, Cambridge, MA, 1990. 3.3.1, 4.4.6, 4.4.6

[Huisman and Snijders, 2003] Mark Huisman and Tom A.B. Snijders. Statistical analysis of longitudinal network data with changing composition. *Sociological Methods and ResearchSociological Methods and Research*, 32:253–287, 2003. 6.2.1

[Jacquemin, 1997] Christian Jacquemin. Guessing morphology from terms and corpora. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 156–165, 1997. 5.2.1

[Jhala, 2004] Arnav Jhala. An intelligent camera planning system for dynamic narratives. Master's thesis, North Carolina State University, 2004. 3.2.3

[Johnson and Fillmore, 2000] Christopher R. Johnson and Charles J. Fillmore. The framenet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, pages 56–62, Seattle, Washington, 2000. 3.2.3

[Jones and Nisbett, 1972] Edward E. Jones and Richard E. Nisbett. The actor and the observer: Divergent perceptions of the causes of behavior. In Edward E. Jones, David E. Kanhouse, Harold H. Kelley, Richard E. Nisbett, Stuart Valins, and Bernard Weiner, editors, *Attribution: Perceiving the causes of behavior*, pages 79–94. General Learning, Morristown, New Jersey, 1972. 3.2.2

[Jones, 1912] V. S. Vernon Jones. *Aesop's Fables: A New Translation.* Avenel Books, New York, 1912. 1, D

[Kingsbury and Palmer, 2002] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-02)*, Canary Islands, Spain, 2002. 2, 3.1, 4.2, 6.3

[Kintsch, 1980] Walter Kintsch. Learning from text, levels of comprehension, or: Why anyone would read a story anyway. *Poetics*, 9:87–98, 1980. 3.2.1

[Kintsch, 1988] Walter Kintsch. The role of knowledge in discourse comprehension: a constructive-integration model. *Psychological Review*, 95:163–182, 1988. 3.2.1

[Kipper *et al.*, 2006] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. Extensive classifications of english verbs. In *Proceedings of the 12th EURALEX International Congress*, Turin, Italy, 2006. 4.2, 5.2, 6.1

[Kittler *et al.*, 1998] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998. 2.5.6

[Knublauch *et al.*, 2004] Holger Knublauch, Ray W. Fergerson nd Natalya F. Noy, and Mark A. Musen. The protégé owl plugin: An open development environment for semantic web applications. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, 2004. 4.3.1

[Konolige and Pollack, 1989] Kurt Konolige and Martha E. Pollack. Ascribing plans to agents: Preliminary report. In *Proceedings of the Eleventh International Joint Conferenceon Artificial Intelligence*, pages 924–930, Detroit, Michigan, 1989. 3.2.3

[Kossinets and Watts, 2006] Gueorgi Kossinets and Duncan J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006. 6.2.1

[Kucera and Francis, 1967] Henry Kucera and W. Nelson Francis. *Computational Analysis of Present-Day American English.* Brown University Press, Providence, RI, 1967. 5.2.1

[Kurlander *et al.*, 1996] David Kurlander, Tim Skelly, and David Salesin. Comic chat. In *Proceedings of SIGGRAPH '96*, New Orleans, LA, 1996. 2.5

[Labov and Waletzky, 1967] William Labov and Joshua Waletzky. Narrative analysis: Oral versions of personal experience. In J. Helm, editor, *Essays on the Verbal and Visual Arts*, pages 12–44. University of Washington Press, 1967. 3.2.2

[Labov, 1972] William Labov. The transformation of experience in narrative syntax. In *Language in the Inner City*, pages 354–396. University of Pennsylvania Press, Philadelphia, 1972. 1

[Lang, 1999] R. Raymond Lang. A declarative model for simple narratives. In Mateas and Sengers [1999]. 3.2.2

[Langston and Trabasso, 1999] Mark Langston and Tom Trabasso. Modeling causal integration and availability of information during comprehension of narrative texts. In *The construction of mental representations during reading*, pages 25–60. Erlbaum, Mahwah, New Jersey, 1999. 3.2.1

[Lapata and Lascarides, 2004] Mirella Lapata and Alex Lascarides. Inferring sentence-internal temporal relations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 153–160, Barcelona, Spain, 2004. 6.2.2

[Lapata and Lascarides, 2006] Mirella Lapata and Alex Lascarides. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27:85–117, 2006. 4.4.2

[Lavoie and Rambow, 1997] Benoit Lavoie and Owen Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC, 1997. 4.4.2

[Lee *et al.*, 2008] Alan Lee, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. Departures from tree structures in discourse: Shared arguments in the penn discourse treebank. In

Proceedings of the Constraints in Discourse III Workshop, Potsdam, Germany, 2008. 3.2.2

[Lee, 2007] John Lee. A computational model of text reuse in ancient literary texts. In *In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 472–479, Prague, 2007. 2.1

[Lehnert *et al.*, 1983] Wendy G. Lehnert, Michael G. Dyer, Peter N. Johnson, C.J. Yang, and Steve Harley. Boris- an experiment in in-depth understanding of narratives. *Artificial Intelligence*, 20:15–62, 1983. 3.2.3, 3.2.3, 6.2.2

[Lehnert, 1981] Wendy G Lehnert. Plot units and narrative summarization. *Cognitive Science: A Multidisciplinary Journal*, 5(4):293–331, 1981. (document), 3.2.3, 3.4, 6.2.2, B

[Lehnert, 1994] Wendy Lehnert. Cognition, computers, and car bombs: How yale prepared me for the 1990s. In Roger C. Schank and Ellen Langer, editors, *Beliefs, reasoning, and decision making*, pages 143–173. Erlbaum, Hillsdale, New Jersey, 1994. 3.2.3

[Lepage and Denoual, 2005] Yves Lepage and Etienne Denoual. Automatic generation of paraphrases to be used as translation references in objective evaluation measures of machine translation. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05) International Workshop on Paraphrasing (IWP 2005)*, pages 57–64, Jeju, Korea, 2005. 5.2

[Levi-Strauss, 1968] Claude Levi-Strauss. *Structural Anthropology*, chapter The Structural Study of Myth, pages 202–212. Basic Books, New York, 1968. 3.2.2

[Lichtenstein and Brewer, 1980] Edward H. Lichtenstein and William F. Brewer. Memory for goal-directed events. *Cognitive Psychology*, 12(3):412–445, 1980. 3.2.1

[Ligozat and Zock, 1992] Gerard Ligozat and Michael Zock. How to visualize time, tense and aspect? In *Proceedings of the 14th International Conference on Computational Linguistics (COLING '92)*, pages 475–482, Nantes, France, 1992. 4.4.2

[Lin and Pantel, 2001] Dekang Lin and Patrick Pantel. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360, 2001. 5.2

[Lin *et al.*, 2010] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. A pdtb-styled end-to-end discourse parser. Technical Report Technical Report TRB8/10, School of Computing, National University of Singapore, 2010. 3.1, 3.2.2, 6.2.2

[Lin, 1998] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, Madison, WI, 1998. 5.2.1

[Louis and Nenkova, 2010] Annie Louis and Ani Nenkova. Creating local coherence: An empirical assessment. In *Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT '10)*, Los Angeles, California, 2010. 3.2.2

[Löwe and Pacuit, 2008] Benedikt Löwe and Eric Pacuit. An abstract approach to reasoning about games with mistaken and changing beliefs. *Australasian Journal of Logic*, 6:162–181, 2008. 3.2.3

[Löwe *et al.*, 2009] Benedikt Löwe, Eric Pacuit, and Sanchit Saraf. Identifying the structure of a narrative via an agent-based logic of preferences and beliefs: Formalizations of episodes from csi: Crime scene investigation. In Michael Duvigneau and Daniel Moldt, editors, *Proceedings of the Fifth International Workshop on Modelling of Objects, Components, and Agents*, pages 45–63, 2009. 3.2.3

[Lynch and van den Broek, 2007] Julie S. Lynch and Paul van den Broek. Understanding the glue of narrative structure: Children's on- and off-line inferences about characters' goals. *Cognitive Development*, 22:323–340, 2007. 3.2.1

[Macleod *et al.*, 1994] Catherine Macleod, Ralph Grishman, and Adam Meyers. Creating a common syntactic dictionary of english. In *Proceedings of SNLR: International Workshop on Sharable Natural Language Resources*, Nara, Japan, 1994. 4.4.1

[Magliano and Radvansky, 2001] Joseph P. Magliano and Gabriel A. Radvansky. Goal coordination in narrative comprehension. *Psychonomic Bulletin & Review*, 8(2):372–376, 2001. 3.2.1

[Magliano *et al.*, 2005] Joseph P. Magliano, Holly A. Taylor, and Hyun-Jeong Joyce Kim. When goals collide: Monitoring the goals of multiple characters. *Memory & Cognition*, 33(8):1357–1367, 2005. 3.2.1

[Mahon, 2008] James Edwin Mahon. The definition of lying and deception. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University, fall 2008 edition, 2008. B.5.1

[Mamede and Chaleira, 2004] Nuno Mamede and Pedro Chaleira. Character identification in children stories. In *EsTAL 2004 - Advances in Natural Language Processing, LNCS*, pages 82–90, Berlin Heidelberg, 2004. Springer. 2.5.1

[Mandler and Johnson, 1977] Jean M. Mandler and Nancy S. Johnson. Remembrance of things parsed: Story structure and recall. *Cognitive Psychology*, 9(1):111–151, 1977. (document), 3.2, 3.2.2, 3.2.2

[Mandler and Johnson, 1980] Jean M. Mandler and Nancy S. Johnson. On throwing out the baby with the bathwater: A reply to black and wilensky's evaluation of story grammars. *Cognitive Science*, 4:305–312, 1980. 3.2.2

[Mani and Pustejovsky, 2004] Inderjeet Mani and James Pustejovsky. Temporal discourse models for narrative structure. In *Proceedings of the ACL Workshop on Discourse Annotation*, Barcelona, Spain, 2004. 3.1, 3.3.1, 4.4.6, 5.2

[Mani and Wellner, 2006] Inderjeet Mani and Ben Wellner. A pilot study on acquiring metric temporal constraints for events. In *Proceedings of the ACL 2006 Workshop on Annotating and Reasoning about Time and Events*, pages 23–29, Sydney, Australia, 2006. 6.2.2

[Mani *et al.*, 2005] Inderjeet Mani, James Pustejovsky, and Rob Gaizauskas, editors. *The Language of Time: A Reader*. Oxford University Press, Oxford, 2005. 3.3.1

[Mani *et al.*, 2006] Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. Machine learning of temporal relations. In *Proceedings of COLING/ACL 2006*, pages 753–760, Sydney, Australia, 2006. 4.4.2

[Mani, 2004] Inderjeet Mani. Recent developments in temporal information extraction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP '03)*, pages 45–60, Borovets, Bulgaria, 2004. 4.4.2

[Mani, 2010] Inderjeet Mani. *The Imagined Moment: Time, Narrative, and Computation*. University of Nebraska Press, 2010. 3.3.1

[Mann and Thompson, 1988] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988. 1, 3.2.2, 6

[Mann, 1983] William C. Mann. An overview of the nigel text generation grammar. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 74–78, Cambridge, Massachusetts, 1983. 4.4.2

[Marcu, 1997] Daniel Marcu. The rhetorical parsing of natural language texts. In *Proceedings of ACL '97 / EACL '97*, pages 96–103, Madrid, Spain, 1997. 3.2.2

[Marcus *et al.*, 1993] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2), 1993. 3.1, 4.3.1

[Maslow, 1943] A. H. Maslow. A theory of human motivation. *Psychological Review*, 50:370–396, 1943. 3.3.2.8

[Mateas and Sengers, 1999] Michael Mateas and Phoebe Sengers, editors. *Narrative Intelligence: Papers from the 1999 Fall Symposium*. Technical Report FS-99-01. American Association for Artificial Intelligence, Menlo Park, California, 1999. D

[Mateas and Stern, 2003] Michael Mateas and Andrew Stern. Facade: An experiment in building a fully-realized interactive drama. In *Game Developer's Conference: Game Design Track*, San Jose, CA, 2003. 3.2.3

[Matthiessen and Bateman, 1991] Christian M. I. M. Matthiessen and John A. Bateman. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Frances Pinter Publishers and St. Martin's Press, London and New York, 1991. 4.4.2

[Max-Neef, 1992] Manfred Max-Neef. Development and human needs. In Paul Ekins and Manfred Max-Neef, editors, *Real-life economics: Understanding wealth creations*, pages 197–214. Routledge, London, 1992. 3.3.2.8

[McCallum *et al.*, 2007] Andrew McCallum, Xuerui Wang, and Andres Corrada-Emmanual. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30:249–272, 2007. 2.1

[McCarthy and Hayes, 1969] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969. 3.3.2.2, B.3

[McCarthy *et al.*, 2006] Philip M. McCarthy, Arthur C. Graesser, and Danielle S. McNamara. Distinguishing genre using coh-metrix indices of cohesion. In *Poster Proceedings at the 16th Annual Meeting of the Society for Text and Discourse*, 2006. 3.2.2

[McKee, 1997] Robert McKee. *Story*. Harper Collins, New York City, 1997. 3.2.2, B.1, B.3

[McKeown, 1985] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, England, 1985. 3.2.2

[McKoon and Ratcliff, 1992] Gail McKoon and Roger Ratcliff. Inference during reading. *Psychological Review*, 99(3):440–466, 1992. 3.2.1

[Michel *et al.*, 2011] Jean-Baptiste Michel, Yuan K. Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011. 2.1

[Minsky, 1975] Marvin Minsky. A framework for representing knowledge. In Patrick Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York City, 1975. 3.2.3, 4.1

[Moens and Steedman, 1988] Marc Moens and Mark Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28, 1988. 4.4.4

[Montfort, 2007] Nick Montfort. Ordering events in interactive fiction narratives. In *Intelligent Narrative Technologies: Papers from the 2007 AAAI Fall Symposium*, pages 87–94, Arlington, Virginia, 2007. AAAI Press. 6.2.2

[Moore and Pollack, 1992] Johanna D. Moore and Martha E. Pollack. A problem for rst: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4):537–544, 1992. 3.2.2

[Moretti, 1999] Franco Moretti. *Atlas of the European Novel, 1800-1900*. Verso, London, 1999. 2.2

[Moretti, 2000a] Franco Moretti. Conjectures on world literature. *New Left Review*, 1:54–68, 2000. 1, 2

[Moretti, 2000b] Franco Moretti. The slaughterhouse of literature. *Modern Language Quarterly*, 61(1):207–227, 2000. 2.2

[Moretti, 2005] Franco Moretti. *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso, London, 2005. 2.1

[Mostellar and Wallace, 1984] Frederick Mostellar and David L. Wallace. *Applied Bayesian and Classical Inference: The Case of The Federalist Papers*. Springer, New York, 1984. 1, 2.1

[Mott and Lester, 2006] Bradford Mott and James Lester. U-director: A decision-theoretic narrative planning architecture for storytelling environments. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, Hakodate, Japan, 2006. 3.2.3

[Mott *et al.*, 2006] Bradford Mott, Sunyoung Lee, and James Lester. Probabilistic goal recognition in interactive narrative environments. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-2006)*, Boston, 2006. 6.2.2

[Mueller, 2003] Erik T. Mueller. Story understanding through multi-representation model construction. In Graeme Hirst and Sergei Nirenburg, editors, *Text Meaning: Proceedings of the HLT-NAACL 2003 Workshop*, pages 46–53, East Stroudsburg, PA, 2003. Association for Computational Linguistics. 3.2.1

[Mueller, 2004] Erik T. Mueller. Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research*, 5(4):307–340, 2004. 3.1, 3.2.3

[Mueller, 2006] Erik T. Mueller. Modelling space and time in narratives about restaurants. *Literary and Linguistic Computing*, 4 2006. 3.1, 3.2.3

[Nackoul, 2010] David Douglas Nackoul. Text to text: Plot unit searches generated from english. Master's thesis, Massachusetts Institute of Technology, 2010. 3.2.3

[Needleman and Wunsch, 1970] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. 1

[Nelson, 1987] Eric Nelson. *Effects on Memory of Discourse Coherence in Encoding*. PhD thesis, University of Chicago, 1987. 1

[Nelson, 1989] Katherine Nelson. *Narratives from the Crib*. University Press, Cambridge, MA, 1989. 1

[Nelson, 2003] Katherine Nelson. Narrative and self, myth and memory: Emergence of the cultural self. In R. Fivush and C. Haden, editors, *Autobiographical Memory and the*

Construction of the Narrative Self: Developmental and cultural perspectives. Lawrence Erlbaum Associates, Mahway, NJ, 2003. 1, 3.2.1

[Nerbonne, 1986] John Nerbonne. Reference time and time in narration. *Linguistics and Philosophy*, 9(1):83–95, 1986. 3.3.1, 4.4.2

[Nissan, 2008] Ephraim Nissan. Nested beliefs, goals, duties, and agents reasoning about their own or each other's body in the timur model: A formalism for the narrative of tamerlane and the three painters. *Journal of Intelligent and Robotic Systems*, 52:515–582, 2008. 3.2.3

[Ogihara, 1995] Toshiyuki Ogihara. Double-access sentences and reference to states. *Natural Language Semantics*, 3:177–210, 1995. 4.4.6

[Oltean, 1993] Stefan Oltean. A survey of the pragmatic and referential functions of free indirect discourse. *Poetics Today*, 14(4):691–714, 1993. 2.7.1

[Özsoyoglu and Snodgrass, 1995] Gultekin Özsoyoglu and Richard T. Snodgrass. Temporal and real-time databases: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 7(4), 1995. 3.3.1

[Paley *et al.*, 1997] Suzanne M. Paley, John D. Lowrance, and Peter D. Karp. A generic knowledge-base browser and editor. In *Proceedings of the 1997 National Conference on AI (AAAI '97)*, Providence, Rhode Island, 1997. 4.3.1

[Palmer *et al.*, 2005] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005. 4.3.1

[Palmer *et al.*, 2007] Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163, 2007. 4.3.3, 5.2.2

[Palmer, 2007] Alan Palmer. Universal minds. *Semiotica*, 165(1–4):202–225, 2007. 2, 3.2.2

[Palmer, 2010] Alan Palmer. Social minds in little dorrit. In Paula Leverage, Howard Mancing, Richard Schweickert, and Jennifer Marston William, editors, *Theory of Mind and Literature*. Purdue University Press, 2010. 3.2.2, 6.1

[Passonneau *et al.*, 2007] Rebecca Passonneau, Adam Goodkind, and Elena Levy. Annotation of children's oral narrations: Modeling emergent narrative skills for computational applications. In *Proceedings of FLAIRS-20*, Key West, Florida, 2007. 3.2.2

[Passonneau, 1988] Rebecca Passonneau. A computational model of the semantics of tense and aspect. *Computational Linguistics*, 14(2):44–60, 1988. 3.3.1, 4.4.2

[Peinado *et al.*, 2008] Federico Peinado, Marc Cavazza, and David Pizzi. Revisiting character-based affective storytelling under a narrative bdi framework. In *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling (ICIDS '08)*, Erfurt, Germany, 2008. 3.2.3

[Perry, 2007] Ben Edwin Perry. *Aesopica: A Series of Texts Relating to Aesop or Ascribed to Him*. University of Illinois Press, 2007. (document), 1, 5.1

[Polanyi and Scha, 1984] Livia Polanyi and Remko Scha. A syntactic approach to discourse semantics. In *Proceedings of the International Conference on Computational Linguistics*, pages 413–419, Stanford, California, 1984. 3.2.2

[Polanyi, 1989] Livia Polanyi. *Telling the American Story: A Structural and Cultural Analysis of Conversational Storytelling*. MIT Press, 1989. 3.2.2

[Pollack, 1990] Martha E. Pollack. Plans as complex mental attitudes. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*. MIT Press, 1990. 3.2.3

[Pouliquen *et al.*, 2007] Bruno Pouliquen, Ralf Steinberger, and Clive Best. Automatic detection of quotations in multilingual news. In *Proceedings of Recent Advances in Natural Language Processing 2007*, Borovets, Bulgaria, 2007. 2.5.1

[Pouliquen *et al.*, 2008] Bruno Pouliquen, Hristo Tanev, and Martin Atkinson. Extracting and learning social networks out of multilingual news. In *Proceedings of the social net-*

works and application tools workshop (SocNet-08), pages 13–16, Skalica, Slovakia, 2008. 2.1

[Power and Evans, 2004] Richard Power and Roger Evans. Wysiwym with wider coverage. In *Proceedings of HLT-NAACL '04*, pages 211–214, Boston, MA, 2004. 4.3.1

[Power and Scott, 1998] Richard Power and Donia Scott. Multilingual authoring using feedback texts. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (ACL '98)*, Montréal, Canada, 1998. 4.3.1

[Power *et al.*, 1998] Richard Power, Donia Scott, and Roger Evans. What you see is what you meant: direct knowledge editing with natural language feedback. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, Brighton, UK, 1998. 4.3.1

[Poynor and Morris, 2003] Daivd V. Poynor and Robin K. Morris. Inferred goals in narratives: Evidence from self-paced reading, recall and eye movements. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 29(1):3–9, 2003. 3.2.1

[Pradhan *et al.*, 2007] Sameer S. Pradhan, Eduard Hovy, Mitchell P. Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: A unified relational semantic representation. In *Proceedings of the International Conference on Semantic Computing*, pages 517–526, 2007. 4.3.1, 5.2

[Prasad *et al.*, 2008] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The penn discourse treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, 2008. 1, 3.1, 3.2.2, 5.2, 6

[Prince, 1973] Gerald Prince. *A Grammar of Stories: An Introduction.* Mouton, The Hague, 1973. 3.2.2

[Propp, 1969] Vladimir Propp. *Morphology of the Folk Tale.* University of Texas Press, second edition, 1969. Trans. Laurence Scott. Originally published 1928. 3.2.2, 5

[Pustejovsky *et al.*, 2003a] James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. Timeml: Robust specification of event and temporal expressions in text. In *Proceedings of the IWCS-5 Fifth International Workshop on Computational Semantics*, Tilburg, The Netherlands, 2003. 3.3.1

[Pustejovsky *et al.*, 2003b] James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Dragomir Radev, Andrea Setzer, and Beth Sundheim. The timebank corpus. *Proceedings of Corpus Linguistics 2003*, pages 647–656, 2003. 3.1, 3.3.1, 4.3.1

[Rambow, 1993] Owen Rambow. Rhetoric as knowledge. In *Proceedings of the ACL Workshop on Intentionality and Structure in Discourse Relations*, Columbus, Ohio, 1993. 3.2.2

[Ramírez and Geffner, 2010] Miquel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, Atlanta, Georgia, 2010. 6.2.2

[Rao and Georgeff, 1995] Anand S. Rao and Michael P. Georgeff. Bdi agents: From theory to practice. In *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312–319, San Francisco, California, 1995. 3.2.3

[Rao *et al.*, 1992] A. Rao, D. Morley, M. Selvestrel, and G. Murray. Representation, selection, and execution of team tactics in air combat modelling. In Anthony Adams and Leon Sterling, editors, *Proceedings of the 5th Australian Joint Conference on Artificial Intellience*, pages 185–190. World Scientific, 1992. 3.2.3

[Rapaport, 1986] William J Rapaport. Logical foundations for belief representation. *Cognitive Science*, 10:371–422, 1986. 3.2.3

[Ratnaparkhi, 1996] Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. In *In Proceedings of the Empirical Methods in Natural Language Processing Conference*. University of Pennsylvania, 1996. 3

[Redeker and Egg, 2006] Gisela Redeker and Markus Egg. Says who? on the treatment of speech attributions in discourse structure. In *Proceedings of Constraints in Discourse*, Maynooth, Ireland, 2006. 2.5

[Reeves, 1991] John F. Reeves. *Computational Morality: A Process Model of Belief Conflict and Resolution for Story Understanding*. PhD thesis, Computer Science Department, University of California, Los Angeles, Los Angeles, 1991. Technical Report UCLA-AI-91-05. 3.1, 3.2.3

[Reichenbach, 1947] Hans Reichenbach. *Elements of Symbolic Logic*. MacMillan, London, 1947. 3.3.1, 4.4.3

[Resnik, 1999] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999. 5.2.1

[Richards and Singer, 2001] Eric Richards and Murray Singer. Representation of complex goal structures in narrative comprehension. *Discourse Processes*, 31:111–135, 2001. 3.2.1

[Riedl and Young, 2004] Mark Riedl and R. Michael Young. An intent-driven planner for multi-agent story generation. In *Proc. of the 3rd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems (AAMAS 04)*, 2004. 3.1, 3.2.3

[Riedl *et al.*, 2003] Mark Riedl, C.J. Saretto, and R. Michael Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '03)*, pages 741–748, Melbourne, Australia, 2003. 6.2.2

[Riedl *et al.*, 2008] Mark Riedl, Jonathan P. Rowe, and David K. Elson. Toward intelligent support of authoring machinima media content: Story and visualization. In *Proceedings of the 2nd International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN '08)*, Cancun, Mexico, 2008. 3.2.1, 4.3.1

[Rubin, 1995] David C. Rubin. *Memory in Oral Traditions: The Cognitive Psychology of Epic, Ballads, and Counting-out Rhymes*. Oxford University Press, New York, 1995. 1

[Rumelhart, 1975] David Rumelhart. Notes on a schema for stories. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 231–236. New York Academic Press, Inc., 1975. 3.2.2

[Rumelhart, 1980] David E. Rumelhart. On evaluating story grammars. *Cognitive Science*, 4:313–316, 1980. 3.2.2

[Ryan, 1991] Marie-Laure Ryan. *Possible Worlds, Artificial Intelligence and Narrative Theory*. Indiana University Press, Bloomington, Indiana, 1991. 3.2.2, 3.2.3, 5.1

[Ryan, 2007] Marie-Laure Ryan. Diagramming narrative. *Semiotica*, 165(1–4):11–40, 2007. 3.2.3

[Sack, 2006] Graham Sack. Bleak house and weak social networks. Unpublished manuscript, 2006. 6.2.1

[Sadri, 2010] Fariba Sadri. Logic-based approaches to intention recognition. In Nak-Young Chong and Fulvio Mastrogiovanni, editors, *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*. Information Science Publishing, 2010. 6.2.2

[Sagot *et al.*, 2010] Benoît Sagot, Laurence Danlos, and Rosa Stern. A lexicon of french quotation verbs for automatic quotation extraction. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, Malta, 2010. 2.5.1

[Sarmento and Nunes, 2009] Luís Sarmento and Sérgio Nunes. Automatic extraction of quotes and topics from news feeds. In *4th Doctoral Symposium on Informatics Engineering*, Porto, Portugal, 2009. 2.5.1

[Schank and Abelson, 1977] Roger Schank and Robert Abelson. *Scripts, plans, goals and understanding*. Lawrence Erlbaum, 1977. 3.2.3

[Schank and Riesbeck, 1981] R. Schank and C. Riesbeck. *Inside Computer Understanding: Five Programs Plus Miniatures*. Erlbaum, Hillsdale, New Jersey, 1981. 3.2.3

[Schmidt *et al.*, 1978] C. F. Schmidt, N. S. Sridharan, and J. L Goodson. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11(1–2), 1978. 3.2.3

[Shklovsky, 1990] Victor Shklovsky. *Theory of Prose*. Dalkey Archive Press, Normal, Illinois, 1990. Trans. Benjamin Sher. Originally published 1929. 6.2.1, B.6.1

[Sindlar *et al.*, 2008] M. P. Sindlar, M. M. Dastani, F. Dignum, and J.-J.Ch. Meyer. Mental state abduction of bdi-based agents. In *Proceedings of Declarative Agent Languages and Technologies (DALT)*, Estoril, Portugal, 2008. 6.2.2

[Slade, 2011] Benjamin Slade. Beowulf on steorarume (beowulf in cyberspace). http://www.heorot.dk, 2011. Accessed 16 August 2011. 2

[Smith and Hancox, 2001] Elliot Smith and Peter Hancox. Representation, coherence and inference. *Artificial Intelligence Review*, 15:295–323, 2001. 3.1

[Smith, 1978] Carlota S. Smith. The syntax and interpretation of temporal expressions in english. *Linguistics and Philosophy*, 2(1):43–99, 1978. 4.4.2

[Song and Cohen, 1988] Fei Song and Robin Cohen. The interpretation of temporal relations in narrative. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, St. Paul, Minnesota, 1988. 4.4.2

[Steedman, 1995] Mark Steedman. Dynamic semantics for tense and aspect. In *The 1995 International Joint Conference on AI (IJCAI-95)*, Montreal, Quebec, Canada, 1995. 4.4.2

[Stein and Albro, 1996] Nancy L. Stein and Elizabeth R. Albro. Building complexity and coherence: Children's use of goal-structured knowledge in telling stories. In Michael G. W. Bamberg, editor, *Narrative Development: Six Approaches*. Erlbaum, Mahwah, New Jersey, 1996. 3.2.1

[Stein *et al.*, 2000] Nancy L. Stein, Tom Trabasso, and Maria D. Liwag. A goal appraisal theory of emotional understanding: Implications for development and learning. In Michael Lewis and Jeannette M. Haviland-Jones, editors, *Handbook of emotions (2nd ed.)*, pages 436–457. Guilford Press, New York, 2000. 3.2.1, B.1

[Steyvers and Griffiths, 2007] Mark Steyvers and Tom Griffiths. Probabilistic topic models. In T. Landauer, D McNamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning.* Laurence Erlbaum, 2007. 6

[Storey *et al.*, 2001] Margaret-Anne Storey, Mark Musen, John Silva, Casey Best, Neil Ernst, Ray Fergerson, and Natasha Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in protégé. In *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001)*, Victoria, British Columbia, Canada, 2001. 4.3.1

[Stoyanov *et al.*, 2010] Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. Coreference resolution with reconcile. In *Proceedings of the Conference of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, Uppsala, Sweden, 2010. 2.4

[Suh and Trabasso, 1993] Soyoung Suh and Tom Trabasso. Inferences during reading: Converging evidence from discourse analysis, talk-aloud protocols, and recognition priming. *Journal of Memory and Language*, 32:279–300, 1993. 3.2.1

[Swanson and Gordon, 2008] Reid Swanson and Andrew S. Gordon. Say anything: A massively collaborative open domain story writing companion. In *Proceedings of the First International Conference on Interactive Digital Storytelling*, Erfurt, Germany, 2008. 3.1

[Tackstrom and McDonald, 2011] Oscar Tackstrom and Ryan McDonald. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of the 33rd European Conference on Information Retrieval (ECIR 2011)*, Dublin, Ireland, 2011. 5.2.3

[Tanev, 2007] Hristo Tanev. Unsupervised learning of social networks from a multiple-source news corpus. In *Proceedings of the Workshop Multi-source Multilingual Information Extraction and Summarization (MMIES 2007) held at RANLP 2007*, Borovets, Bulgaria, 2007. 2.1

[Terenziani and Snodgrass, 2004] Paolo Terenziani and Richard T. Snodgrass. Reconciling point-based and interval-based semantics in temporal relational databases: A treatment of the telic/atelic distinction. *IEEE Transactions on Knowledge and Data Engineering*, 16(5):540–551, 2004. 3.3.1

[Todorov, 1966] Tzvetan Todorov. Les catégories du récit littéraire. *Communications*, 8:125–151, 1966. 3.2.1

[Todorov, 1969] Tzvetan Todorov. Structural analysis of narrative. *NOVEL: A Forum on Fiction*, 3(1):70–76, 1969. trans. Arnold Weinstein. 3.2.2

[Trabasso and Sperry, 1985] Tom Trabasso and Linda L. Sperry. Causal relatedness and importance of story events. *Journal of Memory and Language*, 24(5):595–611, 1985. 3.2.1

[Trabasso and Stein, 1997] Tom Trabasso and Nancy L. Stein. Narrating, representing, and remembering event sequences. In Tammy Bourg Paul van den Broek, Patricia J. Bauer, editor, *Developmental Spans in Event Comprehension and Representation*, pages 237–269. Erlbaum, Mahwah, New Jersey, 1997. 3.2.1

[Trabasso and van den Broek, 1985] Tom Trabasso and Paul van den Broek. Causal thinking and the representation of narrative events. *Journal of Memory and Language*, 24:612–630, 1985. 3.2.1, 3.2.1

[Trabasso and Wiley, 2005] Tom Trabasso and Jennifer Wiley. Goal plans of action and inferences during comprehension of narratives. *Discourse Processes*, 39(2&3):129–164, 2005. 3.2.1, 3.2.1

[Tversky, 1977] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. 5.2.1

[van de Bunt *et al.*, 1999] Gerhard G. van de Bunt, Marijtje A. J. Van Duijn, and Tom A. B. Snijders. Friendship networks through time: An actor-oriented dynamic statistical network model. *Computational & Mathematical Organization Theory*, 5(2):167–192, 1999. 6.2.1

[van den Broek, 1988] Paul van den Broek. The effects of causal relations and hierarchical position on the importance of story statements. *Journal of Memory and Language*, 27(1), 1988. (document), 3.1, 3.2.1

[van Dijk and Kintsch, 1983] Teun A. van Dijk and Walter Kintsch. *Strategies of Discourse Comprehension*. Academic, New York, 1983. 3.2.1, 3.3

[van Dijk, 1976] Teun A. van Dijk. Philosophy of action and theory of narrative. *Poetics*, 5:287–338, 1976. 3

[Vlach, 1993] Frank Vlach. Temporal adverbials, tenses and the perfect. *Linguistics and Philosophy*, 16(3):231–283, 1993. 3.3.1, 4.4.2

[Voloshinov, 1971] V. N. Voloshinov. Reported speech. In L. Matejka and K. Pomorska, editors, *Readings in Russian poetics: Formalist and structuralist views*, pages 149–175. MIT Press, Cambridge, 1971. 2.5.1

[Webber, 1987] Bonnie Lynn Webber. The interpretation of tense in discourse. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-87)*, pages 147–154, Stanford, CA, 1987. 3.3.1, 4.4.2

[Wiebe *et al.*, 2005] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evalution (formerly Computers and the Humanities)*, 39(2-3):165–210, 2005. 3.2.2, 5.2.3, 6.2.2

[Wiebe, 1990] Janyce M. Wiebe. Identifying subjective characters in narrative. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, pages 401–408, Helsinki, Finland, 1990. 2.5.1

[Wiebe, 1994] Janyce M. Wiebe. Tracking point of view in narrative. *Computational Linguistics*, 20:233–287, 1994. 6.2.2

[Wilensky, 1978a] Robert Wilensky. Pam. In R. Schank and C. Riesbeck, editors, *Inside Computer Understanding: Five Programs Plus Miniatures*. Erlbaum, Hillsdale, NJ, 1978. 3.2.3

[Wilensky, 1978b] Robert Wilensky. Why john married mary: Understanding stories involving recurring goals. *Cognitive Science*, 2:235–266, 1978. 3.2.3, B.2

[Wilensky, 1982] Robert Wilensky. Points: A theory of the structure of stories in memory. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*. Erlbaum, Hillsdale, New Jersey, 1982. 3.1

[Wilensky, 1983] Robert Wilensky. Story grammars versus story points. *Behavioral and Brain Sciences*, 6:529–623, 1983. 3.2.2

[Williams, 1975] Raymond Williams. *The Country and The City*. Oxford University Press, Oxford, 1975. 2.2

[Williams, 2002] Bernard Williams. *Truth and Truthfulness: An Essay in Genealogy*. Princeton University Press, Princeton, NJ, 2002. B.5.1

[Winegarden and Young, 2006] Joe Winegarden and R. Michael Young. Distributed interactive narrative planning system. In *Proceedings of the 2006 AAAI Spring Symposium on Distributed Plan and Schedule Management*, Stanford, California, 2006. 3.2.3

[Wolf and Gibson, 2005] Florian Wolf and Edward Gibson. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287, 2005. 3.2.2

[Yang and Bateman, 2009] Guowen Yang and John Bateman. The chinese aspect generation based on aspect selection functions. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP (ACL-IJCNLP 2009)*, Singapore, 2009. 4.4.2

[Zafiropoulos, 2001] Christos A. Zafiropoulos. *Ethics in Aesop's Fables: The Augustana Collection*. Brill Academic Publishers, 2001. 5.1, 5.2.3, 5.3.1

[Zarri, 1997] Gian Piero Zarri. Nkrl, a knowledge representation tool for encoding the 'meaning' of complex narrative texts. *Natural Language Engineering*, 3(2):231–253, 1997. 3.1

[Zarri, 2010] Gian Piero Zarri. Representing and managing narratives in a computer-suitable form. In *Representing and Managing Narratives in a Computer-Suitable Form*, Arlington, Virginia, 2010. 3.1, 3.2.3

[Zhang *et al.*, 2003] Jason Zhang, Alan Black, and Richard Sproat. Identifying speakers in children's stories for speech synthesis. In *Proceedings of EUROSPEECH 2003*, Geneva, 2003. 2.5.1

[Zukerman *et al.*, 2003] Ingrid Zukerman, Sarah George, and Yingying Wen. Lexical paraphrasing for document retrieval and node identification. In K. Inui and U. Hemjakob, editors, *Second International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP 2003)*, pages 94–101. Association for Computational Linguistics (ACL), 2003. 5.2.1

[Zunshine, 2006] Lisa Zunshine. *Why We Read Fiction: Theory of Mind and the Novel*. Ohio State University Press, Columbus, Ohio, 2006. 3.2.2, 6.1

[Zwaan and Radvansky, 1998] Rolf A. Zwaan and Gabriel A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185, 1998. 3.2.1

[Zwaan *et al.*, 1995] Rolf A. Zwaan, Mark C. Langston, and Arthur C. Graesser. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological Science*, 6(5):292–297, 1995. 3.2.1