Long-Answer Question Answering and Rhetorical-Semantic Relations

Sasha J. Blair-Goldensohn

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

COLUMBIA UNIVERSITY

2007

ABSTRACT

Long-Answer Question Answering and Rhetorical-Semantic Relations

Sasha J. Blair-Goldensohn

Over the past decade, Question Answering (QA) has generated considerable interest and participation in the fields of Natural Language Processing and Information Retrieval. Conferences such as TREC, CLEF and DUC have examined various aspects of the QA task in the academic community. In the commercial world, major search engines from Google, Microsoft and Yahoo have integrated basic QA capabilities into their core web search.

These efforts have focused largely on so-called "factoid" questions seeking a single fact, such as the birthdate of an individual or the capital city of a country. Yet in the past few years, there has been growing recognition of a broad class of "long-answer" questions which cannot be satisfactorily answered in this framework, such as those seeking a definition, explanation, or other descriptive information in response. In this thesis, we consider the problem of answering such questions, with particular focus on the contribution to be made by integrating rhetorical and semantic models.

We present DefScriber, a system for answering definitional ("What is X?"), biographical ("Who is X?") and other long-answer questions using a hybrid of goal- and data-driven methods. Our goal-driven, or top-down, approach is motivated by a set of *definitional predicates* which capture information types commonly useful in definitions; our data-driven, or bottom-up, approach uses dynamic analysis of input data to guide answer content. In several evaluations, we demonstrate that DefScriber outperforms competitive summarization techniques, and ranks among the top long-answer QA systems being developed by others.

Motivated by our experience with definitional predicates in DefScriber, we pursue a set of experiments which automatically acquire broad-coverage lexical models of *rhetorical-*

semantic relations (RSRs) such as Cause and Contrast. Building on the framework of Marcu and Echihabi (Marcu and Echihabi, 2002), we implement techniques to improve the quality of these models using syntactic filtering and topic segmentation, and present evaluation results showing that these methods can improve the accuracy of relation classification.

Lastly, we implement two approaches for applying the knowledge in our RSR models to enhance the performance and scope of DefScriber. First, we integrate RSR models into the answer-building process in DefScriber, finding incremental improvements with respect to the content and ordering of responses. Second, we use our RSR models to help identify relevant answer material for an exploratory class of "relation-focused" questions which seek explanatory or comparative responses. We demonstrate that in the case of explanation questions, using RSRs can lead to significantly more relevant responses.

Contents

List of Figures

List of Tables

To my wonderful girls, down the generations –
Betty Goldensohn, Gwenda Blair,
Rebecca Min and Sophie Min Goldensohn

Chapter 1

Introduction

There is a game my father used to play with my uncle. My father would open up the almanac, turn to a random page, and ask his little brother an obscure question, something like: "What is the highest point in Rhode Island?" Of course, he would act as if the answer (Jerimoth Hill, at 812 feet) were utterly obvious.

Apart from the inherent fun of tormenting one's siblings, the interest in this game can be thought of in terms of the difference between a *description* and a *list of facts*. There is a potentially endless list of facts about, say, Rhode Island. It has not only a highest elevation, but a state bird, first governor, and a shoreline whose length can be measured to the nearest foot. But an exhaustive list of all conceivable facts does not constitute a satisfying description.

For this reason, an encyclopedia is usually better than an almanac[1] if you actually want to learn what something is like, and not just quiz your brother. In an encyclopedia article, it is the job of the author to select and synthesize facts to create a description which tells the reader what a state, or any other topic, is "about." We expect this kind of descriptive

[1] For purposes of this motivational discussion, we consider an idealized "almanac" containing lengthy tables of extremely detailed facts, and an idealized "encyclopedia" containing concise textual descriptions which cover a much smaller number of selected, salient facts. Of course, many real almanacs and encyclopedias blur this distinction.

definition to concisely summarize the salient, *central* facts that make something special or significant, and to leave out more obscure information.

For instance, in the case of Rhode Island, the state's small size, not its highest point, is probably its best-known property. This central fact makes a good beginning to an encyclopedia-type definition:

> Rhode Island is a state in the New England region and is the smallest United States state by area.

Whereas in the case of Alaska, which is known for its geography, a description which begins by mentioning its highest point is entirely appropriate:

> Alaska, a state in the extreme northwest of North America, is home to the tallest mountain in the United States, Mount McKinley.

While the individual facts that are salient for a given entity differ, there are some *kinds* of facts which are relevant for descriptions in general. For instance, we often begin by placing something in a larger category, and differentiating it from other category members. The two examples above each show plausible beginnings of definitional descriptions which do just that. They describe their subjects, Rhode Island and Alaska, as states, and differentiate them from other states by their location and other well-known features.

A description must be more than a collection of salient facts of certain kinds, however. We implicitly expect a well-written encyclopedia article to proceed in a coherent manner, not only beginning with key facts of interest, but also avoiding redundancy and ordering content cohesively, perhaps through the use of rhetorical and semantic strategies.

Human authors, of course, are generally aware of these expectations. However, even the most complete human-written encyclopedia does not provide an article on every topic. And even when an article exists, it may be outdated, too short (or long), or simply focus on something other than what we wish to know; it may focus on Rhode Island's history rather than its educational institutions; it may mention that the mayor of Providence served time in prison, without explaining why.

In this thesis, we explore the problem of providing descriptive information automatically. We implement and evaluate techniques for creating descriptive answers to various kinds of *long-answer questions*, including those which seek definitions, biographies or explanations in response.

But how can we go from the "almanac" answer to the "encyclopedia" one? We must not only identify important information in terms of its type and salience, but also present it in a coherent way. In our initial work in DefScriber, we use a combination of "top-down" approaches that identify key information types using strongly structured rules, and "bottom-up" approaches that rely on statistical analysis of input data to identify which facts are most salient.

Despite successful evaluations of this hybrid technique, we find that there is a gap between what our top-down and bottom-up approaches can accomplish in terms of identifying certain kinds of interesting links in the content of our answers. To address this problem, we consider how an understanding of rhetorical and semantic relationships can enhance the fluency and content of our answers, while also allowing us to tackle new types of long-answer questions.

The relationships we study cover aspects of what have been termed "semantic" and "rhetorical" in other research, and thus we refer to them as rhetorical-semantic relations (RSRs). While we define RSRs formally in Chapter 3, our basic idea is to build models which capture lexical evidence of the kinds of information which human authors link by using causal and contrastive "cue phrases." By capturing this knowledge at the word level, we create a system-accessible resource which can tell us that the words *drought* and *hunger* are causally linked, and that *great* and *terrible* stand in contrast.

Having a model of such knowledge can help us enhance and expand our question answering capabilities. We experiment with applications which use RSR models to make answers more coherent by following semantic and rhetorical expectations, e.g. by using a model of contrast to juxtapose complementary information. Moreover, we expand our capabilities to implement QA techniques for questions which have a causal or comparative focus, rather than a more general descriptive purpose.

1.1 Research Questions

In summary, the issues we explore in this thesis can be described by three overarching research questions:

- How do we create well-structured, relevant answers to definitional, biographical and topic-focused questions? (Chapter 2)
- How do we learn and refine a model of rhetorical and semantic concepts for use as a resource in answering these questions? (Chapter 3)
- How can we apply this rhetorical-semantic model to enhance the performance (Chapter 4) and scope (Chapter 5) of our question answering techniques?

In the next section we outline our work in addressing each question in the indicated chapter(s).

1.2 Chapter Outline

1.2.1 DefScriber

In Chapter 2, we examine the problem of creating dynamic, multi-sentence answers to definitional, biographical and topic-focused questions, and present DefScriber, a system for answering these questions.

We begin by describing the hybrid concept of DefScriber, which combines goal- and data-driven approaches. We present a set of *definitional predicates* that we theorize based on a corpus study of definitions. These predicates form the basis of our goal-driven, or top-down, approach. We further discuss how this corpus study leads us to design a complementary data-driven, or bottom-up, approach which adapts and synthesizes various information retrieval and summarization techniques. We explain how both approaches are combined in the implemented system architecture.

We next present results from various evaluations of DefScriber. Our first evaluation is user-based, and collects judgments about DefScriber's performance on definitional ("What is X?") questions, finding improvement over competitive summarization baselines in several areas. The second evaluation is from the DUC 2004 conference, in which we adapt DefScriber to answer biographical ("Who is X?") questions. In this evaluation, we find that DefScriber performs at or above the level of the top peer systems according to an automated metric. The final evaluation discussed in this chapter is from the DUC 2005 conference, where we further adapt the system to answer topic-focused queries ("Describe Topic X, focusing on issues Y and Z."), and find that our performance is among the top peer systems, according to a measure which combines automated and human-judged metrics.

Lastly, we present our research in extending DefScriber's clustering component to improve results when working with input documents from speech sources, such as broadcast news, whose text is produced via automatic speech recognition (ASR). In this work, we attempt to improve the module-level performance of DefScriber's clustering component by analyzing ASR confidence measures. We find that using this additional speech-specific information in the clustering process indeed improves performance.

Key contributions in this chapter are: (1) An implemented system, DefScriber, which takes a novel "hybrid" approach combining top-down and bottom-up techniques for answering various long-answer questions; (2) An extensive set of evaluations analyzing the end-to-end performance of this approach using multiple metrics in various settings. The evaluations demonstrate significant improvements over baseline techniques and highly competitive performance with respect to peer systems developed by other researchers; (3) An extension to DefScriber's clustering module to handle mixed speech-text input, which results in improved performance at the module level.

1.2.2 RSR Modeling and Classification

In Chapter 3, we describe our work in modeling and classification of rhetorical-semantic relations (RSRs). We build on work by Marcu and Echihabi (Marcu and Echihabi, 2002), using a small set of manually identified cue phrases like *because* and *however* to mine examples of text span pairs across which we can infer RSRs such as Cause and Contrast. We implement this approach in TextRels, a system which gathers relation instances from unannotated corpora and uses the resulting data to build models of these relations. These models can then be used to classify the degree to which the relationship between a given pair of text spans resembles one of our RSRs. While the basic method we use to create these models is based on Marcu and Echihabi's work, we add a number of extensions in terms of both the modeling and evaluation process.

First, we analyze the set of basic parameters to the model, such as smoothing weights, vocabulary size and stoplisting. We achieve significant improvements in relation classification performance by optimizing these parameters.

Next, we work on filtering noise from our training data with two complementary techniques. We use automated topic segmentation to improve our extraction of inter-sentence relations by adding segment-based heuristics which specify where a topic segment boundary may occur with respect to a relation. We find that adding these heuristics improves our classification performance in several cases.

We also examine the impact of syntax-based filtering to remove extraneous portions of mined relation instances in our data. Based on a manual analysis of common errors in extraction, we implement several syntax-based heuristics for excising irrelevant material from the training data. Again, we find classification improvement in several cases.

Key contributions in this chapter are: (1) TextRels, a system which implements RSR models and forms the basis of various experiments with RSRs; (2) Several novel approaches for enhancing the quality of automatically extracted models, using topic segmentation and syntactic analysis; (3) An extensive set of evaluations which provide insight into the effect of various parameters on classification performance of our RSR models, using both automatically- and manually-created test data, including a detailed analysis with respect to our syntactic approaches, using the recently released Penn Discourse TreeBank (Prasad et al., 2006). These evaluations demonstrate that our approaches for improving model quality are successful in several cases.

1.2.3 Applying RSRs in DefScriber

In Chapter 4, we experiment with applying the RSR models and classifiers we create in Chapter 3 within DefScriber via an RSR-driven feature for assessing inter-sentence cohesion. As part of the process of introducing this new feature, we make updates to DefScriber's framework for content selection and ordering.

Using this updated framework, we can use supervised learning to set the parameters which DefScriber uses to combine this new RSR-driven feature with other features when building responses. We take advantage of available data from the DUC 2004 and 2005 conferences, from which we learn parameter settings for biographical and topic-focused questions, respectively.

We use a held-out test set of DUC topics and their associated human-created answers to measure our performance, separately evaluating content selection and content ordering tasks. We find that our overall performance improves significantly for the content selection task, but that the specific contribution from the RSR-derived cohesion feature is only incremental. In the ordering task, our performance improves as well, but does not reach statistical significance even with the addition of the RSR feature.

Key contributions in this chapter are: (1) Integration of a new RSR-based feature for measuring inter-sentence cohesion, for which we implement a supervised training framework and other new extensions to DefScriber; (2) An evaluation which separately considers aspects of content selection versus content ordering, and demonstrates that addition of the RSR-based feature achieves incremental improvements in both areas.

1.2.4 Applying RSRs For Causal and Comparison Questions

In Chapter 5, we explore the task of answering challenging question types which call for explanations or comparisons. We propose a “relation-focused” framework for considering these questions in terms of the RSR-type relations which they invoke, and focus specifically on how our Cause and Contrast relation models can help identify relevant information for explanation and comparison questions, respectively.

We implement CCQ, a system which uses a lexical expansion approach built over our RSR models for identifying question-relevant information for these new question types. The lexical expansion method which we implement uses our TextRels RSR models to generate relevant terms and assess intrinsic question relevance with respect to a relation-focused question (as opposed to the classification-based approach in Chapter 4, which assesses inter-sentence cohesion only). In addition to using our TextRels model to generate question-relevant terms, we consider complementary methods for term suggestion via WordNet and

Latent Semantic Analysis. Once relevant answer material is identified through these methods, we repurpose several of DefScriber's modules to organize and present answer information in CCQ.

We conduct a user-based evaluation to analyze the performance of CCQ on a test set of explanation and comparison questions. We find that using RSR-based lexical expansion increases the relevance of our explanation answers, and that the contribution is statistically significant under some conditions. For comparison questions, RSR-based methods only meet the performance achieved with other methods.

Key contributions of this chapter are: (1) Exploration of a sparsely researched area of QA, with a focus on two specific question types, explanations and comparisons; (2) Implementation of CCQ, a system for creating long-answer responses for these questions which applies lexical expansion over RSR models; (3) Evaluation of CCQ system performance using user judgments of answer quality. The evaluation demonstrates that for explanation questions, using RSRs can lead to significantly more relevant responses.

Chapter 2

DefScriber

2.1 Introduction

DefScriber is a fully implemented system for answering the kinds of *long-answer* questions discussed in Chapter 1, including those which seek definitions or biographies in response.

We pursue long-answer question answering to address information needs for which even the most advanced short-answer, or *factoid* systems may be of limited use. Consider, for instance, a student asked to prepare a report on the Hajj, an Islamic religious duty. In the context of short-answer QA, both patience and prescience will be required to elicit the core facts. First, a relatively long list of questions would be required (e.g., “Where is the Hajj carried out?”, “How long does it last?”, “Who undertakes a Hajj?” etc.). Second, knowing which questions to ask requires knowledge that the questioner likely does not have. That is, the questions that best elicit a description of one thing (e.g., the Hajj) can be quite different than those best suited for finding out about something else (e.g., Rhode Island).

Based on this observation, the core concept around which we create DefScriber is that of a *hybrid* system. The goal for this paradigm is to reflect an important aspect of answering these kinds of descriptive questions, which we describe intuitively above and examine empirically in a corpus study discussed later in this chapter: namely, that some kinds of information are frequently useful in defining or describing a wide range of subjects, while other kinds of information may be rarely used in general, but critical within the contexts of some answers. The frequently useful information types include things like category (“is-a") information, which is germane in nearly any definition (e.g., Rhode Island *is a* state; the Hajj *is a* pilgrimage; a Ford Taurus *is a* car). Other kinds of data may be critical in some cases but irrelevant or even nonsensical in others, e.g. the *number of doors* on a Ford Taurus.

For this reason, we follow a paradigm which combines top-down and bottom-up strategies. The top-down, or knowledge-driven, component guides answer content based on information types which are commonly useful. The bottom-up, or data-driven, component performs dynamic analysis of input data to identify themes which are important to the particular question which we are answering. In addition, both kinds of strategies can be used to help us in ordering information which we select in an answer. For instance, we can design top-down rules which attempt to put "is-a" information early in an answer, while using bottom-up analysis to avoid redundancy via subtopic clustering.

In this chapter, we focus on answering the first research question raised in Chapter 1, namely: How do we create well-structured, relevant answers to definitional, biographical and topic-focused questions? We begin with an overview of related work (Section 2.2). We next discuss the design of DefScriber's hybrid architecture for answering definitional ("What is X?") questions, drawing on both previous work and our own corpus study of definitional material (Section 2.3), and including extensions for answering biographical ("Who is X?") and topic-focused ("Describe Topic X, focusing on issues Y and Z.") questions. We analyze results of various evaluations of DefScriber over several versions of the system in Section 2.4. These results highlight our system's strong performance over several different kinds of tasks and evaluations. These include a survey-based evaluation in which users rate several aspects of answers for definitional questions (Section 2.4.1); an evaluation over biographical answers in which we examine both human-judged and automatic metrics for evaluating answers to biographical questions (Section 2.4.2); and an evaluation over topic-focused answers for which we present a combined overall measure which takes into account both human-judged and automatic metrics (Section 2.4.3). In addition to these end-to-end evaluations, we also investigate a specific sub-area of DefScriber's architecture with a module-level analysis of clustering performance in Section 2.5. In this work, we focus on the challenge of successful subtopic clustering when input documents are generated by automatic speech recognition

(ASR) from broadcast sources.

2.2 Related Work

Our hybrid approach to long-answer QA builds on research in both summarization and generation. Previous work in multi-document summarization has developed solutions that identify similarities across documents as a primary basis for summary content (Mani and Bloedorn, 1997; Carbonell and Goldstein, 1998; Hatzivassiloglou et al., 1999; Barzilay et al., 1999; Radev et al., 2000; Lin and Hovy, 2002; Daumé III and Marcu, 2005; Nenkova, 2006; Vanderwende et al., 2006). These similarities are typically included through sentence extraction, although some researchers have implemented systems which do reformulations as in Barzilay et al.'s information fusion (Barzilay et al., 1999) or sentence compression techniques (Siddharthan et al., 2004; Daumé III and Marcu, 2005). Yet even the approaches which are not purely extractive use data-driven strategies, in that themes and similarities which emerge from the data determine content.

Goal-driven, or top-down, approaches are more often found in generation. The concept of a script or plan (Schank and Abelson, 1977) is present in McKeown's schemas (McKeown, 1985), or Reiter and Dale's plan-based approaches (Reiter and Dale, 2000), and in other systems which use rhetorical-structure theory (RST) (Mann and Thompson, 1988) as a basis (Hovy, 1993; Moore and Paris, 1993; Marcu, 1997). These approaches are goal-driven because they use a schema, plan or other knowledge-based rules to specify the kind of information to include in a generated text. In early work, schemas were used to generate definitions (McKeown, 1985), but the information for the definitional text was found in a knowledge base. In more recent work, information extraction is used to create a top-down approach to summarization (Radev and McKeown, 1998) by searching for specific types of information which are of interest for a given topic and can be extracted from the input texts and added to a summary (e.g., perpetrator information for a summary on the topic of terrorism). In such cases, the summary briefs the user on domain-specific information which is assumed *a priori* to be of interest.

Research in question answering (QA) has traditionally focused on short-answer, or "factoid," questions, where the answer being sought is typically a short text snippet containing an "atomic" fact, such as a birthdate of an individual, capital city of a country or height of a mountain. However, the QA track at the annual TREC conference[1] began to include "definitional" questions in 2003 (Voorhees and Buckland, 2003). Beginning in 2004, the annual summarization-focused conference, DUC[2] began to include summarization tasks with a question-focused element, including the main tasks since 2005 (Dang, 2005). The coming together of these two branches of research have born out, to some extent, the vision statement articulated by Carbonell et al. (Carbonell et al., 2000) about the potential synergy in these two areas.

Yet the approaches of systems which participate in these evaluations still vary widely. Some systems have adapted short-answer QA techniques to the long-answer tasks, by decomposing a long-answer question into a series of factoid questions. This approach, coined as "QA by Dossier" by Prager et al. (Prager et al., 2004) seems especially well-suited to questions such as biographies where the decomposition is more reliable because information such as birthdate or family history is commonly of interest for any biography, and can even be learned for specific subcategories, e.g., by occupation (Filatova and Prager, 2005). Lacatusu et al. (Lacatusu et al., 2006) experiment with several techniques for both syntactic and semantic techniques for decomposition for DUC-style question topics, which can be several sentences long.

Hybrid strategies, which include both bottom-up and top-down components, have also been used for long-answer QA. These include work on definitional and biographical question-answering. Cui et al. (Cui et al., 2005) focus on learning text patterns without using semantic assumptions; Xu et al. (Xu et al., 2004) examine issues in porting a hybrid method from English to Chinese. Weischedel et al. (Weischedel et al., 2004) explore biographical questions using a combination of methods that are largely complementary to those used in DefScriber–namely, identification of key linguistic constructions and information extraction (IE) to identify specific types of semantic data.

These differences in approach may owe in part to the evaluation metrics which are used

[1] `http://trec.nist.gov`

[2] `http://duc.nist.gov`

in the different conferences; in the TREC QA track, the finding of specific "nuggets" of information is emphasized by measures like nugget-based recall and a distinction between critical and non-critical nuggets (Voorhees and Buckland, 2005), using nugget-based evaluation systems such as Pourpre (Lin and Demner-Fushman, 2005) and Nuggeteer (Marton and Radul, 2006). While these metrics have the advantage of specifying the task more concretely, they lack a strength of both the Pyramid (Nenkova and Passonneau, 2004) and ROUGE (Lin, 2004) metrics which are used in DUC; namely, both Pyramid and ROUGE are designed to account for natural variation that occurs in long answers, i.e., the intuition that when one is dealing with non-factoid questions and answers which can be several paragraphs in length, there may be multiple valid ways to answer the same question. To account for this, both metrics score system answers with respect to multiple human-created abstracts, which presumably represent some of the possible natural variation in equally valid answers. However, Pyramid and ROUGE take quite different approaches to the problem of lexical variation. While Pyramid uses human judgments to match "Summary Content Units" at a semantic level, ROUGE is limited to word overlap measures.

Another important question which is not directly addressed by these measures of summary *content* is how (or whether) we should judge a long answer as more than a list of unordered bits of information; DUC uses human judges to make ratings which evaluate aspects of response coherence, whereas TREC's nugget-based scoring does not reflect these issues.

Looking beyond text-only applications, there are several areas of research which share some of the properties of long-answer QA. Specialized applications provide dynamic descriptive information in domains like real estate search (Zhou and Aggarwal, 2004), using audio and video channels in an interactive environment, although these descriptions are largely template-based. The TRECVID track (Over et al., 2005) in the TREC conferences has explored content-based search over video; Katz et al. explore using a natural language interface for descriptive responses about moving objects in video footage (Katz et al., 2003).

QA systems which wish to use multi-modal input, such as speech and text, must be able to interpret the input correctly at a component level. In the case of long-answer QA, these can include clustering and ordering components, which in turn depend on similarity measures between candidate content units. A number of papers arising from the Topic Detection and Tracking (TDT) program (Wayne, 2000), focus on an integrated approach to such similarity computations. However, while ASR and text documents are in some cases evaluated separately as in Chen et al. (Chen et al., 2004) to allow for systems to preferentially select one or the other, the actual representation used for similarity measures over speech documents is generally the ASR transcript, with non-textual output of the ASR process, such as word recognition probability, being ignored. While this approach makes for easy integration, as components built for text can be used without any changes on the ASR transcript of a speech document, potentially valuable information in the speech signal may be lost. Allan et al. (Allan et al., 1999) discuss the significant degree to which "real" ASR degrades topic-tracking performance when compared to closed-caption transcripts but do not implement solutions; Maskey and Hirschberg (Maskey and Hirschberg, 2005) make a convincing case that there is indeed important information in a speech signal which is lost by simply using an ASR transcript representation of speech documents, showing that use of acoustic/prosodic features significantly improves a speech summarization task.

2.3 A Hybrid Design

Answering a "What is X?" definitional question based on a set of input documents and creating a general-purpose multi-document summary are clearly related problems. Yet, as readers, we have more specific expectations for a definition than for a general-use summary. In Section 2.3.1, we propose the idea of a set of *definitional predicates* to model some of the key information types which are often seen in definitions, and describe how we implement them for DefScriber's top-down, or goal-driven, strategy. At the same time, we recognize that however strong our goal-driven strategy, we need a system which can recognize and dynamically adapt to the different types of data which characterize any given subject. In Section 2.3.2, we describe our development of a bottom-up, or data-driven, module which adapts methods used in summarization and information retrieval to dynamically identify and organize themes which may emerge from the data.

Predicate	Description	Instance Example
Genus	Conveys a category or set to which the term conceptually belongs.	The Hajj is a type of ritual.
Species	Describes properties of the term other than (or in addition to) the category to which it belongs.	The annual hajj begins in the twelfth month of the Islamic year.
Contrast	Contrasts or compares the term with something else.	Unlike other pillars such as charity, which are habitual if not daily duties in a Muslim's life, the Hajj is generally a one-time effort.
Cause	Explicitly states that the term is involved in as a causal agent in some event or process.	The experience of the Hajj pilgrimage produced a marked shift in Malcolm X's views.
Target Partition	Divides the term into two or more categories, conceptual and/or physical.	Qiran, Tamattu', and Ifrad are three different types of Hajj.
Non-specific Definitional (NSD)	Text which contains information relevant to a broad background understanding of the term, but which may not fall into a specific predicate category mentioned above. (Note that any of the more specific predicates mentioned above is also an NSD instance.)	Costs: Pilgrims may pay substantial tariffs to the occupiers of Makkah and the rulers of the lands they pass through...

Table 2.1: Definitional Predicates: Descriptions and Example Instances for the term "Hajj"

2.3.1 Goal-Driven

Table 2.1 describes the set of *definitional predicates* which we theorize for our goal-driven methods. We choose this set of predicates based predominantly on two factors (1) prevalence in the literature and (2) a manual corpus analysis of definitional documents. For instance, Genus and Species predicates are the first two predicates we include because related work consistently identifies these two concepts as core for definitions. We build on such work as Klavans and Muresan (Klavans and Muresan, 2000), who acquire dictionary-type definitions, and Sarner and Carberry (Sarner and Carberry, 1988), who propose three "strategic predicates," including "identification" and "properties" predicates which are analogous to our Genus and Species, respectively. Research in terminological theory (Sager and L'Homme, 1994) and philosophy (Swartz, 1997) also theorizes that the type of information modeled by many of our predicates (including Genus, Species, Synonym and Target Partition) is crucial to descriptive-type definitions. Our definitional predicates differ from relations of the kind used in Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) in that our predicates are not meant to fit into an overall scheme for mapping document structure. As such, they are closer to RST's "subject matter," or informational-semantic relations. However, unlike RST relations, our predicates are explicitly meant to describe the content of a text span in terms of the information it provides with respect to the subject of a definition. RST's relations, by contrast, may hold between arbitrary clauses, sentences, or larger units of text[3].

In order to implement these predicates in our system, we perform a manual corpus analysis to help estimate the frequency of occurrence and variations in realization of these different types. We select 14 terms as potential definition targets, covering several diverse categories: geopolitical, science, health, and miscellaneous. For each term we collected approximately five documents containing generally descriptive information, in some cases in the form of an encyclopedia-type definition. Two individuals then divided the documents and marked each for occurrences of specific types of information. These types evolved during the coding process to become the set of definitional predicates designated in Table 2.1. However, as the markup was done collaboratively and with evolving standards, we essentially end up with a single set of annotations and do not attempt inter-rater agreement measures, even though some documents were annotated twice.

We explored two approaches for identifying predicates. The first uses machine learning to learn a feature-based classification that predicts when a predicate occurs. The second uses hand-crafted patterns created manually from the annotated data.

We use machine learning techniques to analyze the annotated documents and extract rules for predicate identification. These approaches allow us to efficiently discover relation-

[3] However, in Chapter 4, where we integrate a model of causality into DefScriber for the purpose of assessing inter-sentence cohesion, we do implement a concept closer to that of RST.

ships between text features and the presence of definitional predicate instances. The text unit we use here is the sentence; that is, we wish to discover rules that will take the features of a source sentence as input, and output those (if any) definitional predicates that are predicted to be in the sentence.

Feature selection was done in part using observations from the document markup exercise. For instance, we include several features measuring aspects of term "concentration," or density, i.e., the term's frequency within a sentence and/or nearby sentences, based on the observation that appearance of the term appears to be a robust predictor of definitional material.

We also include features for relative and absolute position of a sentence in a document, based on the observation that information tends to concentrate toward the top of documents. Other features, such as presence of punctuation, are added to detect full-sentence text (as opposed to headings or other fragments), since most predicates other than NSD seem to occur mainly in full sentences. Some "blind" features such as bag-of-words are also used. We applied two machine learning tools to the learning problem: the rule learning tool Ripper (Cohen, 1995) and the boosting-based categorization system BoosTexter (Schapire and Singer, 2000). Both algorithms performed similarly in terms of the accuracy of their predictions on test data; Ripper's rules are used in DefScriber since they are somewhat simpler to implement.

The nonspecific definitional (NSD) predicate, which indicates a sentence's relevance to any aspect of defining the term, fares well using rules that consider term concentration and position in document. Using cross-validation, an F-measure of 81 percent was obtained with Ripper (76 percent using BoosTexter). This is sufficient for DefScriber since this predicate is not used to place sentences directly into the definition, but rather to pare down noisy and voluminous input by pulling out sentences which merit further examination. However, neither tool derives high accuracy for the other predicates; we believe that this is due in part to the sparsity of examples for these predicates, as well as to the fact that the surface features we use may not provide sufficient information to the classifiers in these cases.

Therefore, we turn to a hand-crafted set of lexicosyntactic patterns to extract some of the specific (i.e., non-NSD) predicates. Using syntax trees from the predicate-annotated documents, we create a set of high-precision patterns for the two predicates most core to definitions: Genus and Species. We model sentences containing both predicates at once, as these "G-S" sentences provide a strong grounding context for understanding the term. G-S sentences situate a term in a higher level category (its "genus"), and the "species" information in such sentences tends to give key features that distinguish a term within that category. Rather than modeling the patterns at the word level, i.e., as flat templates with slots to fill, we model them as partially specified syntax trees (Figure 2.1).

One such pattern can match a large class of semantically similar sentences without having to model every type of possible lexical variation. This approach is similar to techniques used in information extraction (Grishman, 1997), where partial subtrees are used for matching domain-specific concepts and named entities because automatic derivation of full parse trees is not always reliable.

While the precision of our G-S patterns is generally high, the data-driven techniques described in the next section offer additional protection from false or extraneous matches by lowering the importance ranking of information not corroborated elsewhere in the data. For instance, in creating the definition for "Hajj" as shown in Figure 2.2, the system finds several G-S sentences, including: "The Hajj was Muhammad's compromise with Arabian Paganism." This sentence is in principle a correct match, but the Genus and Species given here are metaphorical and somewhat extraneous to defining the term. The fact that this information is less central to the definition is detected by our data-driven techniques as a low statistical "centrality" score, and it is thus not included in the output definition.

Figure 2.1 shows the transformation from example sentence to pattern, and then shows a matching sentence. Our patterns are flexible – note that the example and matched sentences have somewhat different trees. Another point of flexibility is the verbal phrase which is matched. The placeholder *FormativeVb* matches, at the lemma level, a manually constructed set of verbs expressive of "belonging" to a category (e.g., *be, constitute, exemplify, represent*, etc.). Using our predicate-annotated data set, we manually compile 23 distinct G-S patterns. In the evaluation described Section 2.4.1, at least one G-S sentence is identified in 16 of 19 questions, with a mean of 3.5 G-S sentences per term (culled from a mean

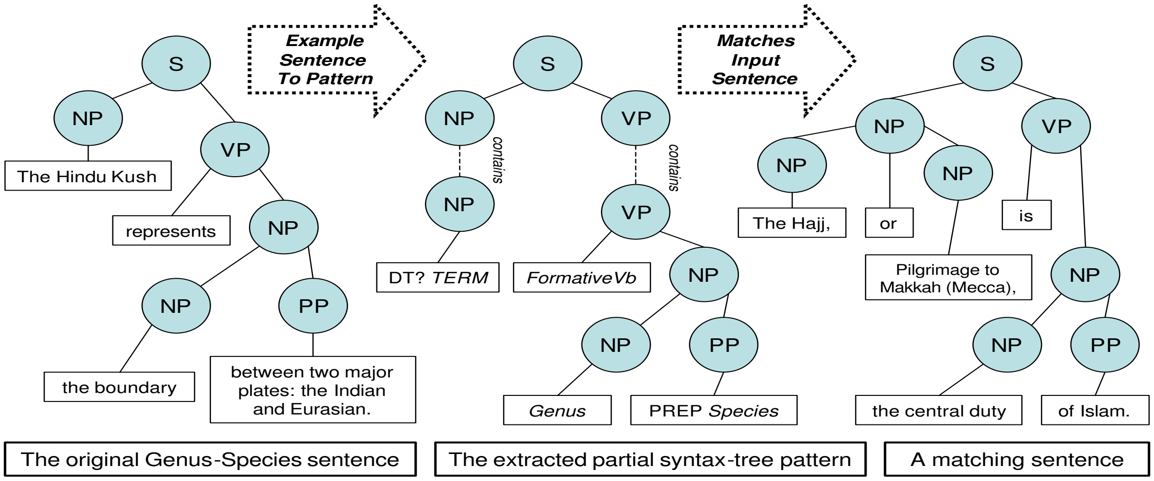

Example Sentence To Pattern
Matches Input Sentence
S
NP
VP
The Hindu Kush
represents
NP
NP
PP
the boundary
between two major plates: the Indian and Eurasian.
contains
DT? TERM
FormativeVb
Genus
PREP Species
The Hajj,
or
Pilgrimage to Makkah (Mecca),
is
the central duty
of Islam.
The original Genus-Species sentence
The extracted partial syntax-tree pattern
A matching sentence

Figure 2.1: Pattern extraction and matching for a Genus-Species sentence from an example sentence.

of 15 documents retrieved), with precision 96%[4].

2.3.2 Data-Driven

While our set of predicates, including Genus and Species, are domain-neutral, they are not meant to model all possible important information for a given term definition. Some information types may be hard to define *a priori*, or be difficult to identify automatically. Also, a given sentence may instantiate a definitional predicate but include only peripheral content. We address these issues in the data-driven stage of DefScriber's pipeline, applying statistical techniques adapted from multi-document summarization over the set of nonspecific definitional (NSD) sentences identified in the goal-driven stage.

First, a definition *centroid* is computed as a term-frequency vector over all NSD sentences. Then the individual sentences are sorted in order of decreasing "centrality," as approximated by IDF-weighted cosine distance from the definition centroid. Used alone, this method can create a definition of length N by taking the first N unique sentences out of this sorted order, and serves as the **TopN** baseline method in the evaluation. Note

[4] Recall is unknown with respect to the number of potential Genus-Species sentences in the corpus of available web documents. For our purpose in creating a single extractive definitional answer, recall is much less critical than precision, since we find that inclusion of a single G-S sentence is sufficient.

that this method approximates centroid-based summarization, a competitive summarization technique (Radev et al., 2000).

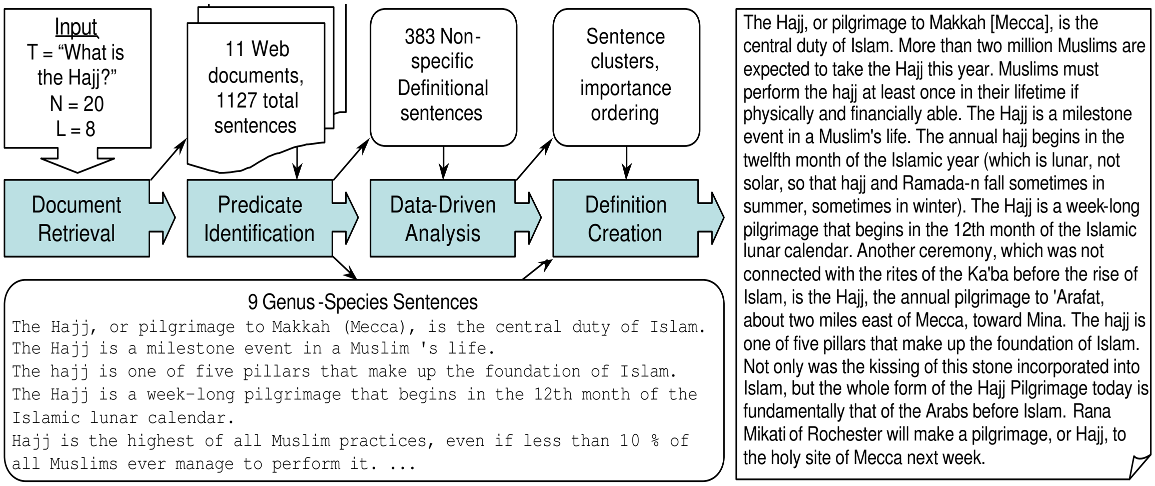
An important wrinkle in our TopN implementation concerns how we calculate the IDF statistic to determine term weighting. With a clustering similarity measure using "global" IDF computed over a large collection, we observe that at times, we can suffer from overweighting of terms which are rare in general, but common in the relevant documents for a given term. For example, even though "hajj" is a globally rare word, we do not want all the sentences containing this word to cluster together in our example of Figure 2.2, or else the clustering will not be able to separate out various sub-topics of the hajj. To account for this, we factor in "local" IDF values calculated dynamically from the pool of NSD sentences in combination with the "global" IDF value calculated statically, when weighting our cosine distance measures.

After using this improved IDF weighting in our TopN method to order the NSD sentences, we perform a Nearest Neighbor *clustering* (cf., (Manning and Schütze, 1999, p.604)), using IDF-weighted cosine distance as our similarity measure. When selecting sentences for the output summary, we attempt to avoid redundancy and maximize subtopic coverage by avoiding inclusion of multiple sentences from the same cluster.

Lastly, we use lexical similarity to assess *cohesion* when adding sentences to an in-progress answer. This method again uses IDF-weighted cosine distance to assess the cohesion "goodness" of choosing a given sentence to add to the summary. The **Data-Driven** method used in our evaluation chooses the first sentence for an answer as in TopN, and chooses the remaining sentences as the top-ranked (by centroid) sentence from the cluster that minimizes cosine distance from the definition centroid and avoids redundancy with respect to the clusters of previously chosen sentences.

2.3.3 Putting it Together

Using a combination of these Goal- and Data-Driven approaches, we implement DefScriber as an end-to-end system for creating definitional answers. A modular view of the system is illustrated in Figure 2.2, and consists of the following modules (the previous sections give additional details on the operation of each module):


Input
T = "What is the Hajj?"
N = 20
L = 8
11 Web documents, 1127 total sentences
383 Non-specific Definitional sentences
Sentence clusters, importance ordering
Document Retrieval
Predicate Identification
Data-Driven Analysis
Definition Creation
9 Genus -Species Sentences

```
The Hajj, or pilgrimage to Makkah (Mecca), is the central duty of Islam.
The Hajj is a milestone event in a Muslim 's life.
The hajj is one of five pillars that make up the foundation of Islam.
The Hajj is a week-long pilgrimage that begins in the 12th month of the
Islamic lunar calendar.
Hajj is the highest of all Muslim practices, even if less than 10 % of
all Muslims ever manage to perform it. ...
```

The Hajj, or pilgrimage to Makkah [Mecca], is the central duty of Islam. More than two million Muslims are expected to take the Hajj this year. Muslims must perform the hajj at least once in their lifetime if physically and financially able. The Hajj is a milestone event in a Muslim's life. The annual hajj begins in the twelfth month of the Islamic year (which is lunar, not solar, so that hajj and Ramada-n fall sometimes in summer, sometimes in winter). The Hajj is a week-long pilgrimage that begins in the 12th month of the Islamic lunar calendar. Another ceremony, which was not connected with the rites of the Ka'ba before the rise of Islam, is the Hajj, the annual pilgrimage to 'Arafat, about two miles east of Mecca, toward Mina. The hajj is one of five pillars that make up the foundation of Islam. Not only was the kissing of this stone incorporated into Islam, but the whole form of the Hajj Pilgrimage today is fundamentally that of the Arabs before Islam. Rana Mikati of Rochester will make a pilgrimage, or Hajj, to the holy site of Mecca next week.

Figure 2.2: DefScriber module-level processing of "What is the Hajj?" Individual modules are described in Section 2.3. (In the input, T, N and L represent query, maximum source documents, and desired answer length, respectively.)

1. **Document Retrieval** This module retrieves text documents up to a user-specified maximum number. We rely on the IR engine implementation to retrieve the most relevant documents for the target term (i.e., the "X" in a "What/Who is X?" question). Our current implementation can search local collections as indexed by Lucene[5], or the web as indexed by Google[6].

2. **Top-Down: Predicate Identification** This module identifies which sentences in the returned documents are instances of definitional predicates, and thus should be analyzed in constructing an answer. (See Table 2.1 for descriptions and examples of these predicates.)

   A first pass identifies Non-Specific Definitional (NSD) sentences, i.e. sentences which are at least broadly relevant for a definition, using a classifier trained over an annotated corpus of definitional material. This classifier takes into account occurrences of the target term ("X"), as well as several other features such as sentence position, length, use of pronouns, etc. Once the NSD sentences are identified, a set of hand-crafted

[5] http://lucene.apache.org

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

lexico-syntactic patterns is applied to find any sentences containing Genus and Species predicates, i.e. sentences containing both category and differentiator information.

3. **Bottom-Up: Data-Driven Analysis** This module analyzes the entire set of NSD sentences in order to determine from the data itself what the key themes of this topic are. First, individual sentences are sorted by vector distance from an NSD topic centroid. Next, we do a Nearest Neighbor clustering to separate out sub-topics. As mentioned earlier, we use a modified IDF weighting that considers dynamically calculated "local" IDF of our NSD sentence set to reduce the influence of globally rare but locally frequent words in clustering and other similarity-based decisions.

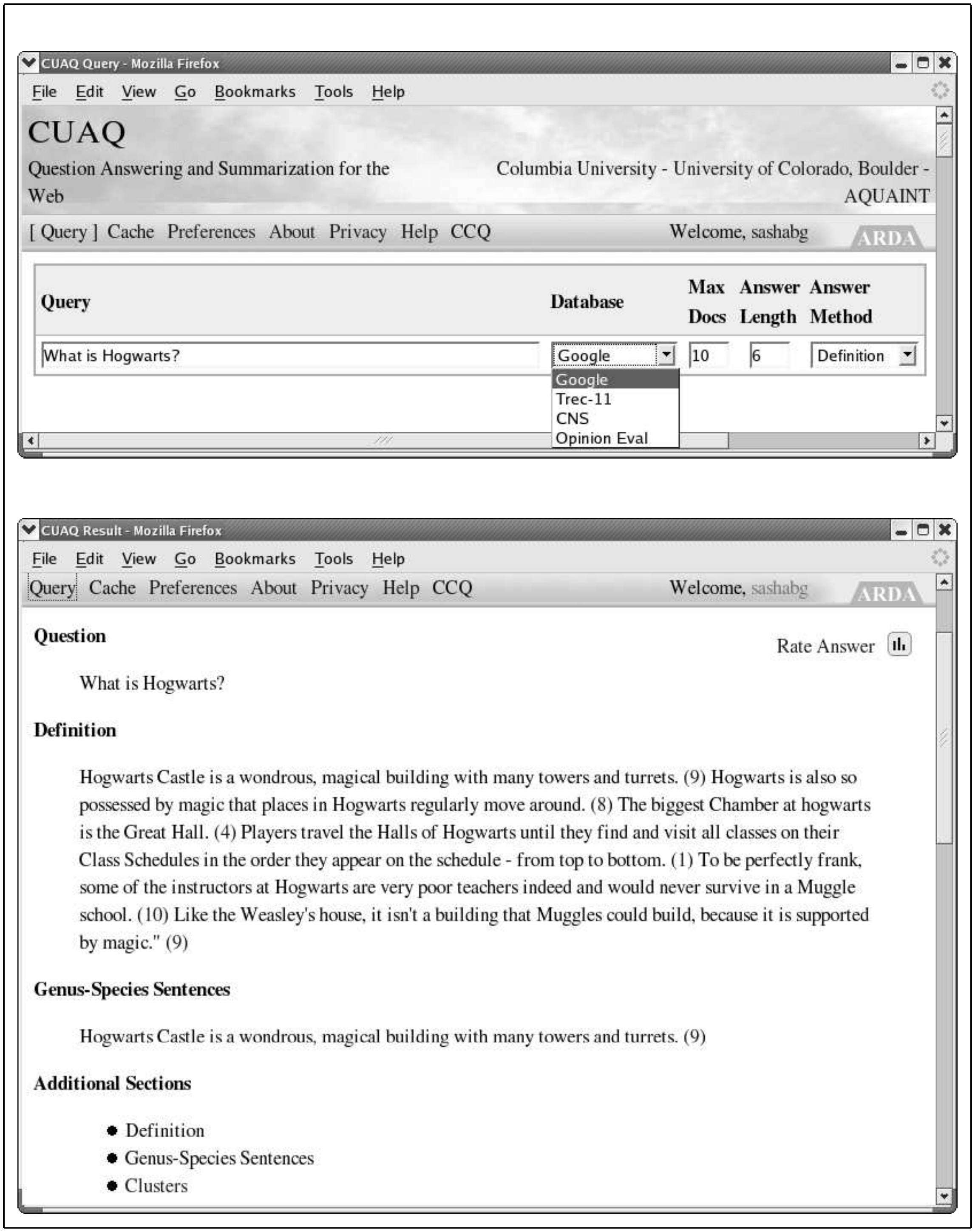
4. **Definition Creation** This module selects which sentences to extract into the output answer, and in what order. We always begin with the top-ranked Genus-Species sentence found (or if none is found, we use the top-ranked sentence by centrality with respect to the topic centroid). We then follow up with NSD sentences[7] , selecting each successive sentence using the data-driven analysis to maximize sentence centrality, cluster coverage, and coherence with the previous sentence (as measured by lexical overlap). Sentences are added until we reach the user-specified answer length.

2.3.4 Web Interface

We integrate DefScriber into a Web-based interface to the CUAQ system, which integrates several QA-related projects jointly developed at Columbia and the University of Colorado. Figure 2.3 shows example query input and output screens.

While the interface itself is not a primary contribution of this thesis, we mention it briefly here as it has been useful within the development process of DefScriber as a means of running ad-hoc experiments and demonstrations.

[7] The algorithm for choosing these "follow up" sentences is discussed in more detail in Chapter 4.


CUAQ Query - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
[Query] Cache Preferences About Privacy Help CCQ
ARDA
Google
Trec-11
CNS
Opinion Eval
CUAQ Result - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Query Cache Preferences About Privacy Help CCQ
ARDA
Rate Answer

CUAQ

Question Answering and Summarization for the Web

Columbia University - University of Colorado, Boulder - AQUAINT

Welcome, sashabg

Query	Database	Max Docs	Answer Length	Answer Method
What is Hogwarts?	Google	10	6	Definition

Welcome, sashabg

Question

What is Hogwarts?

Definition

Hogwarts Castle is a wondrous, magical building with many towers and turrets. (9) Hogwarts is also so possessed by magic that places in Hogwarts regularly move around. (8) The biggest Chamber at hogwarts is the Great Hall. (4) Players travel the Halls of Hogwarts until they find and visit all classes on their Class Schedules in the order they appear on the schedule - from top to bottom. (1) To be perfectly frank, some of the instructors at Hogwarts are very poor teachers indeed and would never survive in a Muggle school. (10) Like the Weasley's house, it isn't a building that Muggles could build, because it is supported by magic." (9)

Genus-Species Sentences

Hogwarts Castle is a wondrous, magical building with many towers and turrets. (9)

Additional Sections

- Definition
- Genus-Species Sentences
- Clusters

Figure 2.3: Input and output screens for the Web-based interface to DefScriber via the CUAQ system.

2.4 Versions and Evaluations

Over the past few years, the DefScriber system has proved flexible as we have successfully deployed it in several evaluations, adapting the system to new tasks at each point.

2.4.1 Definitions

In our original evaluation, we focus exclusively on the task of definitional, or "What is X?" type questions. In this evaluation we use human judgments to measure several dimensions of answer quality, which we assess via a set of five evaluation questions (listed in Table 2.2) to collect ratings for relevance, redundancy, structure, breadth of coverage, and term understanding[8]. We evaluate three configurations of the system, namely:

TopN baseline This baseline, described earlier in Section 2.3.2 approximates a centroid-based summarization (Radev et al., 2000) of the NSD sentences.

Data-Driven This method, also described in Section 2.3.2, adds the data-driven techniques for clustering and cohesion over the TopN baseline.

DefScriber This method is the default DefScriber configuration, which integrates all the above data-driven techniques with goal-driven Genus-Species predicate identification. We place the top-ranking (in terms of TopN) G-S sentence first in the definition, and use the cohesion-based ordering to add the remaining sentences. We call this integrated goal- and data-driven method DefScriber.

We choose a set of 24 terms (Table 2.2) for which to create answer definitions using the Internet as our knowledge source. We pick half of the terms ourselves, aiming for varied domain coverage; the other half were randomly chosen from among the definitional questions proposed as part of a pilot evaluation for the AQUAINT research program (Harman, 2002). For each of the test terms, we evaluate the three system configurations listed above; each configuration produces a 10-sentence definitional answer for each of the terms in Table 2.3 using 15 Internet documents.

[8] To understand why coverage is not simply the opposite of redundancy, imagine a definition of Hajj that is a completely non-redundant history of Mecca.

Category	Question
Structure	How would you rate the structure, or organization of the definition?
Relevance	Approximately how many sentences are relevant to describing or defining the term?
Coverage	How would you describe the breadth of coverage of the term by the information in the passage?
Redundancy	Approximately how many sentences are redundant with some other sentence(s) in the passage?
Term Understanding	How would you rate your overall understanding of the term after reading the passage?

Table 2.2: Questions in the user evaluation described in Section 2.4.1.

Source	Terms
Pilot Evaluation	asceticism, Aum Shinrikyo, battery, fibromyalgia, *gluons*, goth, Hajj, Mobilization for Global Justice, nanoparticles, religious right, *Shining Path*, Yahoo!
Hand-picked	autism, Booker Prize, Caspian Sea, East Timor, *hemophilia*, *MIRV*, orchid, pancreas, passive sonar, skin cancer, tachyons, *tsunami*

Table 2.3: Evaluation terms for user evaluation described in Section 2.4.1. Terms in *italics* were in the development set, while the rest were in the test set.

38 judges participated in the evaluation, and each was asked to rate definitions for six different test terms. This resulted in an average of four rated samples for each of the 57 answer definitions (19 test terms × three system configurations). Figure 2.4 shows the resulting mean feature scores for each system configuration. DefScriber achieves the best scores in structure, redundancy, term understanding, and relevance. We perform Friedman's test for correlated samples (used instead of ANOVA because the input data are ratings and not truly equal-interval measures) (Lowry, 2006) to examine whether there is a significant effect on score depending on the configuration used. We find that there is a significant effect in every category except coverage ($P < .05$); using Dunn's post test to compare individual pairs, we find that DefScriber's is significantly better than both other configurations in both structure and understanding; significantly better than data-driven in relevance; and

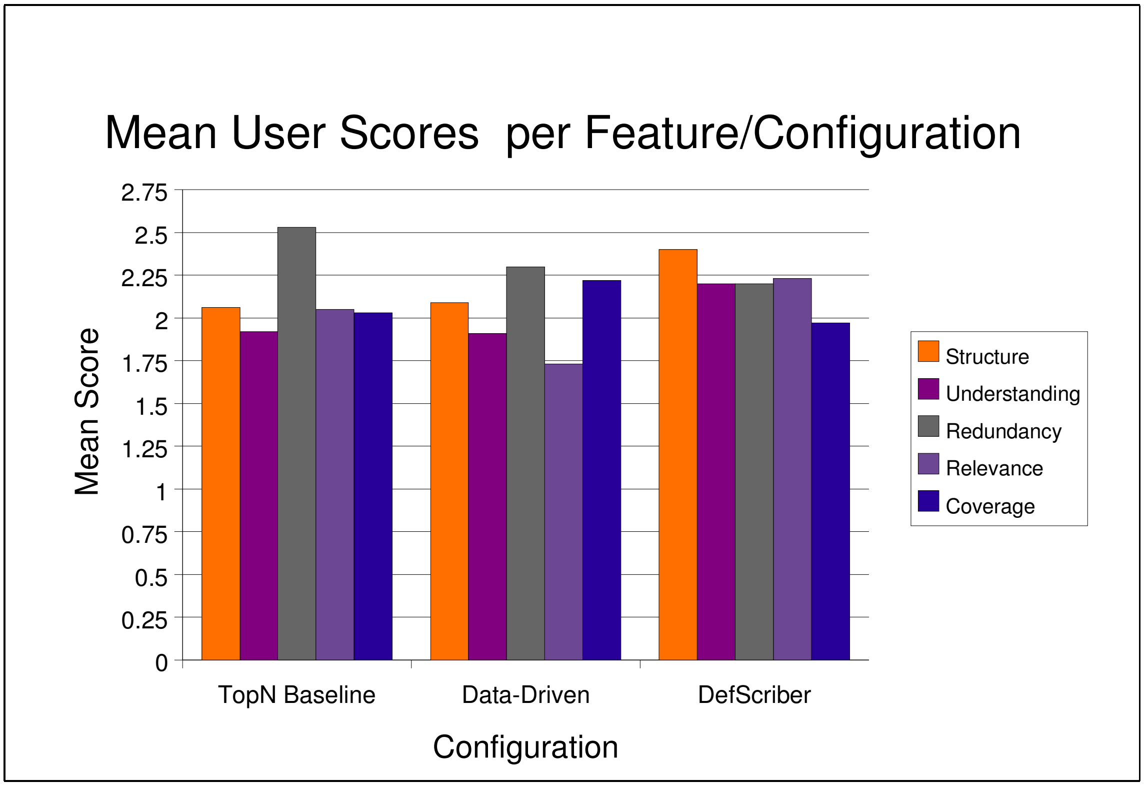


Figure 2.4: Mean user judgment scores for each configuration and feature type in Section 2.4.1. Higher is better for all features *except* Redundancy.

significantly better than baseline in redundancy. While DefScriber does worst in coverage, there are no statistically significant differences.

We observe that the relevance and coverage appear to be somewhat inverse in their relationship, indicating the tradeoff in these properties; also, the baseline system's high redundancy indicates that both other configurations, which use clustering, are more successful in grouping together related concepts. With the best performance in four of five categories, DefScriber is clearly the best configuration. In particular, we are encouraged that users appear to find having a Genus-Species sentence first (the primary difference between the data-driven and DefScriber configurations) to yield better answer structure and overall term understanding.

2.4.2 Biographies

There is clear similarity between definitional ("What is X?") and biographical ("Who is X?") questions. Having found DefScriber to achieve strong results for definitions, we experiment with adapting it for biographical questions as part of our participation in the DUC 2004 conference. We describe that work in this section; our original paper from that conference contains more detail in some areas (Blair-Goldensohn et al., 2004).

In the DUC 2004 conference (Over, 2004), one of the tasks was to create biographically-focused responses given the name of an individual and a set of twelve to eighteen documents which mention that individual (sometimes as a primary focus of the document, sometimes tangentially). There were 50 individuals/topics overall, ranging from John F. Kennedy, Jr. to Sonia Gandhi.

Before applying DefScriber for this task, we made certain targeted changes to improve the quality of summaries for questions relating to individuals and groups of people, as opposed to the more general class of terms which DefScriber is meant to handle.

In the initial step, identification of definitional material performs an information filtering function. Since we rely on string matching on the target of our question (i.e., the "X") to anchor the detection of definitional information, we needed to adapt the system to an X which was a personal name. In particular, we loosened the criteria for matching instances of the target term X, as it was important to allow for the fact that names are often shortened or abbreviated when referenced repeatedly in text. By relaxing sentence filtering to accept sentences containing partial matches of the target name, we observed that recall of relevant sentences in the training sets appeared much improved.

Another addition which we experimented with for this evaluation was a post-processing system for the rewriting of names and pronouns (Nenkova and McKeown, 2003) to make DefScriber's output more coherent. This experimental addition reflected our belief that reference resolution, always an issue with extracted text, can be particularly treacherous when the core concept being defined is a person who may be referred to in a number of ways. While the rewriting system we used has previously been deployed in the context of general news summarization (McKeown et al., 2003), this was our first effort at integrating it within a question answering architecture.

Since we evaluated only a single submission in this task, it is difficult to assess the individual contribution of the modifications discussed above. However, we did perform manual examination on a sample of the submission output to get a sense of the effect of name and pronoun rewriting.

Overall, we observed that felicitous rewritings outnumbered the errors which were introduced. Still, we did encounter occasional significant mistakes, for instance in the well-known difficult case where discrete named entities with extremely similar names occur close together. In the summary for document set 155 ("Who is JFK, Jr.?"), our system attempts to rewrite the name canonically, with disastrous results:

Q: "Who is JFK Jr.?"

A: **President John F. Kennedy** was traveling with his wife, Carolyn Bessette Kennedy, and sister-in-law, Lauren Bessette, to a family wedding when their plane disappeared over the ocean on its way to Martha's Vineyard, Mass. ...

However, such errors were outnumbered by successful rewritings, even when two similarly named individuals are involved. Figure 2.5 shows our summary for document set 192 ("Who is Sonia Gandhi?"), where the system navigates rewriting in the context of two Gandhis (Rajiv and Sonia) quite well. As can be seen, official titles (when identified) are rendered with the first mention of a specific named entity, and shortened forms are used for subsequent references to the same named entity.

Our modified DefScriber system performed well in the automated ROUGE evaluation, achieving the top score among the 23 automated systems on all ROUGE variants except ROUGE-4. According to the ROUGE confidence intervals, we significantly outperformed 19 of the 22 other systems on ROUGE-1, and 20 of 22 on ROUGE-L and ROUGE-W-1.2 ($P < .05$). No ROUGE model produced scores where DefScriber scored significantly worse than any other system.

On the manual evaluation metrics, our results were closer to average (Table 2.4). However, using a paired T-test, we found there were fairly few significant differences ($P < .05$), so it appears that these evaluations may not have been very discriminative. On the coverage question, we rank sixth but no system is significantly better; on the responsiveness question

Summary Before Rewriting	Automatically Rewritten Final Output
Gandhi, who married into what was once India's most powerful political family, is the first non-Indian since independence 50 years ago to lead the Congress.	**Congress President Sonia Gandhi**, who married into what was once India's most powerful political family, is the first non-Indian since independence 50 years ago to lead the Congress.
After **Rajiv Gandhi** was assassinated in 1991, **Sonia Gandhi** was persuaded by the Congress to succeed her husband to continue leading the party as the chief, but she refused.	After **Prime Minister Rajiv Gandhi** was assassinated in 1991, **Gandhi** was persuaded by the Congress to succeed her husband to continue leading the party as the chief, but she refused.
The BJP had shrugged off the influence of the 51-year-old Sonia Gandhi when she stepped into politics early this year, dismissing her as a "foreigner."	The BJP had shrugged off the influence of the 51-year-old Sonia Gandhi when she stepped into politics early this year, dismissing her as a "foreigner."
Sonia Gandhi is now an Indian citizen.	Sonia Gandhi is now an Indian citizen.
Mrs. Gandhi, who is 51, met her husband when she was an 18-year-old student at Cambridge in London, the first time she was away from her native Italy.	**Gandhi**, who is 51, met her husband when she was an 18-year-old student at Cambridge in London, the first time she was away from her native Italy.

Figure 2.5: An example of rewriting in our question-focused summary for document set 192 ("Who is Sonia Gandhi?")

we score significantly better than two systems and worse than one.

In the Quality Questions, we tended to fall in a large middle group of about ten systems, with one or two systems (not always the same systems) standing out as significantly worse or better on each question. Interestingly, we did not fare especially well on the questions which specifically ask about the quality of noun phrase references. On questions four (which asks whether noun phrases should have been in a longer form) and five (which asks the opposite), we were only average (significantly better than three systems, worse than three and four respectively). Given its good performance in extensive evaluations (Nenkova, 2006), we feel it is likely that our rewriting step is helping our scores on these questions, but the precise impact is difficult to assess without having scores for a non-rewritten summary.

Manual Metric	Our Rank	Sig. Worse	Better
Mean Coverage	6	2	0
Mean Responsiveness	8	2	1
Qual Question 1 (Coherence)	9	1	2
Qual Question 2 (Relevance)	7	0	3
Qual Question 3 (Redundancy)	7	0	1
Qual Question 4 (Anaphora Resolution)	10	4	3
Qual Question 5 (Anaphora Use)	6	3	3
Qual Question 6 (Grammaticality)	10	1	5
Qual Question 7 (Basic Formatting)	2	7	0
Mean Qual Quests	7.3	1.9	2.4

Table 2.4: Our results, rankings, and the number of systems doing significantly worse and better than ours for the manual metrics evaluated on task 5. (15 automatic systems were evaluated manually.)

2.4.3 Topic-Focused Questions

Motivated in part by our success in adapting DefScriber for the biographical responses of DUC 2004, we decided to further adapt the system for a new kind of "topic-focused" question which was introduced as the main task in DUC 2005 (Dang, 2005). We describe in this section the main points of our participation, while our original conference publication provides more detail (Blair-Goldensohn, 2005).

In the case of top-focused questions, we were not entirely certain that DefScriber would adapt as well this time; for biographical summaries, we could essentially apply the definitional approach to "define" a person, whereas for topics like those in Table 2.5 and Figure 2.6, there is not a clear single entity on which to focus a "definition." Although the topic title sometimes gives a fairly concise topic description, we need also to pay attention to the extended question, which can contain a number of detailed subtopics.

In order to handle this new task, we made several adaptations to DefScriber's design, mainly in the identification of Non-Specific Definitional (NSD) sentences. In this task, filtering the documents to determine which sentences could be considered as NSD, or generally

Title VW/GM Industrial Espionage

Question Explain the industrial espionage case involving VW and GM. Identify the issues, charges, people, and government involvement. Report the progress and resolution of the case. Include any other relevant factors or effects of the case on the industry.

Title Fertile Fields

Question Discuss making and using compost for gardening. Include different types of compost, their uses, origins and benefits.

Table 2.5: Some example topics from the DUC 2005 test set (d311i, d694j).

Title Threat to Wildlife by Poachers

Question Where have poachers endangered wildlife, what wildlife has been endangered and what steps have been taken to prevent poaching?

Answer If African elephants are to be saved, the economic return on elephant farming must be increased, rather than lowered, perhaps by granting export quotas to countries willing to invest in keeping the poachers out. The area on either side of the river teems with elephants and other game, now threatened by poachers. Kenya banned big game hunting in 1977 and this year said poachers would be shot on sight. Officials from Kenya's Wildlife Service, who have won plaudits worldwide for their anti-poaching efforts, say they need freedom to cross borders when pursuing poachers and smugglers. Tourists pay millions of dollars a year to come and see Africa's wildlife – and smugglers pay millions more in strictly illegal purchases of ivory and rhino horn from poachers. Until recently, rural communities were not allowed to make any use of wildlife on their lands - poaching was common, either for food or to stop animals destroying crops and endangering people. The number of poached carcasses of elephants and black rhinos in Luwangwa fell by 90 per cent between 1985 and 1987. Poaching has wiped out all but - at an optimistic estimate - 500 of Zimbabwe's rhinos; four years ago there were more than 2,000. Three have been shot already, and even more have been killed in Zimbabwe, the only other country with a shoot-to-kill policy toward poachers. Euan Anderson, a Zimbabwean vet, believes that since the dehorning programme started in Zimbabwe, four to five dehorned rhinos have been killed by poachers.

Figure 2.6: An example DUC 2005 topic (d407b) and DefScriber's answer.

relevant, for these extended topic-focused questions required several key changes.

The main feature used to determine sentence relevance for definitional and biographical questions is the density in and around the sentence of words from the target name or term, i.e. the X in the "Who/What is X?" question. In that setting, sentences in areas of high term density are usually classified as relevant. Our algorithm also gives smaller weight to other features, such as sentence length and position in a document.

However, for the DUC 2005 task, the complexity of the topic statements posed a significant challenge. Not only did we have a much more complex question to deal with, but also little training data around which to design and evaluate relevant-sentence detection algorithms (only two sample question-answer pairs were made available beforehand). Given these limitations, we combined several straightforward, robust techniques to identify question-relevant sentences. Rather than attempting to parse these complex questions at a deep level, we consider them as a bag of words (with terms from the question "title" essentially counted twice), and use IDF weighting over all terms in the question to scale their contribution to a given sentence's term density score.

Using these techniques, we implemented an algorithm for determining on a per-sentence basis which sentences in the document set were relevant to a given topic statement. The algorithm made two passes over each document, on the first pass assigning relevance scores to each sentence based on overlap with topic terms (using weighting as explained above). In the second pass, these scores were adjusted using the first-pass scores of nearby sentences, and sentences scoring above a certain cutoff were judged relevant (additional sentences would be kept if less than 30 sentences were above the cutoff score).

In addition to the changes for relevant-sentence selection, we also made a change to the Answer Creation step of the DefScriber pipeline described earlier. The change here involved disabling the use of the top-down strategy which attempts to place Genus-Species sentences containing "is-a" type information about the term/individual being described at the start of an answer. The reason for disabling this technique is that it assumes that a single entity is being described in a response. Given the topics which we saw in the training set, it seemed that this was not likely to be the case for most topics, and that following this heuristic was likely to decrease the relevance of our answers[9].

[9] In post-hoc analysis of the questions in the DUC 2005 topic-focused task, we determined that enabling the Genus-Species top-down strategy could in fact be useful for these questions as well, by considering the

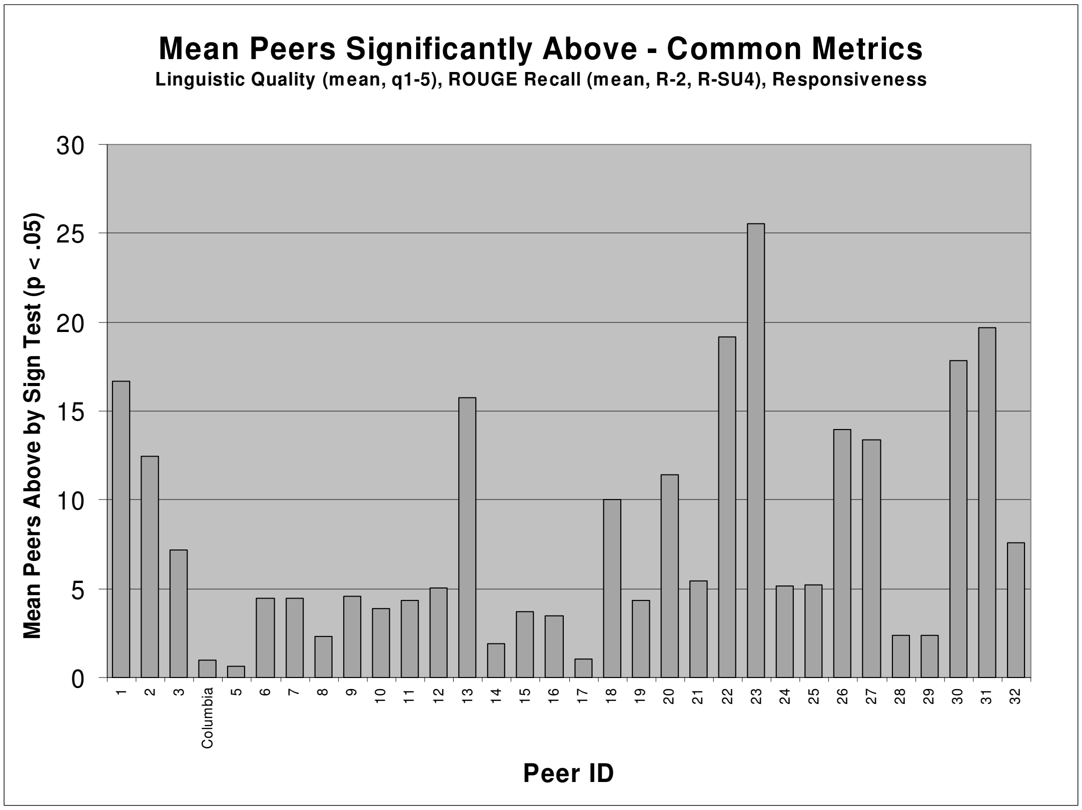


Figure 2.7: Mean systems-above rank across responsiveness, mean of ROUGE-2 and ROUGE-SU4 recall, and mean of linguistic quality questions 1-5. DefScriber is the fourth peer from the left, marked as "Columbia." Note that lower scores are better.

We first performed an informal analysis of our system's answers, and found that DefScriber was largely successful in preparing responsive topic summaries. An example question/answer pair is shown in Figure 2.6. We can see that on this question our main modifications in DefScriber to identify topic-relevant sentences are successful: the output clearly covers material relevant to the topic statement. In addition, we can see that the base data-driven techniques inherited from DefScriber also appear effective here, with clustering

question "title" to be the target term which should appear as the subject in our "is-a" patterns. While we find fairly few matches, since the question titles (e.g. "Threat to Wildlife by Poachers") are often unlikely sentence subjects, when we do find them such sentences can actually be quite good. In our participation in the topic-focused DUC 2006 task, described in Chapter 4, we run our system with Genus-Species enabled in this way. Figure 4.3 in that chapter shows an example of Genus-Species being used to good effect in a topic-focused question.

helping to avoid redundancy, and lexical cohesion for putting related parts of the answer together (e.g., sentences about Zimbabwe).

In addition to this informal examination of our results, we performed various statistical tests to determine the significance of the quantitative results distributed by NIST. We provide further analysis in our original DUC 2005 publication (Blair-Goldensohn, 2005); here we present only the combined metric which we propose for viewing several aspects of results in a single number.

To get an overall picture of which systems performed consistently well across the various metrics, we propose a three-way mean of systems-above ranks for ROUGE scores, linguistic quality questions, and responsiveness questions. Motivations for using this particular combination include: (1) it combines measures of presentation and content (2) it combines automatic and manual scores (3) it uses scores for which there was a significant level of difference found between systems (i.e. different systems are at the top of the ranking on each of these individual metrics) and (4) it uses scores for which all systems were rated (Pyramid scores were not computed for all systems and are thus excluded).

The results on this combined metric are shown in Figure 2.7. Note that we report the results of these rank tests in terms of the number of peers ranked significantly *above* (rather than below) a given system[10], since we believe it is more interesting to know how close to the best performers a given peer is, rather than how many lower performing systems are below it. This means that *lower* numbers are better, since the best systems have no peer ranked above them.

We find that DefScriber, a strong (but not outstanding) performer in each individual metric in the combination, measures second best in the combined measure, slightly behind peer 5 and slightly ahead of peer 17. We take this as validation that our system as adapted for the topic-focused task does indeed succeed at balancing coherence and content considerations.

[10] As detailed in our DUC 2005 publication, we use the sign test for significant rank differences, and a two-way ANOVA to determine significant effect.

2.5 Integrating Speech Input

2.5.1 Motivation

Finding the "correct" information in text documents when answering a long-answer question is hardly a solved problem. Yet a system which has no allowance for documents from other modalities risks missing whole swaths of information which may not appear in a text document at all. Given that using all available data is important, we need a strategy to do so.

As mentioned in this chapter's related work section, a baseline strategy that has proved workable in TDT-like clustering (as well as other fields, such as MT) is simply to run speech documents through a recognizer and treat the resulting transcripts just as if they originated as text documents. Yet for the task of extractive response generation, as in DefScriber, it is clearly preferable to use text from original documents over less fluent ASR output when the two have similar semantic content. Even when word-error rate is low, minor corrections or omissions in the ASR can interrupt the flow of a written document. (By the same token, in the context of a speech interface, it would be preferable to output a clip from a broadcast segment, rather than a relevant text snippet which has been run through TTS.) In such cases, it is especially critical to correctly identify which speech and text units should be clustered together.

Thus, the clustering module in DefScriber, which serves to identify information which covers the same answer subtopic, becomes especially important in this case. That is, if we can correctly detect that a text and speech sentence cover the same theme in the answer, we can not only avoid outputting redundant information, but we can make sure that the sentence we do include is optimal for the output modality, i.e. text. With this in mind, we choose to examine the subproblem of text-speech clustering itself.

While the precise problem we face in DefScriber is to cluster sentences, we do not have a corpus of mixed speech-text clustered sentences which we can use for training. However, we happily do have such a corpus at the document level, namely the TDT[11] corpus which

[11] An overview of the Topic Detection and Tracking (TDT) project is described by Wayne (Wayne, 2000). A number of project resources are also available online from `http://projects.ldc.upenn.edu/TDT/`.

includes both text and speech documents which have been assigned to human annotated topic clusters.

Thus we can study the problem at the document level, with the intuition that whichever techniques prove useful in clustering same-topic documents will have a good chance of translating onto the task of clustering same-subtopic sentences.

2.5.2 Metrics

A major consideration in evaluating this task is the lack of a commonly accepted metric for cluster quality[12], particularly in cases where a proposed set of classes K may not match the true number of clusters C. To this end, we adopt Rosenberg's proposal (Rosenberg, 2006) for a metric which extends based on Dom's work (Dom, 2001). This measure, called "Validity," combines two complementary aspects of cluster quality, which he terms "Homogeneity" and "Completeness". Intuitively, Homogeneity can be understood as something somewhat akin to precision in that it measures the degree to which a given class $k \in K$ contains only objects from a single true cluster $c \in C$; Completeness is somewhat akin to recall in that it measures the degree to which all objects from a single cluster $c \in C$ are grouped in the same class. It is the addition of the Completeness calculation that sets the Validity metric apart from other commonly used measures like cluster "Purity", which increase as the number of clusters increases, up to the degenerate case where we have maximum Purity resulting from all singleton clusters.

The scores for Homogeneity and Completeness are calculated using conditional entropy; since the precise derivations are somewhat involved and are not a contribution of this thesis, they are given in Appendix A. For interpreting these results, it is sufficient to understand that both measures can vary from zero to one, where a "perfect" clustering, i.e. where C and K are isomorphic, will result in $C = K = 1$. Trivially, one can get perfect Homogeneity by proposing all-singleton clusters; perfect Completeness can similarly be achieved by proposing one cluster containing all items; thus we follow Rosenberg's proposal in evaluating with respect to the harmonic mean of the two measures, Validity.

[12] See Oakes (Oakes, 1998, p.120) on the lack of a clear metric despite the importance of evaluating cluster quality.

2.5.3 Evaluation

Using these evaluation measures, we conduct the following experiment. We first create a data set consisting of 69 document clusters containing 1841 total documents. Cluster sizes range from one to 301 documents, with a median size of 13 documents. This set is extracted from the original TDT-4 topic annotations using the following criteria: (1) non-English documents are dropped since these introduce additional variables into our experiment which are undesirable at this point (i.e., quality of non-English ASR) (2) multiple-topic documents are dropped to simplify the evaluation to include only disjoint clusters (3) clusters without any speech documents are dropped, in order to focus our experiment on those clusters where our experiment is of interest. From the 69 clusters which result from applying these, we randomly select 62 document clusters for testing.

We perform the clustering as in DefScriber, using the Nearest Neighbor algorithm, using stemmed-word vectors as our document representation, and IDF-weighted cosine distance as our similarity measure. Recall that in DefScriber, the units which we are to cluster are sentences, which are represented in the clustering algorithm by term-frequency vectors that are compared based on IDF-weighted cosine distance; here, the units are full documents, but we can just as easily represent them with such vectors and apply the same distance metric.

We implement three evaluation conditions, which vary the way in which the vectors for input speech documents are assembled:

ASR (baseline) In our baseline technique, the ASR transcript[13] is interpreted no differently than a plain text document; that is, for each occurrence of a word in a document, the vector count for that word is incremented by one.

Confidence-weighted ASR In our confidence-weighted setting, we use the word-recognition confidence for each word in the ASR output and increment the vector frequency proportionally; that is, if a word in the transcript is recognized with a confidence probability of, say, 0.5, the vector count for that word is incremented by 0.5.

[13] For all experiments with TDT ASR, we use the data from the Dragon ASR transcript which is included with the TDT distributions, available from the Linguistic Data Consortium (`http://www.ldc.upenn.edu`)

Configuration	Homogeneity	Completeness	Validity
ASR Baseline	0.780	0.668	0.720
Confidence-weighted ASR	0.758	0.705	0.731
CC Transcript	0.928	0.952	0.928

Table 2.6: Cluster Homogeneity, Completeness and Validity for speech and text clustering task. We show here results at peak Validity; Figure 2.8 shows a wider swath of tradeoffs between these metrics.

CC Transcript We use the close-captioning (CC) transcript to provide an upper bound for perfect recognition. In this setting, we can treat the transcript as an ASR output with a perfect confidence for every word. (Although we note that, in fact, such transcripts do contain occasional typos.)

Table 2.6 shows the performance of our clustering on the TDT document data as evaluated on Homogeneity, Completeness and Validity and Figure 2.8, shows the results across the set of similarity thresholds evaluated. The peak Validity for the three configurations occurs at three different similarity thresholds (0.15 for Transcript, 0.3 for Confidence-weighted ASR, and 0.35 for ASR), indicating that the similarity threshold in our clustering method can be adjusted to provide Homogeneity or Completeness, but that there is a tradeoff.

As expected, the performance on the close-captioned transcript data is significantly higher; we view this as a measure of the upper limit of performance with purely lexical features in this task.

Lastly, we note that while this evaluation focuses on cluster-level quality, that we have integrated the module for processing ASR documents into an end-to-end version of DefScriber which we have tested on biography and definitions question as part of the DARPA GALE program[14]. In that work, which is ongoing, we have observed overall answer improvement from the inclusion of ASR documents, demonstrating that such documents can and do contain useful information which can be integrated into our answers. However, in part because of limitations of the corpus and currently available data, we have not yet carried out formal experiments which isolate the end-to-end effect of the methods for improved

[14] An overview of the GALE project is available at: `http://projects.ldc.upenn.edu/gale/`

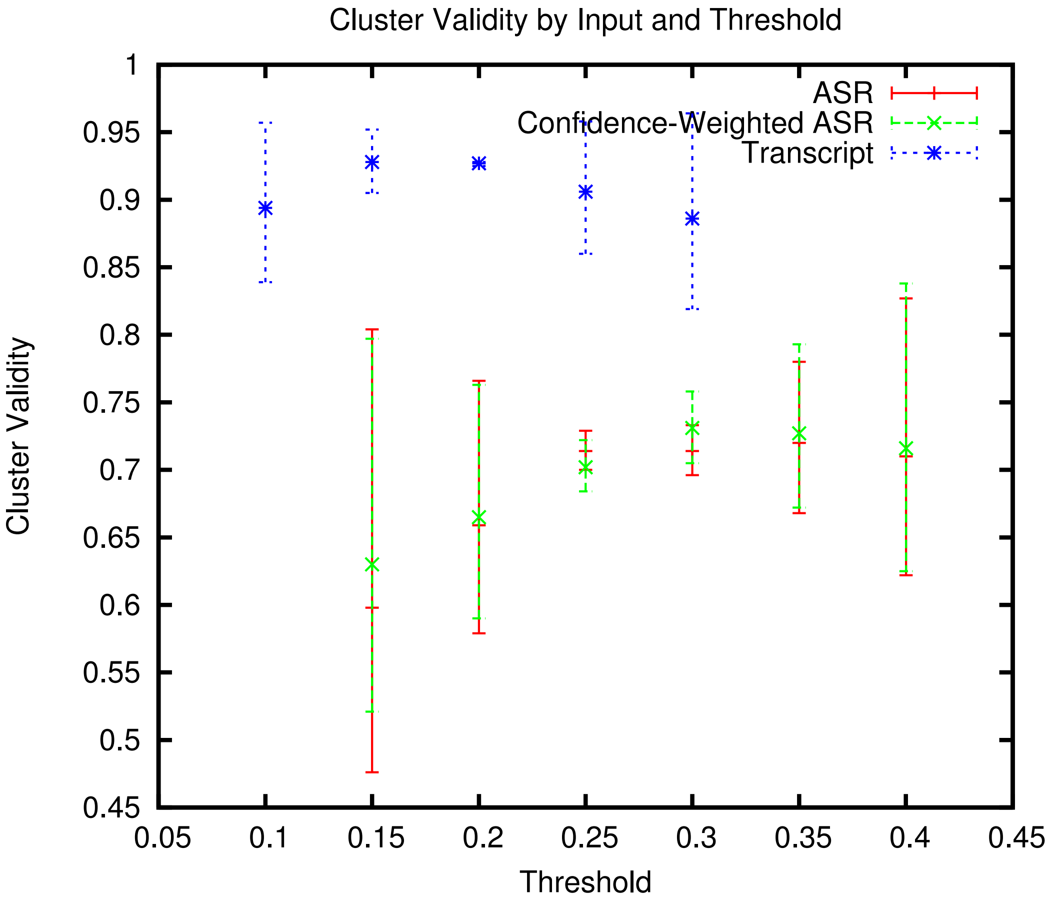


Figure 2.8: Cluster Homogeneity, Completeness and Validity for speech and text clustering task. The error bars show Completeness and Homogeneity, while the midpoints show Validity (at lower thresholds Completeness dominates; at higher thresholds Homogeneity does)

clustering of speech and text presented here. Thus our evaluation of this component is currently only at the module-level, leaving an assessment of impact on end-to-end performance to future work.

2.6 Conclusions and Future Work

DefScriber is a novel contribution of this thesis on several levels. Firstly, it is among the first NLP systems to attempt the task of creating dynamic, multi-sentence answers to definition and biography questions. Second, it uses a novel approach to answering these questions.

Namely, DefScriber takes a “hybrid” approach which combines top-down rules that seek specific types of information that “should” be in a definition/biography, and combines these with bottom-up methods, which shape answer content based on statistical analysis of the themes present in the input.

On the one hand, we have seen that the hybrid approach works well according to various evaluation metrics. When the specific Genus-Species predicate information is found, our evaluations have shown that it provides strong grounding for the rest of the answer, giving the overall answer passage good structure. Yet the bottom-up techniques are sufficiently robust that whether or not we match a specific predicate, we can put together a set of relevant sentences so as to form a high-quality answer.

On the other hand, there are issues of fragility and lack of coverage in our pattern-based predicate identification approach. Our early experiments showed the precision to be quite high, on the order of 0.96. Thus, when it finds matches, it works well. But while recall has not been formally assessed (since we do not have anything to call a “representative corpus” of definitional text marked for all Genus-Species realizations), anecdotally we find that the system can miss Genus-Species information for various reasons such as lack of pattern coverage, parsing failures, or because information is implied rather than explicitly stated. Another issue is that our method is not easily portable for other languages or relation types, and involves significant manual effort to increase coverage.

With regard to our work with speech and text clustering, we are encouraged by our results, which show this technique achieving improvements using the Validity measure of clustering performance. We take this as evidence that, as expected, the information generated in the decoding process of ASR can improve the performance on probabilistic tasks like clustering. While we achieve only modest improvement here, we believe that this is due at least in part to the straightforward way in which we integrate speech features into our system. For instance, future work might consider using the per-word confidence output by the ASR engine in a more nuanced way than our current linear weighting. Similarly, given that word recognition errors may occur more frequently near disfluencies or in an otherwise non-independent fashion, a measure which considers the recognition confidence of nearby words is another possibility.

Overall, this chapter shows that DefScriber has achieved strong results. Yet there is certainly further work to be done. Our experiences in developing the system have motivated us to pursue new approaches to identifying rhetorical and semantic relations which can be used both within DefScriber and beyond. We describe our work in these directions in the remaining chapters of the thesis.

Chapter 3

Rhetorical-Semantic Relations

3.1 Introduction

As humans, we understand relationships in the physical realm, such as that between thirst and liquid, and in the meta-physical, such as that between loss and sadness. In language, these relationships are reflected in the way we understand and use particular words, and thus can be thought of as holding between words themselves. Some word relations have an hierarchical nature, such as hypernyms (*foxhound* and *dog*), or synonyms (*dog* and *canine*). Other relations, such as contrast (*simple* and *difficult*) and cause (*disaster* and *suffering*), intertwine semantic and rhetorical aspects. These relations express a semantic connection which exists outside of a given document or discourse; at the same time, they are often used to invoke or emphasize a rhetorical strategy. In this chapter, we focus on this latter group of word relations, which we refer to as rhetorical-semantic relations (RSRs)[1].

Often, such relations are signaled at the rhetorical level by cue phrases such as *because* or *however* which join clauses or sentences and explicitly express the relation of constituents which they connect:

Example 1 *There have been a spate of executive resignations at the company, in part* **because of** *the recent accounting scandals.*

In this example, *spate of executive resignations* is linked explicitly to *recent accounting*

[1] We define RSRs formally in Section 3.3.

scandals. But such relations may also be implicitly expressed:

Example 2 *The administration was once again beset by scandal. After several key resignations, ...*

Here, the causal relationship between *scandal* and *resignation* is equally apparent to the human reader, but is not explicitly signaled. Our world knowledge that resignations often follow scandals comes into play; so do compositional factors, such as the fact that an *administration* is discussed, which pragmatically fits the concept of resignation more than, say, a marriage (even though the latter could also be beset by scandal).

In this chapter, we examine the problem of detecting such relationships, whether or not they are indicated by a cue phrase, using evidence derived from lexical context. We do this by extending the learning framework of Marcu and Echihabi (Marcu and Echihabi, 2002) (M&E), which proposes to mine a large corpus for relation instances by using cue phrases such as *because of* in Example 1 above. They use these instances to learn automatically that word pairs like *scandal* and *resignation* are strongly associated by a given relation. We formally describe the M&E learning model in Section 3.4.

While the other chapters in this thesis focus on user-facing applications, we devote this chapter to our research on building a model of RSRs which can act as a "behind-the-scenes" resource for such applications. By studying in detail the issues of implementing and improving such a resource, rather than relying completely on tools compiled by others, we gain insight into the nature of this model, and can also customize its design in view of our application desiderata.

In the remainder of this chapter, we first review related work in both theory and applications (Section 3.2). Using this context, we provide a definition of RSRs (Section 3.3) and explain our choice to build a relation model that extends the M&E learning framework discussed above (Section 3.4). In Section 3.5, we describe our implementation of TextRels, a system which implements this framework, and which we use as a platform for several experiments in improving model quality.

First, we optimize the set of basic parameters such as smoothing weights, vocabulary size and stoplisting (Section 3.5.1). We then explore two complementary techniques for filtering our automatically-mined training data.

In Section 3.5.2, we experiment with using topic segmentation to constrain our cue-phrase heuristics for mining RSR instances. In Section 3.5.3, we use syntactic features to trim irrelevant material from our training instances and thus improve the quality of our learned models. Using evaluation over both automatically-created and human-annotated testing data, in each experiment we observe significant improvements in terms of relation classification performance.

3.2 Related Work

3.2.1 Rhetorical and Discourse Theory

Work on rhetorical and discourse theory has a long tradition in computational linguistics (see Moore and Wiemer-Hastings (Moore and Wiemer-Hastings, 2003) for a comprehensive overview). Areas of research include informational and semantic relationships (Hobbs, 1979; Mann and Thompson, 1988; Martin, 1992); processing models in discourse, including attentional and intentional structure (Grosz and Sidner, 1986); as well as work that takes models of both information structure and processing into account for modeling (Moser and Moore, 1996; Asher and Lascarides, 2003) or generating (McKeown, 1985; Moore and Paris, 1993) text.

As various researchers have noted (Hovy and Maier, 1993; Moore and Pollack, 1992; Moser and Moore, 1996; Marcu and Echihabi, 2002), taxonomies of rhetorical and discourse relations may differ in places, yet tend to agree in terms of core phenomena for which they account. For instance, Moser and Moore argue convincingly that Mann and Thompson's "nuclearity" and Grosz and Sidner's "dominance" may be alternative perspectives on the same phenomenon. And Hovy and Maier present a kind of meta-theory which fits various divergent taxonomies into a surprisingly unified framework, e.g. by showing that many relations including "Violated Expectation" (Hobbs, 1990), "Qualification" (Dahlgren, 1988) and "Negative Argument-Claim" (Sanders et al., 1992) can be "merged" as a single "Concession" relation.

Yet even while the details of the various relation theories are debated, the basic tenets of rhetorical and discourse theory – essentially, that texts consist of discourse segments which are related in classifiable ways which influence their meaning – have formed the basis for applications that aim to model the informative content of text, such as text summarization (Marcu, 2000) and essay-grading (Burstein et al., 2001). Models of attentional and informational structure, particularly the influential work of Grosz and Snider (Grosz and Sidner, 1986) (henceforth G&S), have been particularly useful in generation and discourse. For instance, Elhadad and McKeown (Elhadad and McKeown, 1990) showed attentional structure to be important for generation of connectives, while intentional structure is often used in response generation and planning for dialog systems (Moore and Paris, 1993).

A distinction which is sometimes made with regard to both rhetorical and discourse relations concerns the *informational* versus *intentional* aspects of such relations. *Informational* relations describe the semantics of *how* their constituents relate intrinsically; *intentional* relations focus on the status of information with respect to speaker, hearer and dialog-level goals. As Moser and Moore (Moser and Moore, 1996) point out, informational relations are important for the generation of text, even while *intentional* relations (such as G&S' "satisfaction-precedes" constraint) defined at a broader level are often more useful and clearly distinguishable for the reader/hearer.

At the resource level, there have traditionally been few resources for corpus-based study of rhetorical and discourse structures. Until recently, the main such resource was RST Bank (Carlson et al., 2001), a discourse-tagged corpus in the RST framework. However, in recent months the first release of the Penn Discourse TreeBank (PDTB) (Webber et al., 2005; Prasad et al., 2006) has provided the community with a much larger and more comprehensive resource which annotates both explicit and implicit discourse relations. While the release of PDTB largely postdates the work in this thesis, we are able to use it in our work for evaluation purposes.

3.2.2 Lexical Relations

Work in lexical relations examines relationships which hold between individual words, rather than at a discourse or document level. Studies of such relations include a broad swath of theoretical literature, from Pustejovsky's linguistically-oriented work on the generative properties of word meaning (Pustejovsky, 1995), to Lakoff's more cognitive and philosophical approach to the importance of metaphorical relations as a tool for both expressing and comprehending experience (Lakoff, 1987).

One important set of lexical relations are well-known types like synonyms, antonyms and hypernyms. These kinds of relations are referred to by Morris and Hirst (Morris and Hirst, 2004) as "classical." In their definition, classical relations are constrained in that two related words must have some shared "individual defining properties" and belong to the same syntactic class. For example, the antonyms *asleep* and *awake* clearly fit this definition.

Many of the studies on relation modeling and classification mentioned in the next subsection focus on these "classical" relations because of the availability of several large-scale, manually coded taxonomies, perhaps most notably WordNet (Fellbaum, 1998). WordNet covers a broad scope of language, and uses the lemma/part-of-speech combination as the primary object to which the kind of "classical" relations mentioned above can be attached.

Additionally, while not as broad in linguistic coverage, FrameNet (Fillmore et al., 2002) and PropBank (Palmer et al., 2005) provide a more usage-based, compositional account of relations at the sentence level. FrameNet and Propbank focus on semantic frames and verb-argument structures, respectively, as primary objects and create ontologies which map these individual cases onto a standardized set of roles.

However, while these thematic roles and "classical" relations have been the subject of much research in computational linguistics, Morris and Hirst (Morris and Hirst, 2004) point out the importance of "non-classical" relations as well. These are relations which, in contrast with "classical" lexical relations, may be difficult to list *a priori* because they cross word classes or do not share an obvious "individual defining property," perhaps because of a relationship which becomes clear only in a given context. Interestingly, Morris and Hirst report that readers more frequently identify such links than they do classical ones. They describe a study in which participants reliably notice cause-like relations between intrinsically dissimilar words like *alcoholic* and *rehab*, or identify a context-sensitive relationship between *homeless* and *brilliant* as used to contrast negative circumstance with a positive character trait.

These findings highlight some of the important gaps, perhaps most concisely pointed out by Pustejovsky (Pustejovsky, 1991), in verb- or role-oriented semantic representations (such as PropBank) which do not account for adjectival or nominal relationships, and static networks such as WordNet which do not account for context and composition.

3.2.3 Automatic Relation Modeling

While the above survey aims to provide theoretical context for our work with Rhetorical-Semantic Relations, the implementation and experiments in this chapter also have many connections to previous work.

A significant amount of applied work in modeling and classifying textual relations is in the categories of the more strongly structured, "classical" lexical relations and semantic roles discussed in the previous subsection. Generally speaking, this work aims to distill patterns, rules or features which can detect the relation of interest, using WordNet (Hearst, 1998; Snow et al., 2006) or PropBank (Gildea and Jurafsky, 2002; Chen and Rambow, 2003; Hacioglu et al., 2004) for training and/or test data.

Other work in relation modeling differs in resources and goals. For instance, other structured resources used to find lexical relations include dictionary definitions (Richardson et al., 1998) or hyperlinked web pages (Glover et al., 2002). Some experiments rely on supervised classification over human-annotated examples, e.g. in Girju et al.'s studies of on part-whole (Girju et al., 2003) and causation relationships (Girju, 2003).

Another popular approach is bootstrapping, where a set of known relation instances (Ravichandran and Hovy, 2002; Agichtein and Gravano, 2000; Snow et al., 2005) or relation-associated patterns (Hearst, 1998; Berland and Charniak, 1999) are used to seed a process which learns additional instances and/or patterns.

Marcu and Echihabi (Marcu and Echihabi, 2002) classify RST-like relations, using a small set of manually identified cue phrases to mine relation instances from a large, unannotated corpus. Their approach resembles bootstrapping, although they do not iterate the process to learn additional cue phrase patterns. This work was a major leap forward in the problem of automatic RST modeling, because it allows large amounts of training data to be collected relatively cheaply, resulting in a model of relation content with extremely broad coverage.

Other attempts to automatically identify rhetorical and discourse structure have limited their scope to achieve advances in more specific areas. For example, Marcu's work (Marcu, 1997) in creating a cue-phrase-based discourse parser constructs full-document parse trees, but does not take into account the informational content of documents which it analyzes. Soricut and Marcu (Soricut and Marcu, 2003) do make some content-based decisions by analyzing points of connection in discourse trees, but focus only on sentence-internal relations. In an alternative approach, Barzilay and Lee (Barzilay and Lee, 2004) develop domain-specific models of document structure rather than trying to fit an *a priori* set of relations a la RST. They learn these models automatically from a training corpus of newswire documents in a small set of manually defined topics such as earthquakes and financial reports.

The approach of Marcu and Echihabi (Marcu and Echihabi, 2002) can be viewed as a refinement of earlier data-driven approaches like Latent Semantic Analysis (LSA) (Deerwester et al., 1990), which derives conceptual representations for terms based on the contexts in which they appear. However, in addition to differences in representation (LSA uses dimensionality reduction to abstract across individual terms), the crucial difference is that LSA does not distinguish between different kinds of relationships. Thus, it can offer the information that two words are "related," but the manner of that relation is unspecified. Another variable is that the relatedness measure can vary depending on how a word's "context" is understood – different implementations may construe a context segment to extend to the sentence, paragraph or even document level (Landauer and Dumais, 1997). Nonetheless, LSA-based measures have been shown to correlate with human judgments of general text coherence (Foltz et al., 1998).

Topic segmentation and syntax have also been used in efforts to detect lexical and discourse relations. Galley et al. (Galley et al., 2003) show that a lexical cohesion-based segmentation approach is a successful component in mapping discourse segments in meeting transcripts. Higgins et al. (Higgins et al., 2004) also consider the contribution of topic segment strength in identifying portions of connected text for an essay grading task.

Syntactic features have been previously in detecting various relations. Marcu and Echihabi (Marcu and Echihabi, 2002) report success in a limited experiment which uses Part-of-

Speech tags to filter their training data to include nouns and verbs only. Soricut and Marcu (Soricut and Marcu, 2003) use syntactic parsing to identify and classify sentence-internal RST structures. Syntax is also used by Lapata and Lascarides (Lapata and Lascarides, 2004) to focus a study of temporal relations by focusing on the relationship between main and subordinate clauses. Others have shown that syntactic features can be used to refine measures of general semantic similarity (Lin, 1998a; Lee, 1999; Padó and Lapata, 2003), as well as helping to detect and infer "classical" WordNet-like lexical relations (Padó and Lapata, 2003; Widdows, 2003; Snow et al., 2005).

3.3 Defining Rhetorical-Semantic Relations

Until this point, we have referred to Rhetorical-Semantic Relations (RSRs) informally. In this section, we define explicitly what we mean by RSRs, providing a grounding for RSRs in terms of existing theories. Our theoretical perspective on RSRs draws on the central intuition of Marcu and Echihabi (Marcu and Echihabi, 2002) (M&E) regarding which relations to model and at what theoretical level. Their idea is to abstract away the fine-grained differences between alternative discourse theories discussed in Section 3.2.1, and instead model relations at a more coarse-grained level. This approach has many benefits. First, it reduces problem size (in terms of potential number of relations to model) and complexity (in terms of specificity/granularity of relation definition). Second, it allows us to select a set of relations which we can define at the theoretical level through reference to a number of previous theories.

We choose to model three RSRs: Cause, Contrast and Adjacent. Following M&E (as well as Hovy and Maier (Hovy and Maier, 1993)), for each RSR we provide an intuitive definition in Table 3.1 via a list of example relations from existing taxonomies. As in M&E, our intuitive definition for these relations can be understood as a union of the listed relations.

For the Cause and Contrast relation, we aim to replicate the intuitive idea of M&E's Cause-Explanation-Evidence and Contrast relations, respectively (the "comparable relations" in our table are a superset of those listed by M&E for their comparable relations; we

	Cause	**Contrast**	**Adjacent**
M&E	Cause-Explanation-Evidence	Contrast	
Hobbs	Cause, Explanation	Contrast, ViolatedExpectation	
H&M	Ideational-Cause/Result, Interpersonal-Support	Ideational-Comparison/Contrast, Interpersonal-Concession	Textual-Pres-Sequence
K&S	Causal-Positive, Causal-Positive	Causal-Negative	
L&A	Result, Explanation, Narration		
M&T	Evidence, Justify, Volitional-Cause, Nonvolitional-Cause Volitional-Result, Nonvolitional-Result, Purpose	Antithesis, Concession	Presentational Sequence
McKeown	CauseEffect, Evidence	Comparison, Antithesis	

Table 3.1: Rhetorical-Semantic Relations and comparable relation types from previous work: Hobbs (Hobbs, 1990), H&M (Hovy and Maier, 1993), K&S (Knott and Sanders, 1998), L&A (Lascarides and Asher, 1993), M&T (Mann and Thompson, 1988), M&E (Marcu and Echihabi, 2002) and McKeown (McKeown, 1985).

provide several additional relations for completeness and additional "votes of confidence"[2]). One important aspect of discourse theories which is collapsed in these unioned definitions is the distinction between rhetorical/intentional and semantic/informational relation aspects. For instance, in Mann and Thompson's taxonomy, they draw this distinction between "subject matter" versus "presentational" relations (Mann and Thompson, 1988, p.18). Evidence

[2] As argued by Hovy and Maier (Hovy and Maier, 1993), the appearance of a given relation across multiple discourse theories can in some sense be interpreted as a "vote of confidence" that it has resonance and exists in an objective or at least understandable sense.

is in the latter category, because it includes an explicit notion of effecting the reader's belief in a causal relationship, whereas Cause itself applies when there is a "subject matter", or semantic relationship irrespective of a reader's belief.

Unlike Cause and Contrast, the Adjacent RSR departs slightly from the relations which M&E choose to model.

As shown in Table 3.1, this relation is closest to "sequence"-type relations proposed in several taxonomies. In this sense, it can be viewed as a model of the order-influencing tendencies of authors. However, beyond being a pure model of the sequence of likely text, Adjacent can be understood as a model of rhetorical-semantic relatedness that is agnostic as to the specific nature of the relation. This is comparable to the way in which Moser and Moore (Moser and Moore, 1996) analyze Grosz and Sidner's (G&S) (Grosz and Sidner, 1986) model of intentional structure, which provides only two "specific" relationships, namely "satisfaction-precedes" and "dominance." In Moser and Moore's analysis, a more detailed taxonomy like Mann and Thompson's RST can be reconciled with G&S by viewing RST's individual relations as "flavors" of G&S's broader relations. In this sense, Adjacent, whose only specificity has to do with the order of two intentions, can be viewed as a coarse-grained derivative of Grosz and Sidner's "satisfaction-precedes" constraint (Grosz and Sidner, 1986)[3].

In this way, we see Adjacent as abstracting to a level above Cause and Contrast relations in modeling the case where some relation, or even multiple relations, hold, even if the specific identity is undifferentiated. In this sense, we follow Martin's observation that "... simply putting clauses next to each other suggests some logical connection between them, whether or not [it] is made explicit ..." (Martin, 1992, p.165). Adjacent serves as a kind of catch-all for cases where there is *any* relation holding between two sentences. That is, we believe that, given our limited set of relations, as well as the intractability of enumerating or choosing a "complete" or "correct" set of relations, Adjacent can serve as a blanket for capturing

[3] We see Adjacent as more closely related to G&S' "satisfaction-precedes" than "dominance" because the way in which we capture example instances (explained in the next section) is sensitive to their sequential order in the source text.

the connections of various kinds which hold across adjacent sentences[4]. While the Adjacent relation is thus more amorphous than the Cause and Contrast RSRs, we include it because of its complementary role and also with an eye toward its possible usefulness in one of our planned applications for RSR models, namely the task of measuring inter-sentence coherence in DefScriber responses in Chapter 4.

In sum, our chosen set of these three RSRs – Cause, Contrast and Adjacent – is not meant to be a correct set as much as a useful one. That is, we believe this set has several important strengths including (1) prevalence in the literature (2) applicability within DefScriber, as well as other problems of interest to us (3) extractability within a pattern-based framework, i.e. by a recognizable set of cue phrases or surface patterns and (4) sufficient frequency in the training corpora. (The final two points are amplified in the next section.)

Lastly, a question in the mind of some readers may be: Why do we have to create "our own" definition for Rhetorical-Semantic Relations when the literature is already strewn with relation taxonomies? That is, what are RSRs that is different from what has been proposed earlier? There are two important pieces to the answer:

- We *do* avoid creating our own taxonomy where possible. RSRs aim, where possible, to be a unification of previous efforts. Most directly, we adopt Marcu and Echihabi's sense of coarser-grained surrogates for relations such as those in RST.

- As mentioned above, RSRs do not distinguish between informational and intentional relations as such. This is a key aspect of the simplified relations of M&E which we adopt, i.e. that they abstract across the informational-intentional line that is often drawn in other theories using alternate relation versions or tiers of relations. We use the name rhetorical-semantic relations to explicitly evoke this coarser-grained view (whereas, in previous work, the terms "rhetorical" and "discourse" relations tend to connote at least some intentional component, and "semantic" relations a more

[4] This begs the question, how often do adjacent sentences indeed have some rhetorical or semantic link? According to the Penn Discourse TreeBank, which annotates implicit discourse relations over a large set of sentence pairs in its corpus, the answer is "almost always." Over a set of 2003 adjacent sentence pairs *not* joined by an explicit discourse connective, only 53 were annotated as having no relation (Prasad et al., 2006, p.44).

informational view).

3.4 The M&E Relation Modeling Framework

We use the framework of Marcu and Echihabi (Marcu and Echihabi, 2002) (M&E) to model RSRs. We include the details of this model here rather than simply citing their work so that we can discuss particulars of this model, and our additions and extensions to it, more clearly. However, we emphasize that the methods for RSR modeling and classification described in this section are entirely from M&E.

The premise of the M&E approach is common in corpus linguistics: collect a large set of class instances (*instance mining*), analyze them to create a model of differentiating features (*model building*), and use this model as input to a *classification* step which determines the most probable class of unknown instances. We review their framework in this section, describing it in terms of these three stages in order to later compare with our own work.

The intuition of the M&E model is to apply a set of RSR-associated cue phrase patterns over a large text corpus to compile a training set without the cost of human annotation. For instance, Example 1 will match the Cause-associated pattern "Because of W_1 , W_2 .", where W stands for a non-empty string containing word tokens. In the aggregate, such instances increase the prior belief that, e.g., a text span containing the word *scandals* and one containing *resignations* are in a Cause relation.

The M&E model works with only a single such feature, namely the individual word pairs across the cartesian product $W_1 \times W_2$. In our example, this would include apparently common word pairs (e.g., *recent, of*) as well as those which on their face seem more relation-specific (e.g., *scandals, resignations*). The intuition is that over a sufficient number of examples, the latter pair will have a higher relative frequency in causal vs non-causal contexts.

A critical point in this model is that the cue phrase words themselves (e.g., *because*) are discarded before extracting these word pairs, otherwise pairs resulting from cue phrases themselves would likely be the most distinguishing features learned. In this way, the model focuses on content words rather than patterns, and can therefore be used even when no cue phrase is present. Recalling our earlier example, consider:

> The administration was once again beset by scandal. After several key resignations ...

In this case, there is no overt causal marker, but an appropriately trained M&E-style model can use the word pair (*scandals, resignations*) to increase belief that we are nonetheless looking at a causal relationship between the first and second sentences.

In order to explain how M&E put this intuition into practice, we now formally characterize the stages of their framework formally:

Instance Mining Input $= \{T, R, P\}$ where T is a corpus of text documents with marked document and sentence boundaries; R is a set of rhetorical-semantic relations; P is a set of regular expression patterns defined over literal text characters and sentence/document boundary markers; each pattern is associated with an RSR $r \in R$ such that P_r is the set of patterns for finding instances of RSR r; each pattern contains slots for two text spans W_1 and W_2 which are inferred as the constituents related by r in a matching text; A text span W is a string of ordered, space-separated tokens $W = \{w_1 ... w_n\}$.

Output $= I$, a set of instance examples for one or more of the RSR r, such that I_r is the set of text span pairs found by matching P_r against T and storing the captured text spans.

Model Building Input $= \{I, \phi_m\}$ where ϕ_m is a set of parameters for filtering data from I, e.g. by aggregating or discarding certain instances.

Output $= F$, a set of feature/value pairs associated with each RSR F such that F_r is the set of pairs associated with relation r. M&E propose computing a single type of feature, namely the frequency of token pairs derived from taking the cartesian product of $W_1 = \{w_1 ... w_n\} \times W_2 = \{w_{n+1} ... w_m\} = \{(w_1, w_{n+1}) ... (w_n, w_m)\}$ over each span pair instance $(W_1, W_2) \in I$.

Classification Input $= \{F, \phi_c\}$ where ϕ_c is a set of classification parameters, e.g. smoothing constants.

Output = $\{C\}$, a set of naïve Bayesian classifiers. These classifiers determine the most probable relationship between an input text span pair (W_1, W_2). This is determined in our classifier using the prior probability of the individual token pairs $(w_i, w_j) \in W_1 \times W_2$. Thus, the most likely relation r is given as $\text{argmax}_{r \in R} P(r|W_1, W_2)$[5]. The probability $P(r|W_1, W_2)$ is simplified by assuming the independence of the individual token pairs, this is equivalent to: $\prod_{(w_i,w_j) \in W_1, W_2} P((w_i, w_j)|r)$. The frequency counts F_r are used as maximum likelihood estimators of $P((w_i, w_j)|r)$.

3.5 TextRels: Modeling and Classifying RSRs

TextRels is our implementation of the M&E framework for our RSRs Cause, Contrast and Adjacent (described above). In addition, we follow M&E in modeling a fourth set of probabilities for non-related, same-topic text by pulling text span pairs which are simply pairs of sentences from the same document separated by at least three intervening sentences. This relation is known as NoRelSame.

We can describe the implementation of TextRels in terms of the stages of the M&E approach defined in the previous section.

Instance Mining Inputs: The corpus T is the Gigaword newswire corpus of 4.7 million newswire documents[6]. Our relation patterns P for the Cause and Contrast relations are derived from published lists (e.g., (Marcu, 1997; Prasad et al., 2006)) which we have edited and augmented; a sample of the patterns used for each relation are included in Table 3.2 with the full list in Appendix B.

Note that as in M&E, the cue phrase patterns are anchored by the Beginning-of-Sentence (BOS) and End-of-Sentence (EOS) markers[7], and we use the extended regular expression notation "*(BOS EOS)*$\{3,\}$" to indicate three or more intervening sen-

[5] The classifiers can be constructed so as to classify over any subset of relations $r \in R$, i.e. if we have $|R| = k$, we can construct one k-ary classifier, $\binom{k}{2}$ binary classifiers, etc.

[6] Distributed by the Linguistic Data Consortium, `http://www.ldc.upenn.edu`

[7] We insert these into the corpus using the LingPipe sentence boundary detection module, available from `http://alias-i.com`

RSR	Sample Pattern(s)	Extracted Instances	Training Instances (Set 1+2)	M&E Training Instances
Cause	*BOS* Because W_1 , W_2 *EOS* *BOS* W_1 *EOS BOS* Therefore, W_2 *EOS*.	1,867,619	926,654	889,946
Contrast	*BOS* W_1 , but W_2 *EOS* *BOS* W_1 *EOS BOS* However , W_2 *EOS*.	6,070,650	3,017,662	3,881,588
Adjacent	*BOS* W_1 *EOS BOS* W_2 *EOS*	3,901,922	1,938,720	-
NoRelSm	*BOS* W_1 *EOS (BOS EOS)*{*3,*} *BOS* W_2 *EOS*	3,799,450	1,887,740	1,000,000

Table 3.2: RSR types, sample extraction patterns, number of unique extracted instances from the Gigaword corpus, number of training instances in our active training sets 1 and 2, and number of training instances used by M&E for comparable relations.

tences in the NoRelSame pattern.

The W_1 and W_2 stand for the two text spans which we capture and across which the given relation is presumed to hold. Note that the captured text span does not include the cue phrase pattern itself, so that the model does not become skewed simply toward the cue phrases. Note also that Cause and Contrast patterns include those meant to match against both one-sentence and two-sentence instances.

Outputs: The number of instance examples I which we extract are shown in Table 3.2 (because of duplication inherent to newswire text, the counts in this table are for unique pairs after duplicates are removed). We randomly segment these instances as follows: We hold out 5,000 instances of each relation type to use for development purposes, and another 5,000 to use for testing. The remainder make up our training set. The development, test and training sets make up the count shown in Table 3.2 under the heading "Extracted Instances."

We further divide our training set into three subsets, Sets 1, 2 and 3, which are made up of approximately 25, 25 and 50 percent of our total training instances, respectively. The precise combined count for Sets 1 and 2 are shown under a separate heading in Table 3.2. Ultimately, in this chapter, we experiment only with Sets 1 and 2, in part because the size of these sets is closer to the size used by M&E and thus allows us to compare with their results more easily (although as we note later, there remain a number of differences between our experiments and theirs) and in part for pragmatic reasons which make it difficult to deal with large data sets.

Note that for Cause and Contrast, the number of pairs we extract is limited by the coverage of our patterns in the training corpus; for the Adjacent and NoRelSame patterns, we can extract more examples than we do, but currently impose a limit for efficiency reasons.

Model Building Inputs: The instance examples I in Table 3.2. the model parameters ϕ_m include Vocabulary Size, Stoplist Binning and Minimum Frequency cutoffs (we experiment with multiple values for these parameters as described below).

Outputs: Since we are interested in using the learned relation models in an online setting, e.g. as a resource for systems like DefScriber, we also pay significant attention to implementation of model storage and retrieval. We store the data in a MySQL relational database as it is optimized for high-volume, high-speed access and efficient use of system resources, and in addition is accessible through various software platforms. We store the frequency counts as four-column rows (*word1, word2, relation, frequency*). This format (rather than one row per word pair, with separate columns for each relation's frequency) allows us to use efficient, fixed length rows without allocating space for the large number of zero frequencies. Column types and other storage options have been made sufficiently efficient to store the entire TextRels model in memory for all experiments described in this chapter. Additional efficiencies are achieved by experiments described in the next section with stemming and vocabulary-size reduction.

Classification Inputs: The frequency counts F in database form as calculated in the previous step, as well as the classification parameters ϕ_c which specify the smoothing technique to use for unseen frequencies.

Outputs: For our experiments, we create a set of six binary classifiers (i.e. one classifier for each unique RSR pair, since we model four RSRs this amounts to $\binom{4}{2} = 6$ pairs) rather than a single n-ary classifier. This decision is motivated by the usage cases we present in the coming chapters, where we focus on situations where we either (a) are interested primarily in the NoRelSame versus not-NoRelSame distinction, but are not concerned with the identity of the non-NoRelSame relation (for assessing RSR cohesion in Chapter 4) (b) are interested in a NoRelSame versus Cause or NoRelSame versus Contrast relation only (for "relation-focused" questions in Chapter 5).

Overall, this baseline implementation of the TextRels system adopts the core pieces of M&E. To this point, we diverge from their work primarily in the exposition (breaking the modeling and classification into the above-mentioned three steps), selection of a slightly different relation set (adding Adjacent and omitting their Elaboration and Condition) and focus on an implementation with efficient online performance. In the next sections, we expand on M&E's work in depth and breadth. In Section 3.5.1 we expand depthwise, examining the impact of the model and classification parameters ϕ_m and ϕ_c. In Sections 3.5.2 and 3.5.3, we expand in breadth, examining how relation extraction can be improved by considering topic segments and syntax.

3.5.1 Parameter Optimization

In optimizing the model parameters ϕ_m and classification parameters ϕ_c, we have two goals. First, we wish to gain a better understanding of how these choices impact the quality of TextRels as reflected in its classification performance. Second, we wish to establish strong settings for these parameters so that any further experiments to broaden the model build upon a strong baseline.

Using classification accuracy on our automatically extracted development set as our guide, we initially experiment with a number of parameters, including tokenization rules (we find that using stemmed word forms[8] reduces model size without degrading performance; we also discard non-alphanumeric characters such as punctuation) and choice of smoothing methods (we find that Laplace smoothing is as efficient as other methods while being simpler to implement). We also experiment more rigorously with the following parameters:

[8] We use the Porter stemming algorithm (Porter, 1980) via its implementation from `www.tartarus.org/martin/PorterStemmer`

Vocabulary Size The vocabulary of tokens for which we compute frequency pairs individually in the Model Building stage of TextRels. The cutoff frequency is applied in terms of overall corpus frequency of word stems, i.e. when we use 3,200 as our vocabulary size, the most frequent 3,200 stems in the corpus form our vocabulary. All other tokens in I are replaced with UNK pseudo-tokens (or UNK_NUM for numerics). Then, when we compute the token-pair frequencies in F, we discard frequencies for which both tokens are a pseudo-token vocabulary, but maintain counts when only one member of the pair is a pseudo-token (effectively binning counts for pairs where one member is an UNK term). As we discuss below, we experiment with a cutoff from 400 to 12,800, finding our best performance in terms of mean classification accuracy across our three binary classifiers with a cutoff of 6,400.

Stoplist Binning In addition to cutting off the least frequent tokens using vocabulary size, we experiment with binning the most frequent n tokens, i.e. tokens which would typically be found on a stoplist such as *the*, *of*, etc. We do this by replacing them in the instance set I with a special COM pseudo-token. As with the out-of-vocabulary pseudo-tokens, we discard from F frequencies where both members are a pseudo-token. As we discuss below, we experiment with cutoffs between 0 and 50, and find a negative effect on accuracy even at low values; thus our best performance uses no stoplist binning (a bin size of 0).

Minimum Frequency After tallying frequencies from our training set (including any binning which happens as a result of the above parameters), we discard token pair counts with a total frequency less than n. We experiment with an n between 0 and 4 and find that the setting has no clear impact on accuracy. As we discuss below, we thus set minimum frequency truncation to 4 as this does not degrade accuracy and reduces the size of our database (depending on other parameters) by as much as half.

Laplace λ We use Laplace smoothing in our classification experiments to estimate the frequency of unseen token pairs. The lambda parameter specifies what frequency value to use for such pairs. As we discuss below, our experiments achieve peak performance with a Laplace λ of 0.25.

The motivation for experimenting with these particular parameters is both practical and theoretical. From a practical perspective, by truncating and binning frequencies we can reduce the size of our database of frequency counts to less than one-third size, depending on how aggressively we use these parameters. Theoretically, there is also grounding for discarding or binning low-frequency data points because their reliability can be doubtful [9]. The Laplace smoothing parameter, λ, is also critical, especially when we expect a large proportion of possible token pairs to be unseen, for instance when Vocabulary Size is large. Given a large number of unseen pairs, we can easily end up devoting an over-large portion of our probability space to such pairs if we set the λ value too high[10].

3.5.1.1 Stoplist Binning and Minimum Frequency Cutoffs

Figure 3.1 shows the mean classification accuracy across all binary classifiers over variations in Stoplist Binning size and Minimum Frequency cutoff. In Figure 3.2, we can see that the shape of the curves is similar, although not exactly the same, for the Adjacent versus Cause classifier. (The other five individual binary classifiers show similarly minor variations from the mean; in general, we will show “all vs all” mean plots for our experiments, as well as individual pairwise classifier plots where they differ from the mean in an interesting or substantial way.)

We observe first that stoplist binning, particularly using the longer stoplist size, has a strong negative effect on accuracy. The interpretation of this result is that even very common words are distributed differently across the different RSRs.

Furthermore, we observe that increasing Minimum Frequency cutoff size from zero to four has only minor effects on accuracy. We do not carry out thorough experiments at a higher cutoff but note that several smaller tests showed a degradation of results at a cutoff of eight; however a cutoff of four appears to provide similarly accurate estimates to the lower cutoffs while reducing the size of our database by approximately one-half (depending on other parameters).

[9] Dunning (Dunning, 1994) and Manning and Schütze (Manning and Schütze, 1999, Chapter 6) explore this issue in detail.

[10] See Manning and Schütze (Manning and Schütze, 1999, pp.202-3) on this issue.

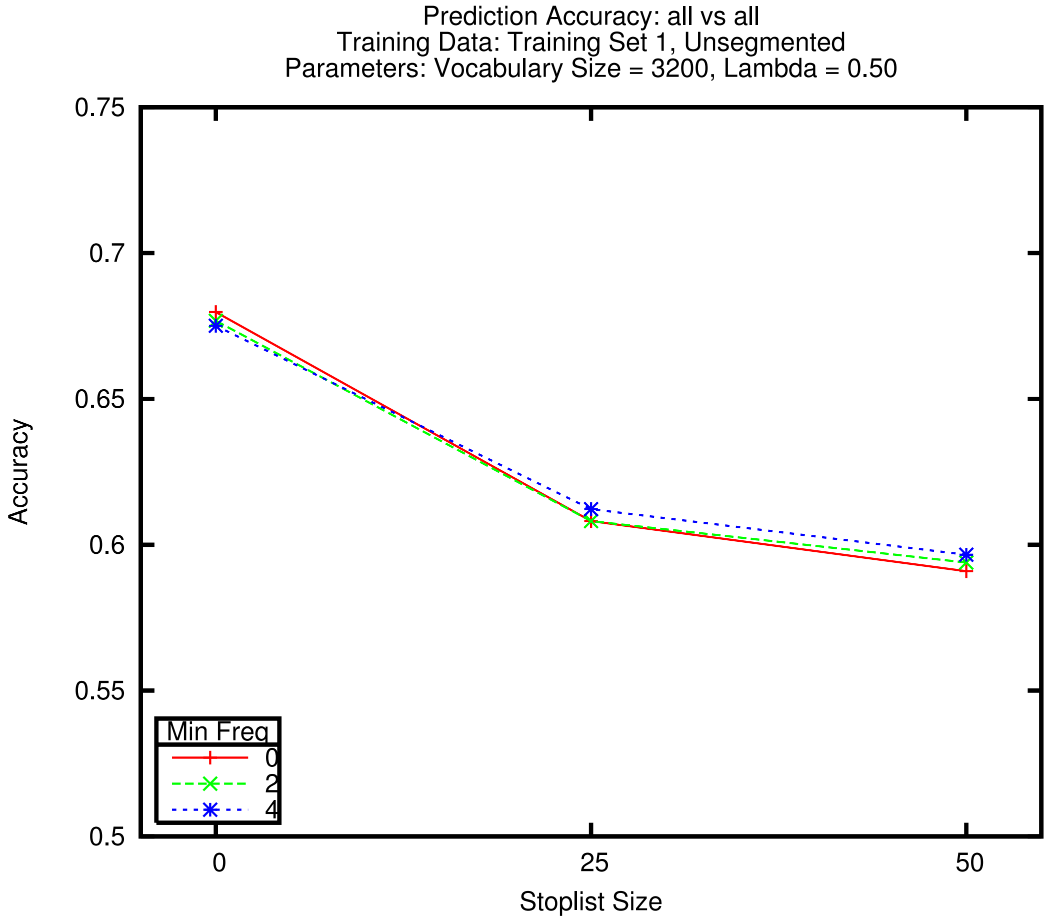


Figure 3.1: Effects of Stoplist Binning stoplist size and Minimum Frequency cutoff on mean classification accuracy across all binary classifiers.

Using these results, we establish baseline parameters for Minimum Frequency cutoff of four and Stoplist Binning stoplist size of zero in our remaining experiments.

3.5.1.2 Vocabulary Size and Laplace Lambda

Figure 3.3 shows the effect of varying the Vocabulary Size and Laplace λ parameters. We find that Vocabulary Size and Laplace λ both have a strong and related effect on performance.

We see that at lower values of Vocabulary Size, the effect of not keeping separate counts for rarer tokens (and corresponding binning of these frequencies with the UNK or UNK_NUM pseudo-tokens) has a negative effect because information about more specific

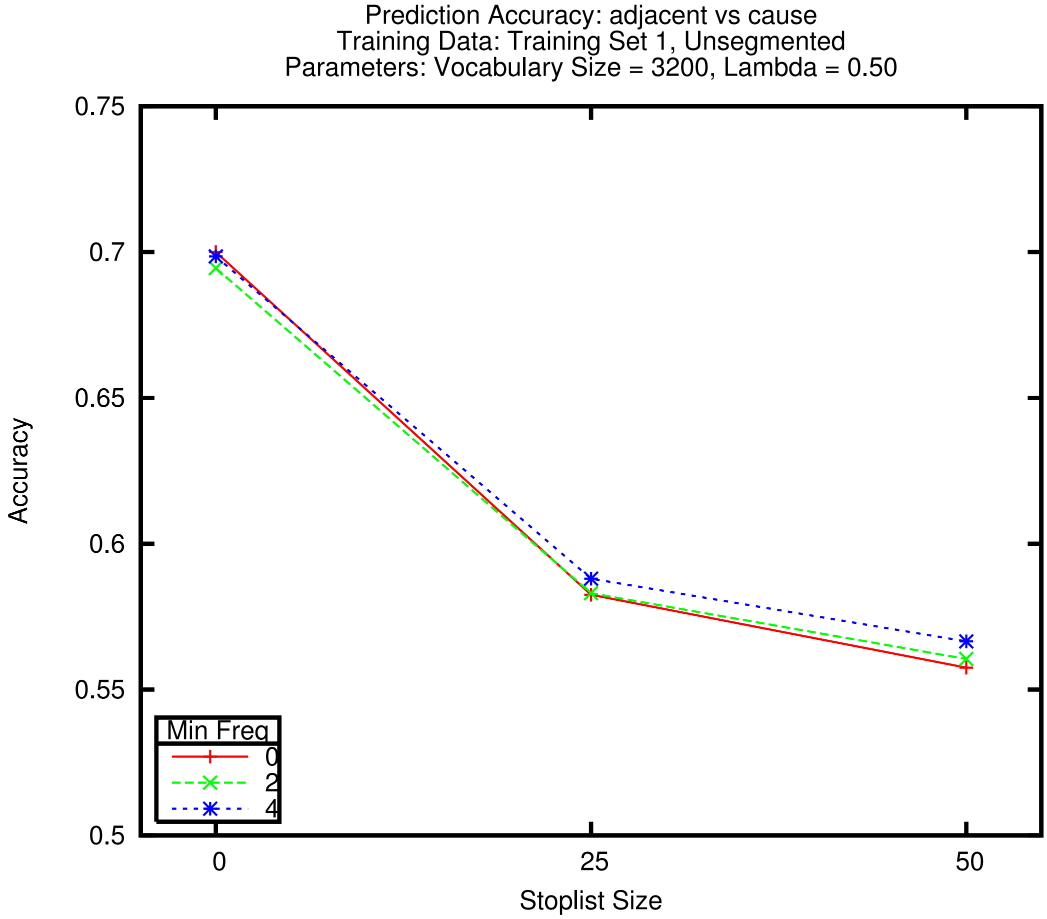


Figure 3.2: Effects of Stoplist Binning stoplist size and Minimum Frequency cutoff on accuracy of Adjacent versus Cause binary classifier.

terms is lost. However, we see that at the highest value, 12,800, we appear to reach a point of diminishing returns, where the frequency counts are too sparse and result in unreliable data.

We also see that as the Vocabulary Size increases, the effect of the λ parameter is more pronounced. This is to be expected since the λ parameter is used when a given pair is unseen; as we increase our vocabulary size, the chance of a pair being unseen, rather than collapsed into a seen UNK frequency, also increases. We observe that too-high or too-low values decrease performance at the higher Vocabulary Size values, but that choosing an above-optimal value at higher vocabulary sizes is an especially poor choice.

We also hypothesize that some of the fall-off in performance as Vocabulary Size grows

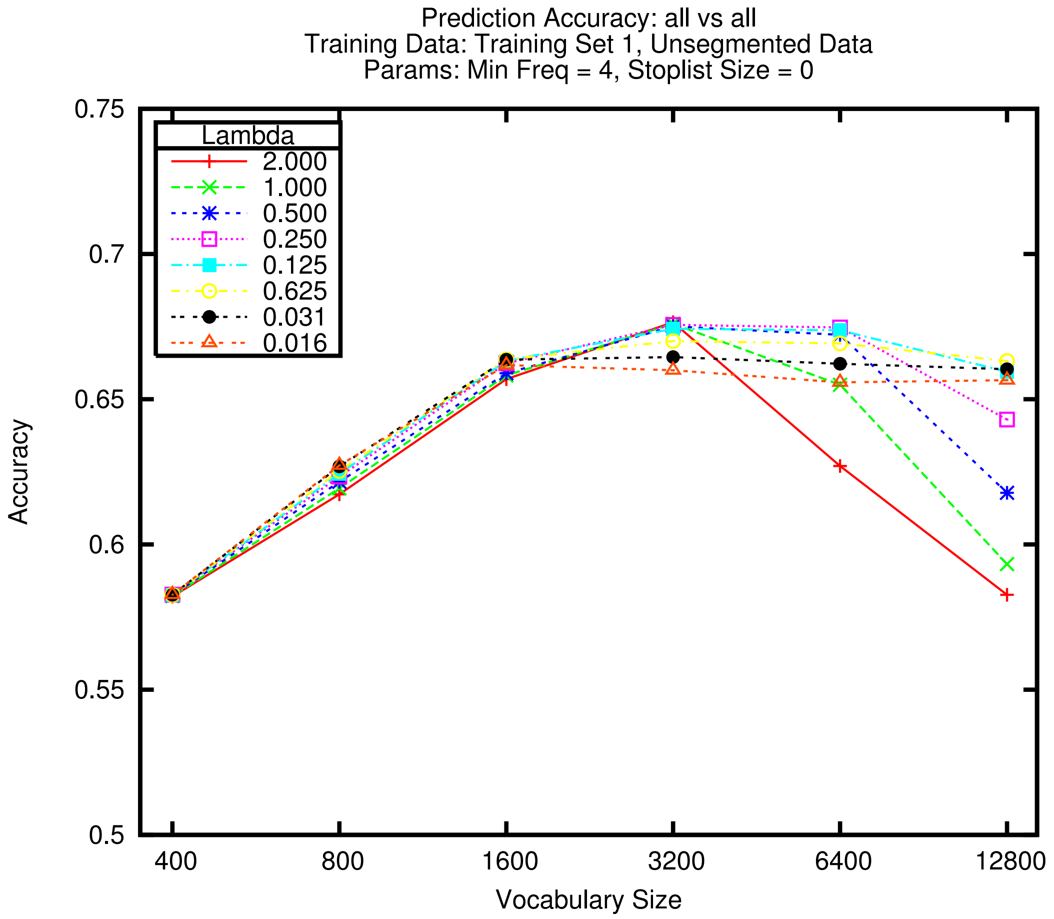


Figure 3.3: Effects of Vocabulary Size and Laplace λ on mean accuracy across all binary classifiers.

from 3,200 to 12,800 can be addressed with more data. Until this point, we have presented results from these parameter optimizing experiments using only one quarter of our training set (a.k.a. "Training Set 1") in order to allow us to experiment more quickly. We now combine the second quarter of our full data set to see the effect of the presumably less-sparse frequency counts in the combined set (a.k.a. "Training set 1+2").

Figure 3.4 shows that, indeed, when we add the second half of the training data, performance does improve at the higher values, with the best performance in terms of mean accuracy at the 6,400 level. Looking more closely at the individual classifiers in Figure 3.5, additional training data is more helpful to some classifiers than to others; the Cause-Contrast classifier benefits the most from the extra data; the Adjacent-Contrast the least.

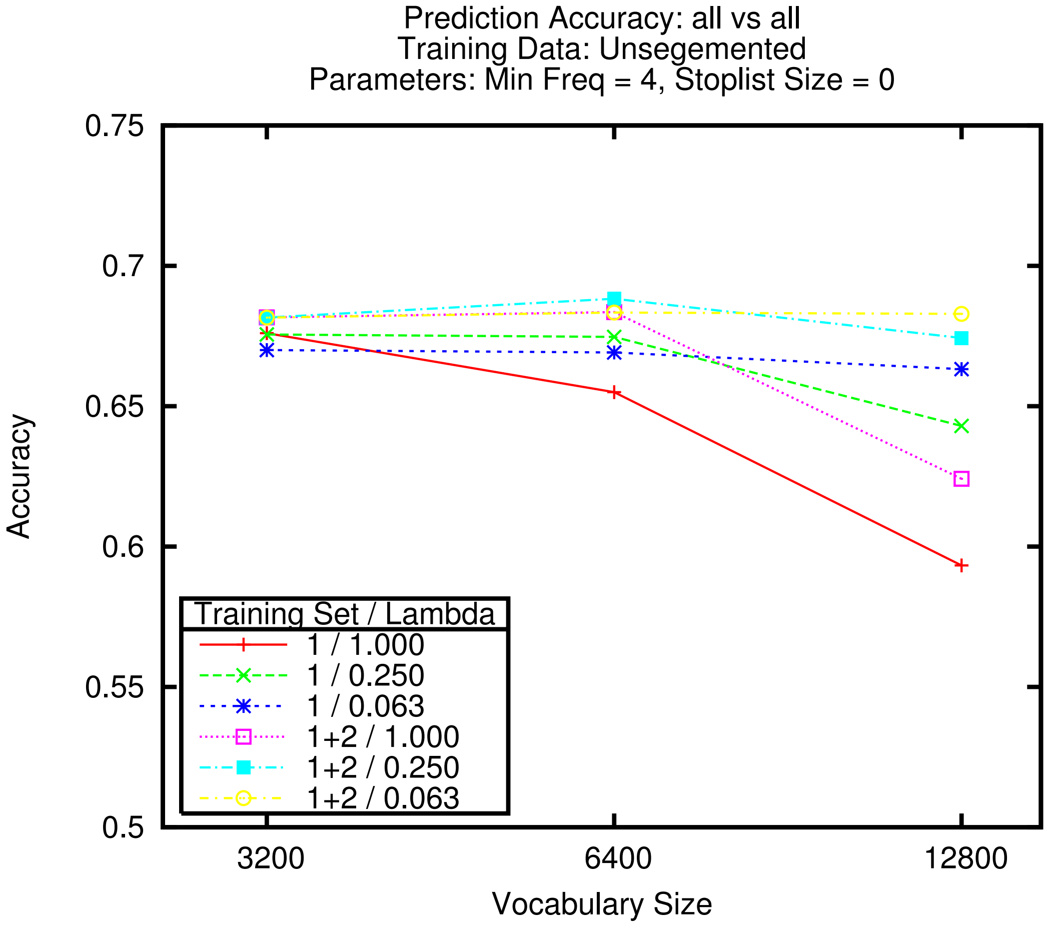


Figure 3.4: Incorporation of training set size variation alongside Laplace λ and Vocabulary Size on classification accuracy using TextRels model data.

Because the increases are for the most part modest, and for reasons of CPU time and storage capacity as well, we do not ultimately experiment with the remaining part of our training data ("Training Set 3").

3.5.1.3 Overall Optimization

Overall, we achieve peak performance on our development set, in terms of mean classification accuracy, at vocabulary size 6400, using no Stoplist Binning, a Minimum Frequency Cutoff of four, and a Laplace $\lambda = 0.25$.

We test the performance of these parameters on two test sets. The first test set we use ("Auto") is composed of 5,000 held-out instances of each relation culled from our automatic

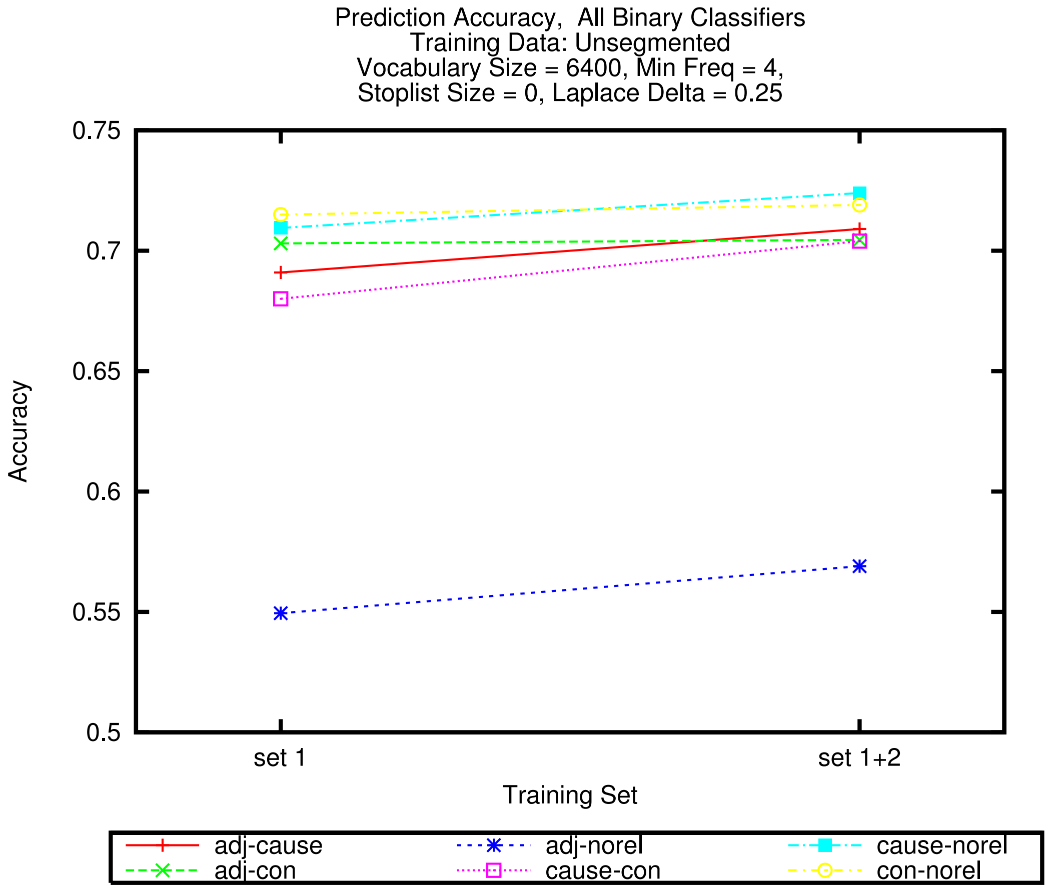


Figure 3.5: Effect of increased training set size on classification accuracy for each binary classifier; adding training set 2 approximately doubles the number of training instances I which are input to the Model Building phase of TextRels.

RSR instance extraction process as described in Section 3.4. As with our training data, all cue phrases are removed from these instance examples.

The second data set ("PDTB"), is derived from the Penn Discourse TreeBank (Webber et al., 2005; Prasad et al., 2006). The purpose of testing on this data is to address two issues which arise from our training and testing with automatically extracted, cue phrase-based, relation instances. First, we can use PDTB instances to be certain that in the eyes of a human annotator, the instances which our classifier classifies are actually related by the given relation. With the Auto data, there is the possibility that an extracted instance, while matching a cue phrase extraction pattern, may not be a "true" relation instance (for instance, resulting from a pattern match on a non-Contrast use of *but* in its secondary meaning as a synonym of *except*). Second, even in the case when an example is "true," it is not necessarily the case that performance on the Auto set will result in good performance on detecting actual implicit RSRs in text. That is, our Auto examples are in some sense synthesized implicit relation examples; while cue phrases themselves have been removed from our Auto instances, it is possible that our classifiers are relying on some other terms which associate with the cue phrases themselves but would not be present in a real implicit case.

To address these issues, we use data from the PDTB. We extract "Implicit" relations, i.e. text spans from adjacent sentences between which annotators have inferred semantics not marked by any surface lexical item. Recall that this is essentially the situation in Example 2 which our models are built to detect. To extract test instances for our Cause RSR, we take all PDTB Implicit relations marked with "Cause" or "Consequence" semantics (344 total instances); for our Contrast RSR, we take instances marked with "Contrast" semantics (293 total instances). PDTB marks the two "Arguments" of these relationship instances, i.e. the text spans to which they apply; these are used as test (W_1, W_2) span pairs for classification. We test the performance on PDTB data using 280 randomly selected instances each from the PDTB Cause and Contrast sets, as well as 280 randomly selected instances from our test set of automatically extracted NoRelSame instances (while there is a NoRel relation included in PDTB, it is too sparse to use in this testing, with 53 total examples; there is no clear PDTB equivalent to the Adjacent relation).

Table 3.6 lists the accuracy for the optimized ("Opt") classifier over the PDTB and Auto test sets[11].

We also list for reference the accuracy reported by M&E; however, their training and test sets are not the same as ours so this comparison can only be seen as approximate. In addition, their test set is extracted automatically, i.e. in the exact same manner as our

[11] We do not provide pre-optimization baseline accuracy because this would arbitrarily depend on how sub-optimally we select parameter values. For instance, by using a Vocabulary Size of 3,200 (rather than 6,400) and a Laplace λ value of 1, the mean accuracy of the classifiers on the Auto test set drops from 71.6 to 70.5; using a Stoplist size of 25 (rather than 0) drops this number to 67.3.

	PDTB-Opt	Auto-Opt	M&E
Cause vs Adjacent	-	71.4	-
Contrast vs Adjacent	-	69.6	-
NoRelSame vs Adjacent	-	56.8	-
Cause vs Contrast	59.1	69.8	87
Cause vs NoRelSame	75.2	72.7	75
Contrast vs NoRelSame	67.4	70.7	64

Table 3.3: Accuracy of parameter-optimized ("Opt") classifiers across PDTB and Auto test sets described in Section 3.5.1. Accuracy results from M&E use different test data, and are reported for approximate reference only. Baseline in all cases is 50%.

Auto set.

One immediate observation is that, in the Cause versus Contrast case, M&E's reported performance exceeds ours significantly. However, they also report a subset of experiments which evaluate classification on the human annotated RSTBank corpus (Carlson et al., 2001) over instances where no cue phrase is present, similar to our PDTB experiments. In those experiments, for the Cause versus Contrast case, they report only 63% accuracy over a 56% baseline (the baseline is > 50% because the number of input examples is unbalanced). However, they do not provide accuracy results for the other classifiers which we list in Table 3.3. We note that in our results, we also observe a large, although not nearly as precipitous, drop in performance between our Auto and PDTB results for Cause versus Contrast.

While we cannot conclusively explain this difference, we posit that accuracy on human-annotated cases where no cue-phrase is used is a more reliable measure. As noted above, several potential issues in regard to automatically extracted test cases are taken out of play. First, we are assured that the relation of interest is actually present; furthermore we remove the possibility that the procedure for removing the cue phrase patterns from the training/test examples has left some biasing information in the model such as cue phrase-associated terms (or even punctuation; M&E include punctuation in their frequency counts, although we do not).

We observe that in other cases our performance is competitive with M&E on the Auto test set. In the Adjacent cases, which have no comparison point with M&E or PDTB, we are surprised to find that differentiating Adjacent from Cause and Contrast appears to be slightly easier than differentiating from NoRelSame. This is especially surprising given that some number of Adjacent examples are expected to be implicit, or even explicit, Cause or Contrasts themselves[12] On the other hand, performance of the Adjacent versus NoRelSame classifier is much lower. We hypothesize that our idea of modeling a catch-all relation with Adjacent may simply be too broadly defined for effective classification in this context.

We do not compare the "Opt" results with a single pre-optimization baseline, because this number would be arbitrarily low depending on how sub-optimally we set the parameters in ϕ_m and ϕ_c. But the previous subsections examining the impact of these parameters show convincingly that the optimization of these parameters is critical for achieving strong results. Rather than attempting possible further refinements and experiments in optimizing this model (i.e. by further parameter tuning or increased volume of training data), we turn our attention to techniques for improving performance given this set of optimized parameters and training data.

3.5.2 Topic Segmentation Filtering

We use automatic topic segmentation to investigate the impact of a key assumption in our two-sentence RSR patterns, namely that these patterns consider all sentence boundaries as equivalent. As readers, we realize that some sentence boundaries indicate merely a pause in a continuous thought, while others (such as those at the end of a paragraph) can separate quite distinct topics; we consider how topic segmentation can allow us to take this into account when extracting RSR instances.

For instance, in the case of NoRelSame, our assumption in the previous section is that having at least three intervening sentences is an appropriate heuristic for finding spans which are not joined by one of the other RSRs. We posit that the additional requirement

[12] Given that Adjacent instances are extracted at random, we would in general expect about three percent to be explicit Cause or Contrast instances, based on the fact that only about 3.1 million two-sentence Cause or Contrast instances are found in the approximately 100 million-sentence Gigaword corpus.

of having sentences belong to distinct topical segments could increase the accuracy of this assumption. Conversely, with the Adjacent relation, the assumption is that the mere adjacency of two sentences puts them in a well-connected space, and we wish to analyze whether the additional constraint that the sentences be in the same topical segment can result in improved instance selection.

While the Cause and Contrast relation instances are not exclusively found in two-sentence patterns, a small number of Cause instances and a significant number of Contrast instances are found this way. Specifically, if we look at the pattern-associated frequencies in Appendix B, we see that 5.3 percent of Cause and 50.0 percent of Contrast instances are extracted with two-sentence patterns. In these cases, we can consider applying the same heuristic constraint to that discussed above for Adjacent, namely adding the constraint that in a two-sentence pattern match, in addition to matching the surface-level cue-phrase pattern, both sentences must fall within the same topic segment.

In this way, we can consider this proposal for filtering in terms of our formal representation of input to TextRels as an added symbol in both the corpus T and the patterns P. In our corpus, we insert a special SEG symbol for topic segment marking, just as we did with the BOS-EOS sentence boundaries. Then, we can specify the absence of this between sentence boundaries in our two-sentence Cause, Contrast and Adjacent patterns, and the presence of (at least one) SEG between NoRelSame instances.

3.5.2.1 Segmenting the Corpus

In order to insert these topic segment markers in the corpus, we must have an appropriate tool. While we originally consider using the manual paragraph markers which are present in much newswire text as our topic boundaries, we reject this approach for two reasons: (1) It restricts us to training on data with paragraph markers, which we may not always have and/or may be used inconsistently (2) In the Gigaword corpus, we find that on average a paragraph contains 1.54 sentences, meaning that many of the “segments” derived in this way would be one sentence long. Using these segments would essentially mean drastically reducing the available number of Adjacent examples, result in no change in NoRelSame, and elimination of most two-sentence Cause and Contrast instances.

Total Documents	1,052,310	Total Paragraphs	12,629,889
Total Sentences	19,544,394	Total SEGs	5,558,946
Mean Sentences / Paragraph	1.54	Mean Sentences / SEG	3.51
Precision	0.76	Recall	0.33
Baseline Precision	0.68	Baseline Recall	1

Table 3.4: Overview of performance for automatic insertion of SEG topic boundaries into Gigaword corpus. An insertion is considered correct where it occurs at an existing paragraph boundary. (Figures are based on performance over a random sample of approximately one-fifth of the corpus.)

For these reasons, we use an automated topic segmentation tool, LCSeg [13] (Galley et al., 2003) for this purpose. LCSeg is a parameterizable segmenter which uses lexical coherence and a variant of the TextTiling algorithm (Hearst, 1997) to build topic segments. While the concept of a topic "segment" can be defined at various levels of granularity, we adopt a goal-based view and aim to mark segments of approximately four sentences in length, reasoning that these segments will be long enough to exclude some candidate NoRelSame examples, yet short enough to exclude a non-trivial number of Adjacent, Contrast and Cause examples too. Based on the intuition that "true" segment boundaries should occur at existing paragraph boundaries, we evaluate the automatically-inserted segments with respect to the paragraph markers in our training corpus to compute precision and recall of segment boundaries in terms of alignment with paragraph boundaries. Note that we are in fact aiming for low recall, since we are actively trying to make the segments longer than typical paragraphs. However, we would like to see high precision to indicate that the segmenter is not putting segments in the middle of paragraphs which do exist, since this would indicate that its performance were suspect. Table 3.4 shows an analysis of our segmentation performance on a random sample of approximately one-fifth of our corpus; we achieve 76 percent precision, as compared with 68 percent precision for a random baseline (inserting a segment boundary between all sentences).

[13] Available from `http://www1.cs.columbia.edu/ galley/tools.html`

RSR Type	Segmented Instance Count	Unsegmented Count
Cause	1,842,246	1,867,619
Contrast	6,070,650	5,318,336
Adjacent	3,901,922	3,901,922
NoRelSame	3,799,450	3,799,450

Table 3.5: Instance counts I extracted in the Instance Mining phase using the SEG-augmented patterns over the automatically segmented Gigaword corpus. Counts extracted using the unsegmented (baseline) patterns/corpus are provided for comparison.

3.5.2.2 Training with Segmented Data

We can now retrain TextRels to build a "Seg" model. In terms of the three RSR modeling steps defined earlier, this involves fairly few changes.

For *instance mining*, we use the Gigaword corpus with the LCSeg-inserted SEG boundary markers, and extract RSR instances with our SEG-augmented patterns. The number of instance examples we find are listed in Table 3.5. We can attribute the reduced number of instances in the corpus found to the selectivity of the SEG-augmented two-sentence patterns for the Cause and Contrast relations; we observe that the effect on the number of Contrast instances found is significantly greater. (Appendix B, shows the precise number of instances found for each pattern.) As in the baseline implementation, we again segment instance data into test, development and three training sets. However, by decreasing the size of the third training set (which is ultimately unused), we keep the size of Training Set 1 and 2 constant over the unsegmented and unsegmented instance sets. That is, the Seg model is trained on the same number of input instances for all RSRs as the optimized classifier discussed in the previous section.

For *model building* and *classification*, we use the optimized parameters discovered in building the "Opt" model in the previous section. Thus, the only difference between the Seg and Opt classifiers is that the former is trained on SEG-constrained RSR instances.

3.5.2.3 Evaluation

We follow a similar evaluation protocol as in our optimization experiments to evaluate the impact of training on segmented data, i.e. we use a held-out test set of 5,000 instance examples of each type to test each of the binary classifiers. However, an important question to consider is whether we should use test data from our original unsegmented instance set, or instead take a new set of test data from our segmented instances.

We decide to test on both segmented (“Auto-S”) and unsegmented (“Auto”) test sets because both choices have relevance for us. On one hand, since the segmentation is done automatically, classifying on segmented examples is a realistic test scenario. That is, given a “real world” document, we can always use the same segmentation procedure to help our classification judgments.

On the other hand, testing on unsegmented input allows us to compare more directly to the numbers from our previous section and address concerns that any differences might result from testing on segmented examples. In addition, for tasks which apply RSR models outside of a single-document context, such as those we consider in the next chapters, a test on unsegmented input may be more relevant inasmuch as in those cases, e.g. where we are comparing sentences from separate documents, the idea of distinguishing whether the text spans under consideration are from the same topic segment does not make sense.

Table 3.6 shows the results on both the Auto and Auto-S test sets, as well as the PDTB test set, for the Seg classifiers, with the results of the Opt classifier presented in the previous section for comparison.

We observe that the performance of the classifiers is indeed impacted by training on the segment-constrained instances. On the PDTB test data, performance using the segment-trained classifiers improves in two of three cases, with a mean improvement of 1.2%. However, because of the small size of this set, this margin is not statistically significant.

On the Auto and Auto-S test data, our first observation is that, overall, it appears that the topic segmentation constraints on the definition of Adjacent produce a model of this relation which is harder to differentiate from the others than was the case without segmentation. The decline is somewhat intuitive with respect to Cause/Contrast versus Adjacent classifiers in the following sense: given the constraint that Adjacent spans must

Classifier / Test Set	Pdtb Opt	Seg	Auto Opt	Seg	Auto-S Opt	Seg
Cause vs Adjacent	-	-	71.4	70.6	73.5	73.0
Contrast vs Adjacent	-	-	69.6	68.7	72.0	71.3
NoRelSame vs Adjacent	-	-	56.8	53.4	58.7	58.9
Cause vs Contrast	59.1	61.1	69.8	69.7	70.4	70.6
Cause vs NoRelSame	75.2	74.3	72.7	73.5	71.2	72.3
Contrast vs NoRelSame	67.4	69.7	70.7	71.3	68.2	70.1
Mean	67.2	**68.4**	**68.5**	67.9	69.0	**69.4**

Table 3.6: Classifier accuracy across PDTB, Auto and Auto-S test sets for the optimized classifiers ("Opt") described in Section 3.5.1 and the retrained version using segment-constrained instances ("Seg") described in Section 3.5.2.

be pulled from the same topic segment, we might imagine that the training span pairs might be somehow semantically "tighter" and thus more difficult to differentiate from the Cause and Contrast relations which, after all, Adjacent is meant to include (along with any other relation). However, by the same token, one would expect Adjacent and NoRelSame to become easier to differentiate, yet they do not. For the Adjacent vs NoRelSame classification, the Seg classifier does considerably worse than the Opt one on Auto test data, and only marginally better on the Auto-S test data.

For this reason, we separately analyze the impact of the segmentation constraints on the "X vs Adjacent" classifiers, i.e. the first three rows of Table 3.6, and the other classifiers. For the X vs Adjacent classifiers, results decline when using the "Seg" classifier in five of six cases (and are virtually even in the sixth case). This pattern carries across both the Auto and Auto-S test sets. On the Auto test set, which declines more precipitously, the decline in performance when using the Seg rather than Opt classifier is significant using a Chi-squared test ($P < .05$) both for the individual NoRelSame vs Adjacent classifier, as well as the "X vs Adjacent" classifiers when considered as a group. (The differences on the corresponding Auto-S classifiers do not reach statistical significance.)

However, the second three rows in Table 3.6 are more encouraging. In these three cases, the Seg classifier is the best performer in five of six cases (and virtually even in the sixth), across both the Auto and Auto-S test sets. On the Auto-S test data, the Seg classifier is the best performer in all three cases; while the margin is not statistically significant for a single classifier, the overall accurate-inaccurate improvement is significant ($P < .05$) using a Chi-squared test. On the unsegmented test data, the improvement does not rise to statistical significance. Nonetheless, we conclude that for the subset of cases involving only Cause, Contrast and NoRelSame, a Seg-based classifier improves RSR classification accuracy.

3.5.3 Syntax-Based Filtering

A second avenue we pursue concerns improving our ability to find the correct extent of a mined relation instance. In both the baseline and segment-augmented models, the patterns which we use rely on sentence boundaries or cue phrase text as the limit of a text span. However, we often observe that an instance which matches such a pattern can capture a text span of which only one part is relevant to the relation in question. For instance:

Example 3 *Wall Street investors, citing* **a drop in oil prices because of weakness in the automotive sector***, sold off shares in Detroit's big three at a record pace today.*

In this case, a syntactically informed analysis could be used to extract the constituents in the cause-effect relationship from within the boldfaced nominal clause only, i.e. as "a drop in oil prices" and "weakness in the automotive sector." Our current flat patterns, however, will simply split around the cue phrase "because of" and extract the entire first and second parts of the sentence as the constituents.

Of course, it is possible that this is for the best, i.e. that this is not really over-capturing behavior and results in "useful" noise. That is, having pairs like (*citing, sold*) or even (*investors, shares*) may help more than it hurts – to some extent that is what the experiment being described is testing. Is it better for our model to train on the more tightly construed semantics which can be attached to these cue phrases, or preferable to pull in spans of text which may not be true constituents but are likely closely related as part of the same sentence?

Recognizing the potential complexity of using syntactic phenomena to improve our model quality, we reduce the dimensions of the problem somewhat before embarking on

Category	**Example Instance**
Full Relevance	*Cause (25 instances)*: [[We have suffered so much]] because [[appliance manufacturing is very competitive]] . *Contrast (42 instances)*: [[Georgia fans weren't happy]] , but [[they couldn't be too angry]] .
Irrelevant modifier (manner, temporal, etc.)	*Cause (17 instances)*: [[Fields has drawn media attention here]] , because [[at 38 he is the youngest person to head a major automaker]] in Japan , a culture long dominated by the rule of seniority . *Contrast (24 instances)*: [[Between Strangers " is not in competition for any award]], but [[nonetheless had much of the attention]] Friday on the second day of the festival .
RSR occurs entirely within NP or other phrasal adjunct.	*Cause (10 instances)*: In addition to the wildlife park and the transvestite show , the itinerary included a series of state-run shops in Bangkok that eagerly accepted [[Chinese currency , the yuan , which was shunned until recently]] because [[banks will still not freely convert it into Thai baht or American dollars]] . *Contrast (2 instances)*: A civilized , healthy society comprised not just of [[human animals]] , but [[conscious , moral human beings]] , needs for David Cash to comprehend the enormity of his moral breach and to stagger under the burden of that understanding.
RSR in embedded finite clause (often with "say").	*Cause (45 instances)*: Mahmood Ali , one of the injured Muslims , said [[he survived]] because [[an elderly Sikh woman asked the attackers to spare him]] *Contrast (37 instances)*: A British emissary , Derek Fatchett , said here this week that [[European countries were ready to help ease Jordan ' s debt problems]] , but [[with debt rescheduling , not debt forgiveness]] .
RSR limited to one constituent of a conjunction.	*Cause (3 instances)*: But you know , [[we respect what he's doing]] because [[he's looking out for us]] and hopefully we'll continue to go out there and take care of him so he'll take care of us. *Contrast (3 instances)* He suggested that the Juan de Fuca plate is curved and twisted as it dives beneath the continental margin , and that [[last week ' s quake may have transferred fresh stresses to the deep ground south of the epicenter beneath Olympia while decreasing some of the stress northward beneath Seattle – " good news for Seattle ,]] but [[bad news for Olympia]] .

Table 3.7: Overview of syntactically categorized errors over manually labeled set of 100 examples each for Cause and Contrast instances. RSR-relevant extents are marked in brackets. (Instance counts add to more than 100 since categories are not mutually exclusive.)

our experiments.

Single-Sentence, Cause and Contrast We focus on improving instance mining of single-sentence instances only. This means we analyze only Cause and Contrast patterns, since NoRelSame and Adjacent use only multi-sentence patterns.

Limited Pattern Set Within the Cause and Contrast patterns, we further narrow our investigation to the most productive single-sentence pattern of each type, in terms of instances found. We do this based on the intuition that different syntactic phenomena may be in play for different patterns, and we wish to limit the amount of analysis and maximize the "multiplier effect" of any improvements we do make.

To this end, we analyze the patterns "W_1 because W_2" for Cause and "W_1 , but W_2" for Contrast. As shown in the listing of cue phrase patterns in Appendix B, these patterns account for a significant portion of our training set, or approximately 54.3 percent of total Cause and 38.0 percent of Contrast examples respectively. Moreover, in both cases, these patterns yield at least ten times more instances than the next-most-productive single-sentence patterns.

Reduced Instance Set We experiment on a limited-size training set because of the CPU-intensive nature of parsing a large set of training examples. However, our implementation of syntactic filtering is such that we need only parse known Cause/Contrast instances, rather than the entire corpus, so our parsing effort is well targeted. We are able to parse 400,000 examples each of Cause and Contrast for our final results. Compared with our baseline model, this is approximately 43 percent of our total Cause instances and 13 percent of our total Contrast instances. For the NoRelSame model, we use a randomly selected subset of 200,000 instances from our baseline instance set.

(Another option for a parsed training corpus would be to use a pre-parsed corpus, namely the BLIPP corpus or Penn Treebank. However, while these are large corpora, only a small percentage of their content contains RSR example instances, and would yield perhaps one-tenth as many parsed instances as we are able to by targeting our parsing effort[14], while limiting our flexibility in terms of parser choice etc.)

[14] This estimate is based on Marcu and Echihabi's reporting of their work with the BLIPP corpus (Marcu

3.5.3.1 Analyzing and Classifying Syntactic Errors

We begin by analyzing possible syntactic bases for the type of over-capturing behavior shown in Example 3. Our approach was as follows:

1. Create small development set from holdout training data, with 100 randomly selected instance examples each of the Cause and Contrast instance patterns mentioned above.
2. Manually identify and categorize any instances found to have over-capturing behavior by labeling the relation-relevant and irrelevant spans.
3. Implement a set of syntactic filters to attempt to reduce the capturing of irrelevant text.
4. Test the filters on our manually labeled spans; attempt to achieve high precision rules. Iterate over training and testing data.

We perform the first two steps of the above procedure for both the Cause and Contrast models and find that there are several common reasons for over-capturing which affect a significant percentage of examples. Examples of each are shown in Table 3.7.[15]

For Step 3, we use the Collins Parser (Collins, 1996)[16] We then design syntactic filtering heuristics manually based on an examination of parse trees of several examples from our development set.

3.5.3.2 Formulating Syntactic Filtering Heuristics

For Contrast, we find that using the coordinating conjunction (CC) analysis of *but*, we can use a straightforward rule which limits the extent of RSR spans captured to the two phrasal

and Echihabi, 2002, pp.5-6) and the Penn Discourse TreeBank manual (Prasad et al., 2006).

[15] An interesting note here is that in the 200 examples which we analyze, none capture a non-discourse keyword use, i.e. a situation where our instance mining pattern captures text where there is no RSR relation whatsoever.

[16] We initially considered several options for syntactic analysis, including MiniPar (Lin, 1998b), but choose the Collins' parser since working with its TreeBank-style trees will allow us to use and/or compare our data to Penn Discourse TreeBank(Prasad et al., 2006) data more easily going forward.

```
S:
  PP: For (IN)
    NP-A:
      NPB: the (DT) past (JJ) six (CD) months (NNS)
      ADVP: or (CC) so (RB) , (PUNC,)
  NP-A:
    NPB: management (NN)
  VP: has (VBZ)
    VP-A: been (VBN)
      VP-A: but (CC)
[[      VP: revamping (VBG)
          NP-A:
            NPB: positioning (NN) and (CC) strategy (NN)
]]
[[      VP: scaling (VBG)
          ADVP: also (RB)
          PRT: back (RP)
          NP-A:
            NPB: operations (NNS) . (PUNC.)
]]
```

Figure 3.6: Parse tree for the sentence, “For the past six months or so, management has been revamping positioning and strategy but also scaling back operations.” Brackets indicate the spans which are considered to be the constituents of the RSR according to the Contrast heuristic described in Section 3.5.3.2.

children of the CC node. In Figure 3.6, we see how this heuristic is applied to the sentence "For the past six months, management has been revamping positioning and strategy, but also scaling back operations."

The brackets in the example indicate the extents which are kept by the heuristic; we observe that it successfully cuts out the irrelevant temporal relative clause ("for the past six months ...") and pulls out the relevant VP phrases which are in Contrast. Note that in this instance the adverb *also* would ideally be filtered out as well, but it is more difficult to make a general heuristic to do this, since one can imagine contentful adverbials as well, e.g. *strategically*, which one might not want to filter out.

The analysis of the correct extents for the *because* pattern is slightly more complex. We experiment with several heuristics, and ultimately consider two. In both heuristics, we capture the right-hand span as any text in child(ren) nodes of the *because* IN node. However, the heuristics differ in how much text they select as their left-hand span. In the *Aggressive* heuristic, we follow parent nodes of *because* only until we reach the nearest phrasal (e.g. VP) or finite clause (e.g. SBAR) node, taking any text at or below that level of the syntax tree down to the *because* node. In the *Inclusive* heuristic, rather than stopping at the first phrasal or finite clause node encountered, we capture the left-hand span of the Cause RSR by ascending the tree up to the highest enclosing phrasal/finite clause ancestor, and capturing the text in that node and any of the nodes passed along the way.

In Figure 3.7, we see a case where the Aggressive heuristic (indicated by the opening brackets with a minus (-) sign) is successful in excising the irrelevant section of the tree, i.e. the portion of the tree attached above the SBAR-A *said* node. In this case the Inclusive heuristic (indicated by the opening brackets with a plus (+) sign) will capture the word *said* in addition, although in our analysis, this is not part of the relation span[17].

However, if we consider the sentence in Figure 3.8, we see that in this case the Aggressive heuristic cuts out an important part of the parse tree, namely the noun *drop* which clearly has a role in the Cause relation here. The Inclusive heuristic, on the other hand reaches all

[17] In this kind of sentence, i.e. where a matrix clause is embedded under a verb like *say*, the PDTB annotators also exclude the governing verb, lending credence to our analysis here. We discuss our heuristics in terms of PDTB agreement in more detail in Section 3.5.3.3.

```
S:
  NP-A:
    NPB: Mahmood (NNP) Ali (NNP) , (PUNC,)
    NP:
      NPB: one (CD)
      PP: of (IN)
        NP-A:
          NPB: the (DT) injured (JJ) Muslims (NNP) , (PUNC,)
+[VP: said (VBD)
    SBAR-A:
-[    S-A:
      NP-A:
        NPB: he (PRP)
      VP: survived (VBD)    ]]
        SBAR: because (IN)
[[      S-A:
          NP-A:
            NPB: an (DT) elderly (JJ) Sikh (NNP) woman (NN)
          VP: asked (VBD)
            NP-A:
              NPB: the (DT) attackers (NNS)
            SG-A:
              VP: to (TO)
                VP-A: spare (VB)
                  NP-A:
                    NPB: him (PRP) . (PUNC.)   ]]
```

Figure 3.7: Parse tree for the sentence, "Mahmood Ali, one of the injured Muslims, said he survived because an elderly Sikh woman asked the attackers to spare him." Brackets indicate the spans which are considered to be the constituents of the RSR according to the Contrast heuristics described in Section 3.5.3.2. Where there is a difference between the span selected by the *Aggressive* and *Inclusive* heuristic, the Aggressive is indicated with a minus (-) sign and the Inclusive with a plus (+) sign.

```
S:
  NP-A:
    NPB: Wall (NNP) Street (NNP) investors (NNS) , (PUNC,)
+[SG:
    VP: citing (VBG)
      NP-A:
        NPB: a (DT) drop (NN)
        PP: in (IN)
-[        NP-A:
            NPB: oil (NN) prices (NNS)    ]]
            PP: because (IN) of (IN)
              NP-A:
[[              NPB: weakness (NN)
                PP: in (IN)
                  NP-A:
                    NPB: the (DT) automotive (JJ) sector (NN) , (PUNC,) ]]
  VP: sold (VBD)
    PRT: off (RP)
    NP-A:
      NPB: shares (NNS)
    PP: in (IN)
      NP-A:
        NPB: s (PRP) big (JJ) three (CD)
          NPB: Detroit (NNP) ' (POS)
    PP: at a record pace today .
```

Figure 3.8: Parse tree for the sentence, "Wall Street investors, citing a drop oil prices because of weakness in the automotive sector, sold off shares in Detroit's big three at a record pace today." (Final PP is collapsed onto one line to save space.) Brackets indicate the spans which are considered to be the constituents of the RSR according to the Contrast heuristics described in Section 3.5.3.2. Where there is a difference between the span selected by the *Aggressive* and *Inclusive* heuristic, the Aggressive is indicated with a minus (-) sign and the Inclusive with a plus (+) sign.

the way up the parse tree to the the sentence gerund (SG). While the inclusion of *citing* is arguably wrong, it does reach the key noun *drop*.

In these two examples, we can see that neither heuristic appears perfect, although each has desirable properties. In the next sections, we evaluate the heuristics themselves, and then examine their effect on the RSR classification task.

3.5.3.3 Evaluating the Heuristics

To evaluate the heuristics themselves, the first question we ask is, how well do our heuristics work in identifying the actual correct RSR extents? We evaluate this against two data sets. First, we use our own marked development set as described earlier ("Dev") which contains 100 marked examples each of Cause and Contrast relations. Second, we use the Penn Discourse TreeBank (PDTB), restricting ourselves to discourse-annotated *but* and *because* sentences which match the RSR patterns which are the subject of our syntactic filtering. Since the PDTB is annotated on the same corpus as Penn TreeBank (PTB) (Marcus et al., 1994), we separately evaluate the performance of our heuristics using gold-standard PTB parses ("PDTB-Gold") versus the trees generated by Collins' parser ("PDTB-Prs"). We extract our test data from the PDTB data corresponding to section 23 of PTB, i.e. the standard testing section, so that the difference between the gold-standard and real parse trees is meaningful. Section 23 contains 60 annotated instances of *but* and 52 instances of *because* which we can use for this purpose. We define the measurement of accuracy here in terms of word-level precision/recall. That is, the set of words filtered by our heuristics are compared to the "correct" words to cut, i.e. those which the annotated RSR extents exclude. We give the results of this analysis in Table 3.8.

Heuristic	Dev Corpus	PDTB-Prs	PDTB-Gold
Contrast	74.2 / 59.0	89.6 / 73.0	79.0 / 80.6
Cause-Incl	74.4 / 67.0	83.8 / 62.1	89.3 / 62.0
Cause-Aggr	80.2 / 74.2	78.5 / 78.8	87.3 / 79.5

Table 3.8: Precision/Recall of syntactic heuristics under various data sets and settings as described in Section 3.5.3.3.

We performed an analysis of our heuristics on Section 24 of the PDTB. In that section, there are 74 relevant sentences: 20 sentences with *because*, and 54 sentences with *but*. In both sets, slightly more than one-third are marked as having at least some text which is not part of the relation, with a mean of 7.9 words per sentence which should be cut in such cases for *but*, and 9.6 for *because*.

Exactly half of all sentences (37) have no problems in the application of the heuristics (7 *because* sentences, 30 *but* sentences). Among the remaining sentences, the main source of problems is that our heuristics do not always remove matrix clauses with verbs of saying (15 cases total, 8 of which are *because* sentences). For the *but* clauses, our heuristics removed the subject in 12 cases, but not the PDTB. Additionally, the heuristic for *but* sentences does not correctly identify the second conjunct in five cases (choosing instead a parenthetical, for instance).

In looking at our Aggressive heuristic for the Cause relationship, we see that they indeed eliminate the most frequent source of discrepancies with the PDTB, namely the false inclusion of a matrix clause of saying, resulting in 15 out of 20 perfect analyses. We assume that the other heuristic problems we detected can also be fixed, but we leave for future work an investigation of the effects of better approximating the human judgments in the PDTB on the classification of RSRs.

The second question we ask is, what is the effect on heuristic accuracy of using automatic (versus gold-standard) parses? We evaluate the difference in performance between the PDTB-Gold and PDTB-Prs performance to determine to what extent using a parser (as opposed to the Gold Standard) degrades the performance of our heuristics. We find that in Section 24, 13 out of 74 sentences contain a parsing error in the relevant aspects, but the effects are typically small and result from well-known parser issues, mainly attachment errors. As we can see in Table 3.8, the heuristic performance using an automatic parser degrades only slightly, and as such we can expect an automatic parser to contribute to improving RSR classification (as indeed it does).

When we use the Aggressive heuristic for the Cause relation, the number of sentences with relevant parsing errors for our 20 *but* sentences increases from two to five. This is because the Aggressive heuristic limits the scope of the two text spans more finely, so parsing errors are more likely to interfere. All errors are errors of attachment, of the *because* clause itself, or of conjuncts or adverbials.

Overall, we see that the performance is sufficiently good to expect a parser to contribute to improving RSR classification (as indeed it does). The parsing problems we encounter are well-known problems for parsers; given that we are not interested in the whole parse, it may be possible to train parsers specialized in making the syntactic choices necessary for our purpose.

3.5.3.4 Training with Syntactically Filtered Data

We retrain the TextRels model using instance data filtered with the syntax-based heuristics, in order to evaluate the resulting classifiers. In terms of the three steps described earlier, we build a syntax-filtered model similarly to the parameter-optimized model discussed earlier, with the following differences:

Instance Mining Inputs: While our input corpus T remains the sentence-segmented Gigaword corpus, we limit ourselves to the two patterns mentioned previously, i.e. "W_1 because W_2" for Cause and "W_1 , but W_2" for Contrast, for which we have derived syntactic filtering rules.

Outputs: We randomly select from our Opt (i.e. non-segment-constrained) instance set 400,000 instances of Cause and Contrast found by the patterns above, and 400,000 instances of NoRelSame.

Model Building Inputs: We parse the instance examples I for Cause and Contrast and and then filter according to the syntactic heuristics described above.

Outputs: The frequency counts F, using the filtered instances and model parameters ϕ_m optimized earlier, except that we lower the Vocabulary Size parameter to 3200 since those earlier results indicated this to be a better size when working with a smaller number of instances.

Classification Inputs: The frequency counts F and the classification parameters ϕ_c as optimized in the previous section, except for adjusting the Laplace λ to a slightly higher value (0.5) since our earlier results indicate this to be a better value when working with sparser data.

Output: A set of binary classifiers which we evaluate below.

3.5.3.5 Evaluation

We evaluate the impact of our syntactic heuristics on classification over the Auto and PDTB test sets. We primarily compare the syntax-filtered ("Filtered") model described above with an "Unfiltered" baseline, which is prepared in the exact same manner (and on the same reduced-size training set), but with no filtering of the training instances.

Since a scarcity of training data is a significant concern in these experiments, due to the resource constraints of requiring parsed training examples, we focus first on a set of experiments which evaluate accuracy as a function of the number of training examples in I.

Figure 3.9 shows the performance of our Cause versus NoRelSame classifier on the Auto test set when using the Aggressive syntactic heuristic to trim Cause instances. We observe that the filtered model, while appearing to gain accuracy at a more rapid rate as training examples are added, achieves only equal performance with our 400,000 example set.

Examining the data further, we realize that the Aggressive heuristic, while appearing much more "accurate" in terms of identifying PDTB-marked relation spans in our error analysis (Section 3.5.3.3), is simply filtering out much more data. Figure 3.10 illustrates this phenomenon, showing the same accuracy points as in Figure 3.9, but as a function of the total token pair frequencies in F_{cause}, rather than as a function of the total instance pairs in I_{cause} (the number of pairs in the NoRelSame model is equal in both cases since that model is not syntactically filtered). We see in this figure that the Filtered model for Cause has approximately half the total token pairs of the Unfiltered model

When we examine the performance of the Cause versus NoRelSame classifier using the Inclusive heuristic for filtering Cause instances, we find that it indeed results in a more accurate Cause versus NoRelSame classifier. As shown in Figure 3.11, the performance with 400,000 training examples with the filtered model is 0.717, as compared to 0.703. The difference is significant using the Chi-squared test ($p < .05$).

When evaluated on the Auto test set, the effect of syntactic filtering is more modest.

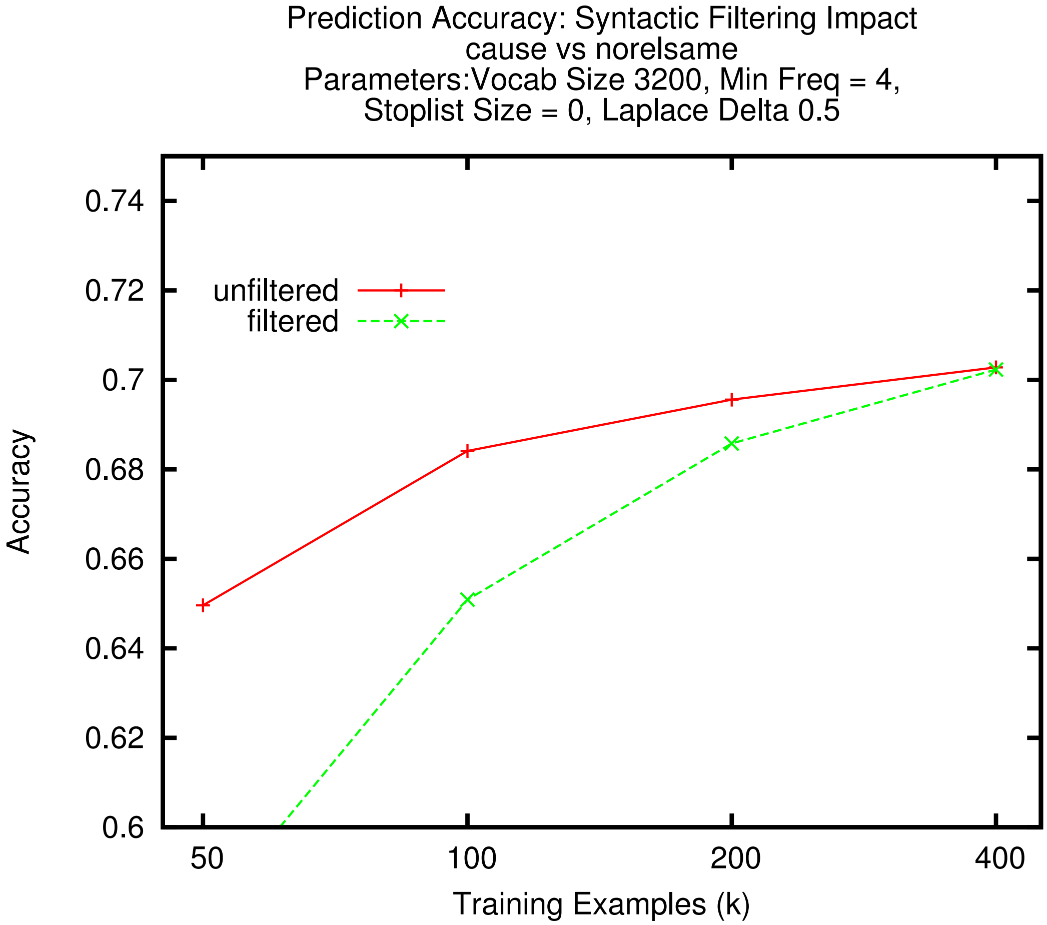


Figure 3.9: Accuracy of Cause versus NoRelSame classifier as a function of training examples in Filtered versus Unfiltered models, using Aggressive filtering heuristic for Cause model.

We have a small gain in performance in Contrast versus NoRelSame (Figure 3.12), and a small decrease in performance in Contrast versus Cause (Figure 3.13), using the Inclusive heuristic for Cause gives better performance here, however.) In neither case is the change statistically significant.

Finally, we focus on the best performance for each configuration (i.e. using the full 400,000 training instances) across the Auto and PDTB test sets in Table 3.9. In addition to the Unfiltered baseline, we show the results for both the Inclusive and Aggressive versions of the Filtered model (this distinction does not impact the Contrast vs NoRelSame classifier, since only Cause has the two versions of the filtering heuristic). Further, to provide a point of comparison in terms of syntactically-founded filtering, we also evaluate the Part-

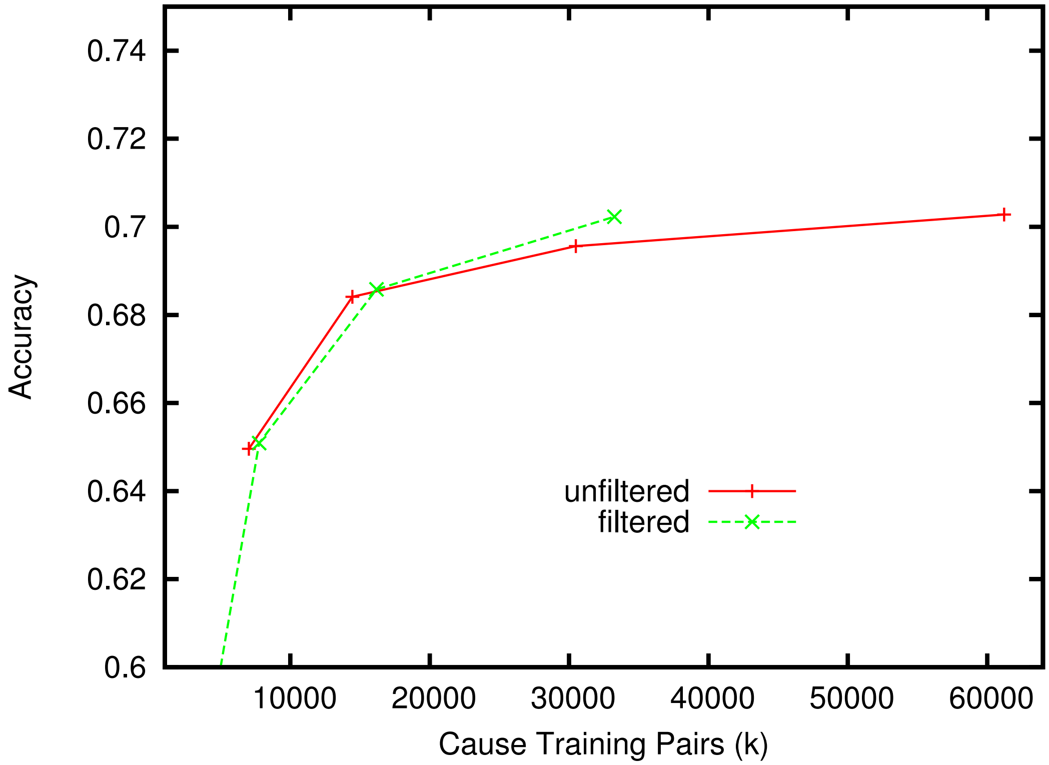


Figure 3.10: Accuracy of Cause versus NoRelSame classifier as a function of F_{cause} token pair frequency in Filtered vs Unfiltered models, using Aggressive filtering heuristic for Cause model.

of-Speech based filtering heuristic described by Marcu and Echihabi. Their approach filters the instance set I by retaining only nouns and verbs, before computing the pair frequencies F. Marcu and Echihabi suggest that this technique may outperform an unfiltered model when trained on the same number input instances; we wish to see how it compares with our syntactically-based filtering rules. We carry out this comparison by implementing their POS-based filtering over the same sets of training data which we use to test our syntactic filtering rules (i.e. the 400,000 examples for Cause, Contrast and NoRelSame). Lastly, for reference, we also include in the table the accuracy of the Opt classifier described in Section 3.5.1; note that unlike the other classifiers in the table, this classifier is trained on the full training sets.

Our first observation is that the difference between performance between the two Cause heuristics differs based on the test set. Given our heuristic analysis, which showed the

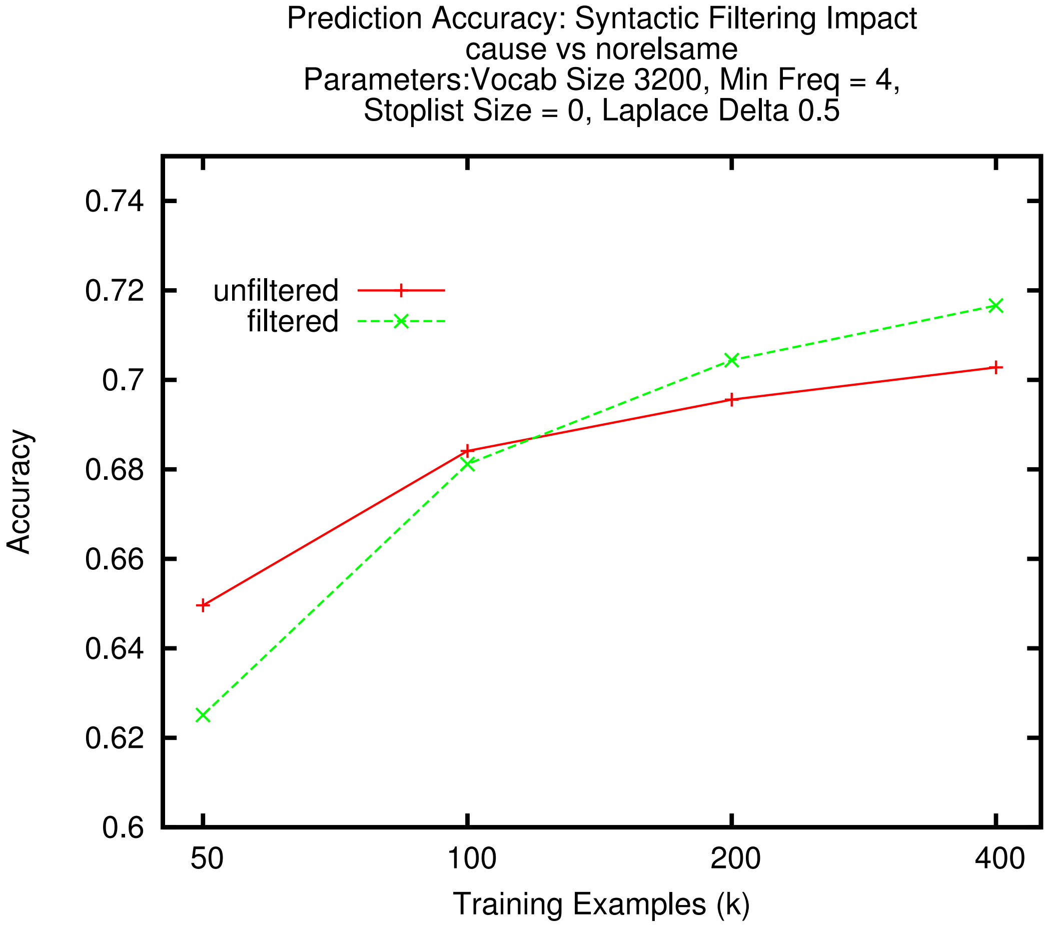


Figure 3.11: Accuracy of Cause versus NoRelSame classifier as function of training examples in Filtered vs Unfiltered models, using Inclusive heuristic for Cause model.

	Pdtb	Test	Set			Auto	Test	Set		
	U	**I**	**A**	**P**	**O**	**U**	**I**	**A**	**P**	**O**
Cause vs Contrast	59.6	59.2	60.5	54.5	59.1	66.3	66.2	65.8	60.8	69.8
Cause vs NoRelSame	72.2	72.5	74.9	52.6	75.2	70.3	71.7	70.2	57.3	72.7
Contrast vs NoRelSame	61.6	60.2	-	52.2	67.4	69.4	69.8	-	56.8	70.7

Table 3.9: Classifier accuracy for the Unfiltered (U), Filtered-Aggressive (A) / Inclusive (I) and POS (P) models described in Section 3.5.3.5, and Opt (O) model described in Section 3.5.1, over PDTB and Auto test sets. Baseline in all cases is 50%.

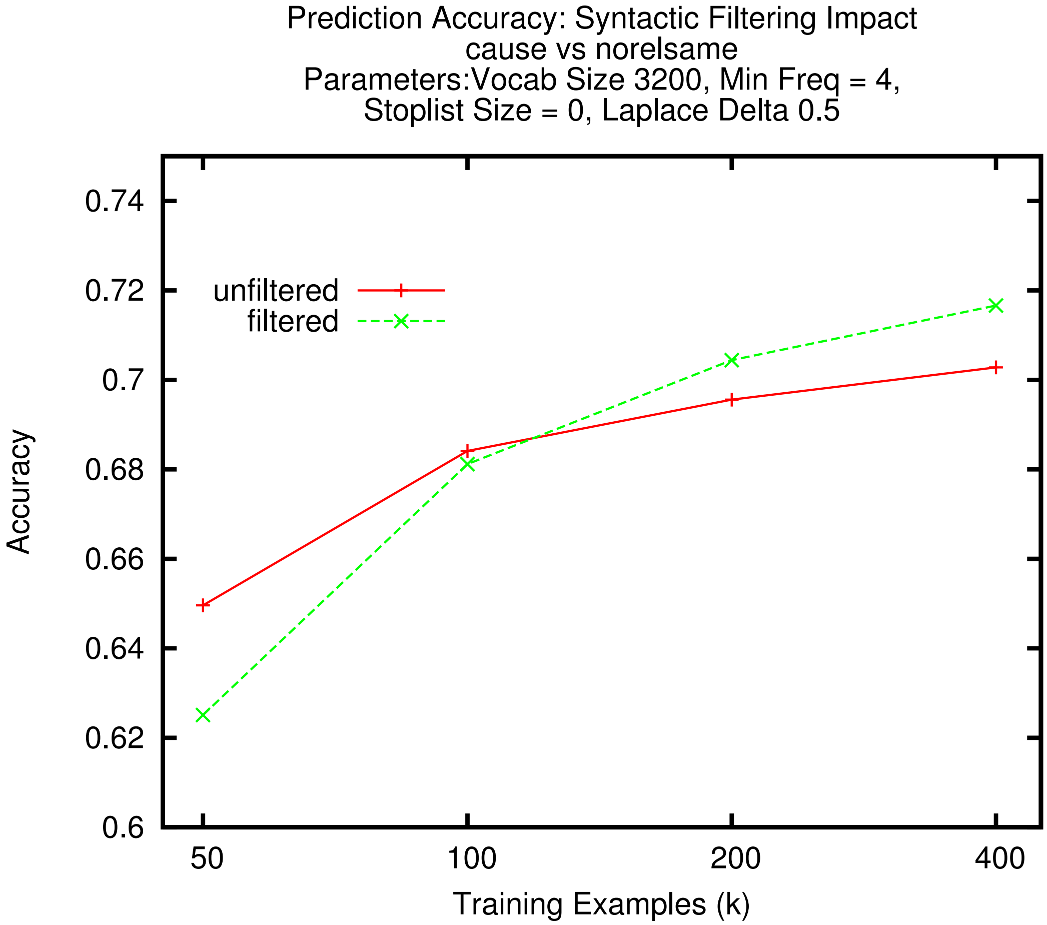


Figure 3.12: Accuracy of Contrast versus NoRelSame classifier as a function of training examples in Filtered versus Unfiltered models.

Aggressive heuristic to achieve much higher recall while sacrificing little precision, we would expect the Cause classifiers to perform better with this heuristic. And indeed, both the Cause vs NoRel and Cause vs Contrast classifiers do perform better on the PDTB test set under the Aggressive heuristic. However, on the Auto test set, we observe the opposite effect: the Aggressive heuristic performs worse than Inclusive, and in fact slightly worse than Unfiltered. As described above, we attribute this behavior to the fact that the Aggressive heuristic simply cuts a large portion of training data out. Yet even with this “lost” data, the Aggressive heuristic yields a better Cause vs NoRel classifier on the PDTB test set.

Examining the data further, we find that the Aggressive heuristic simply cuts a large portion of training data out. In terms of the total sum of frequencies in F_{cause}, i.e. the

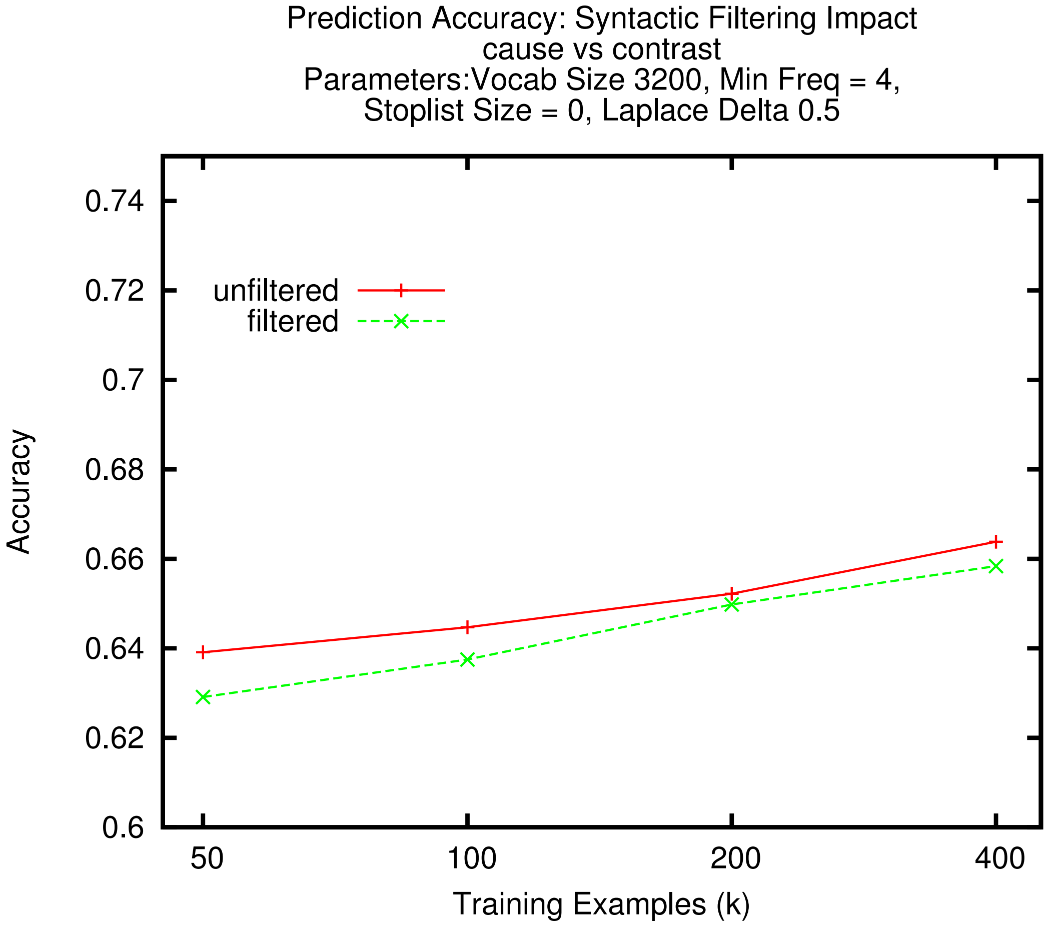


Figure 3.13: Accuracy of Cause versus Contrast classifier as function of training examples in Filtered versus Unfiltered models. The Inclusive heuristic is used for filtering the Cause instances.

word pairs extracted from all cause instances, the Aggressive filtering cuts out nearly half. With this in mind, we can say that while the Aggressive filtering only achieves (virtually) equal accuracy with the Unfiltered baseline on the Auto test set, that the pairs it does keep appear to be used more efficiently. Yet even with this lost data, the Aggressive heuristic yields a better Cause vs NoRel classifier on the PDTB test set.

We posit that this difference in behavior across the two training sets suggests that the Auto set may maintain some bias toward spans which, while not true arguments of the relation in the PDTB sense, nonetheless occur regularly with the cue phrases used in instance mining and thus are more likely to be present in the test set. Nonetheless, syntactic filtering again achieves the best performance for the Cause vs NoRel classifier in the Auto set; as noted earlier, the improvement of Inclusive over Unfiltered baseline is significant by the Chi-squared test ($p < .05$).

For the Contrast versus NoRel classification, the syntactic filtering has a negative effect on accuracy on the PDTB test set. On the Auto test set, there is a slight positive effect. However, neither difference is statistically significant.

Surprisingly, we also observe that the POS-based filtering described by M&E performs uniformly poorly. We have no explanation for this at present, given that M&E's results with this filter appear promising.

Lastly, we observe that, for the most part, the results obtained with the Opt classifier using its larger training set are still the best overall. However, surprisingly, for the Cause versus Contrast classification, the Opt classifier performs slightly (although not significantly) worse than the Unfiltered one, even though the former has approximately seven times as many training examples for Contrast and five times as many for Cause; this result indicates that, for the *implicit* relation instances of the kind we get from PDTB, the additional training data is not improving our model in the absence of syntactic filtering. Interestingly, the highest score we get is for the Aggressive heuristic, even thought that pares down the smaller training set even further. However, we also note that because of the relatively small size of the PDTB test data set, none of the results (except the POS-filtering) are significantly bette or worse than any others.

Despite this, the Opt classifier does achieve the best results on the Cause vs NoRelSame and Contrast vs NoRelSame tests of PDTB data, in the latter case by a significant margin. In these cases, even though the syntactic heuristics do improve results as discussed above when the number of training instances is constant, this difference is more than made up by the much greater number of training examples used for the Opt classifier. The exception of the Cause vs Contrast, while surprising, only motivates further the need for using the training data that we have more intelligently rather than simply adding more instances.

3.6 Conclusions and Future Work

The experiments we report in this chapter show important results in extending the RSR modeling approach of Marcu and Echihabi (M&E). In addition to showing their results to be generally repeatable, we have expanded on their work in both depth and breadth.

In depth, we show that the selection of appropriate parameters for the modeling and classification task has strong bearing on the classification accuracy task, and find appropriate levels for these parameters which allow us to achieve comparable accuracy to M&E's original work using the online-efficient implementation in TextRels.

In breadth, we investigate two new directions for addressing a key concern with this modeling technique, namely the quality of automatically-extracted examples. In the first set of experiments, we use automated topic segmentation to refine the extraction patterns for our RSRs. While we find that the effect of this approach is mixed, it does improve performance in some cases. In particular, it improves mean classification accuracy with respect to the PDTB/manually-annotated data. For the automatically annotated data, topic segmentation-based filtering generally decreases accuracy for classifiers involving the Adjacent relation, but increases accuracy for classifiers which do not classify with respect to Adjacent. For automatically annotated test data which is prepared using segmentation-based heuristics (i.e. the "Auto-S" test set), the overall performance improvement with respect to classifiers which do not involve the Adjacent relation is statistically significant.

We also investigate the efficacy of using syntactically motivated filters for more accurately finding the extent of RSR-related text spans. Using both a hand-annotated corpus and data from the PDTB, we evaluate the accuracy of our syntactic heuristics with respect to both gold-standard and automatic parses. Applying the syntactic filters to a large subset of our training data, we build classification models, and find that these models improve over an unfiltered baseline in several cases, in particular with regard to the model of a Cause relation, for which we analyze two different syntactically-motivated heuristics. While the topic-segmentation filtering approach achieves the best results overall, our analysis of the syntactic filtering approach indicates that refined heuristics and a larger set of parsed data can further improve those results. That is, in the case of the Cause heuristics, we find that the Aggressive heuristic's better performance in terms of correctly identifying relation "arguments" in PDTB carried over to better classification performance on the PDTB-derived test set.

In terms of future work, this last point is quite encouraging, especially given that the PDTB gives us a substantial data set on which to develop more refined heuristics for capturing cleaner training data for a number of RSRs. Combining more refined filtering rules with a larger volume of parsed training instances is thus a promising avenue for building RSR models. Moreover, we would also like to experiment with combining the segmentation and syntax approaches; a straightforward possibility would be to simply apply the filters in sequence, i.e. by applying the syntactic heuristics to an instance set extracted using topic segmentation constraints.

Another important area for future work is experimentation with techniques to reduce dimensionality in the internal representation of TextRels RSR models. Currently, we use a full-dimensionality matrix based on actual word stem pairs. This straightforward representation allows us to maintain an intuitive model which can be easily manipulated and examined in the various experiments and model variants we present in this chapter. But mapping to a lower-dimensionality model would increase efficiency and conflate individual terms which may have similar distributions with respect to our relations of interest. This kind of dimensionality reduction is typically done with singular value decomposition (SVD), perhaps most commonly in latent semantic analysis (Deerwester et al., 1990), and could be used in an analogous way on our RSR models.

Overall, the work in this chapter achieves a twofold purpose. First, we successfully adopt and extending the M&E model, and show that by using multiple, complementary techniques, we can successfully refine RSR models and improve our ability to classify unknown relations. Second, we create in TextRels a resource which we can use to apply RSR-derived features in the context of long-answer question answering. We explore such applications of TextRels in the next two chapters.

Chapter 4

Applying RSRs in DefScriber

4.1 Introduction

In this chapter, we experiment with applying the TextRels Rhetorical-Semantic Relation models developed in Chapter 3 as a resource for improving DefScriber. Using data from the Document Understanding Conference (DUC) evaluations, we analyze the impact of using RSRs on the tasks of both biographical- and topic-focused long-answer questions. Whereas in Chapter 3, our evaluation focused on the intrinsic differentiability of our RSR models as measured on a classification task, we investigate here whether the models can be helpful as a component in the applied task of long-answer question answering.

Our motivation for exploring the potential contribution of RSRs in the context of these questions can be seen in the following example, taken from the actual DefScriber answer for "Who is Sonia Gandhi?" in the DUC 2004 biography task:

Example 4 *The BJP had shrugged off the influence of the 51-year-old Sonia Gandhi when she stepped into politics* **early** *this year, dismissing her as a* **foreigner***. Sonia Gandhi is* **now** *an Indian* **citizen***.*

These two sentences are found in separate source documents and placed together in a way that is clearly well-chosen. The ordering here enhances *cohesion*, or a fluent, smooth flow of related concepts. It also improves *coherence*, making the information more easily comprehensible by virtue of reinforcing important themes in context.

What is it that explains the felicitousness of a sentence pairing and leads to these desirable properties? Other work has examined the contribution of entity realizations and transitions (Barzilay and Lapata, 2005; Nenkova, 2006), global ordering strategies (Barzilay et al., 2002), and latent semantic models of relatedness (Foltz et al., 1998) in explaining phenomena of cohesion and coherence.

However, we posit that in this case, the pairing of these sentences is good largely because of rhetorical-semantic relationships. Specifically, the cross-sentence word pairs *early/now* and *foreigner/citizen* make the passage more fluent via semantic contrast, and more comprehensible by creating implied rhetorical opposition between evoked concepts which emphasizes their importance and reinforces the passage's meaning. Yet our earlier versions of DefScriber, including the one which created this answer, do not use any such rhetorical or semantic concepts, and are lucky to create such a felicitous passage.

We experiment in this chapter with methods which aim to actively create stronger answers which include these kinds of links, using our TextRels RSR models of Cause, Contrast and Adjacent. Using the models in this way is meant to complement methods of content ordering and selection already in place in DefScriber. These existing methods are primarily implemented in the "Bottom-Up" modules of DefScriber described in Chapter 2, using variations of established techniques like centroid-based summarization (Radev et al., 2000) and lexical chaining (Barzilay and Elhadad, 1997). While robust, they do not consider RSR-like relationships, which we theorize may help produce more passages like Example 4 in our answers.

Integrating RSRs into DefScriber using TextRels is especially appealing given that TextRels is built over a broad corpus of examples which learn the ways in which real authors organize text and express rhetorical and semantic relationships. That is, while our "Genus-Species" predicate taps into an important kind of information type for inclusion in definitions and other descriptions, we implement it using manual patterns which focus on the surface form of the relation and can suffer from low recall. Here, we pursue the inclusion of relation types which are interesting for "Top-Down" reasons, but via an implementation which models the content of relations in a "Bottom-Up" statistical manner, so as to have high coverage across various domains and the ability to find relationally-linked information without relying on particular surface syntax.

Other bottom-up similarity measures have been used in text ordering, including lexical relatedness measures (Foltz et al., 1998; Higgins et al., 2004) like LSA for assessing coherence, or combined lexical-syntactic models for text-to-text generation (Lapata, 2003). Unlike these approaches, RSRs link back to our initial motivation in proposing *definitional predicates* in DefScriber, namely that they model *a priori* interesting classes of relatedness, like Cause and Contrast.

As a whole, this chapter addresses the third research question which we address in the introduction to this thesis, namely: How can we apply this rhetorical-semantic model to enhance the performance and scope of our question answering techniques? In particular, we focus in this chapter on performance issues, using a supervised learning approach for integrating RSR models to improve performance on two question types first discussed in Chapter 2.

The remainder of the chapter is organized as follows. We begin with a discussion of related work in Section 4.2, focusing on approaches to measuring cohesion and coherence in summarization. We then describe extensions to DefScriber which allow us to integrate RSR-based features alongside existing system features in a common framework for sentence selection (Section 4.3). We take advantage of this common framework to conduct supervised training in which we learn weights for RSR and other features based on data from the DUC 2004 biographical summarization task, and DUC 2005 topic-focused question task (we discuss our original participation in these tasks in Chapter 2, Sections 2.4.2 and 2.4.3 respectively). We implement these experiments in two stages, which evaluate the impact of our various features on two complementary qualities which we aim to improve via the use of RSRs. In Section 4.4, we consider whether *content selection* can be improved by using these models, estimating quality via ROUGE (Lin, 2004) scoring. In Section 4.5, we focus on *content ordering* performance using a metric which scores the number of ordering inversions with respect to an ideal summary. In both kinds of experiments, we analyze the impact of RSR-specific features separately from the impact of other improvements. We find that the addition of RSRs, as well as other attendant system upgrades, can improve performance in content ordering and selection, although only some of the improvements rise to the level of statistical significance.

4.2 Related Work

The idea of using RSRs within the context of creating coherent answers follows some of the key intuitions of lexical chaining(Barzilay and Elhadad, 1997), a prominent technique in text summarization. The general principle of identifying strands of related information has been extended with various techniques in the past, e.g. by using WordNet-linked terms in addition to simple lexical identity (Silber and McCoy, 2000) to form lexical chains.

Statistical similarity models have also been applied for the task of evaluating (Foltz et al., 1998; Higgins et al., 2004) and creating (Miller, 2003; Hachey et al., 2005; Jagadeesh et al., 2005) coherence in multi-sentence answers or summaries. Creating coherent summaries with the aid of latent semantics has shown only mixed results, however. Jagadeesh et al. report a slight negative impact from using LSA as evaluated by DUC topic-focused summary ROUGE scores; Hachey et al. and Miller both report slightly (but not significantly) positive results on DUC-like tasks. This is somewhat surprising given that the general principle of grouping strongly-linked topics which LSA should intuitively support is validated as important by Barzilay et al.'s empirical analysis (Barzilay et al., 2002). In that work, chronological and original-document-based ("majority") ordering are found insufficient strategies for summary ordering, while constraints which group clusters of topically related information together are more reliable.

In addition to topical relatedness based on lexical identity or latent semantics, entity-centric approaches have also been shown useful in summary coherence. Barzilay and Lapata (Barzilay and Lapata, 2005) evaluate an entity-transition based model of sentence ordering, focusing on the ways in which named entities are realized across successive sentences. They find that by tracking transitions in presence and manner (e.g., syntactic role) of entity mentions, they can achieve results in ordering and summarization tasks which outperform an LSA baseline. Nenkova (Nenkova, 2006) finds that in extractive summaries, fluency can be improved by rewriting entity mentions to follow learned rules for the way coreferent noun phrases are realized in natural text.

Lapata (Lapata, 2003) and Barzilay and Lee (Barzilay and Lee, 2004) implement sequence models for text ordering which learn both lexical and syntactic constraints. Particularly in Barzilay and Lee, the impressive results for the ordering task indicate that for stories on certain topics (e.g., drug-related crimes or financial reports), that strong Domain Communication Knowledge-like (Rambow, 1990) structure can be effectively learned and represented in their HMM formalism. However, it is unclear how well this approach will perform on unrestricted topic texts. By contrast, Lapata finds that a shallower model of local structure measured by verb, noun and dependency-structure transitions, can capture ordering regularities of newswire text successfully without constraining the text topic.

Work which investigates the use of rhetorical models in summarization has been limited. Most notable are Marcu's experiments with discourse structure and summarization (Marcu, 2000), which find that in single-document summarization, rhetorical structure can usefully identify key sentences, for instance by summarizing a text by visiting nucleus nodes and skipping satellites in traversing an RST (Mann and Thompson, 1988) tree. But the work of many others, going back to schema-based generation (McKeown, 1985; Maybury, 1989) or plan-based understanding (Schank and Abelson, 1977) supports the importance of rhetorical and semantic relations in coherent text. However, the difficulty in identifying relations automatically in the general case, particularly when their structure is complex, is acknowledged by Marcu (Marcu, 2000) as an important limitation of using such an approach in a fully automated online setting. And while LSA-based techniques can take advantage of unsupervised training data to build larger and more accurate models, the work discussed above casts doubt on whether the relatedness measured by such models may simply be too general to be of use in tasks like multi-document summarization or long-answer QA. The application of automatically trained models of specific relations, as in Marcu and Echihabi's work (Marcu and Echihabi, 2002) and our TextRels implementation, has not been attempted in this context to our knowledge. Moreover, the question of which relations can be most profitably used in such a context is also an open one; however, as noted in Chapter 3, while foundational work on rhetorical and semantic relations often differs on issues of relation taxonomy, there are basic recurring themes. For instance, causal and contrastive relationships are consistently recognized as important even if their precise descriptions vary

(Hovy and Maier, 1993; Moser and Moore, 1996).

4.3 Extending DefScriber for RSR Integration

Our primary motivation in extending DefScriber is to experiment with RSR-based features to detect the kinds of cross-sentence links illustrated in Example 4. In order to integrate these kinds of features into DefScriber in a principled way, however, we see the need for creating a common framework where their effectiveness can be measured alongside other existing features of centrality, lexical cohesion and coverage. In addition, we have noted in our manual analysis of previous results several features which we would like to add or refine. Thus, we take the occasion of preparing the system for RSR integration to update several aspects of DefScriber. Moreover, we decide to use a supervised learning approach to integrate these new features, leveraging the data sets available from the DUC 2004 (Over, 2004) biographical question task and DUC 2005 (Dang, 2005) topic-focused question task.

Our point of departure for these experiments is the method currently used for selecting answer content in DefScriber, described in Chapter 2. To recap, DefScriber selects a *first* sentence using a combination of data-driven topic centrality measures and goal-driven heuristics that select for particular sentence types. In this chapter, we focus on improving the method for choosing the *remaining* sentences in the answer, i.e. how to pick a good next sentence given that the previous one has already been selected. In our earlier DefScriber implementation, we select each following sentence by maximizing a weighted combination of topic centrality, lexical cohesion, and topic coverage. However, this procedure has several limitations, particularly with regard to implementing new features, which we address here.

Feature / Description	**Calculation Method**
Centroid Relevance to overall topic.	IDF-weighted, word-stem vector cosine similarity between s and S. Remains static as P is built.
Query Overlap with query terms.	IDF-weighted, word-stem vector cosine similarity between s and the query q. Can be calculated statically, or adjusted as P is built by de-weighting terms already "covered" in the answer.
Coverage Coverage of sub-topics.	One if c has least (or tied for least) representatives already in P as compared with other clusters in C; zero otherwise. Dynamically adjusts as P is built.
Lexical Cohesion Lexical cohesion with previous sentence.	IDF-weighted word-stem vector cosine similarity between s and p. Dynamic adjusts as P is built.
Document Cohesion Original document cohesion.	Score is non-zero iff s and p are from the same input document, and p preceded s in original ordering; score is inversely proportional to the number of sentences between p and s. Dynamically adjusts as P is built.
RSR Cohesion RSR-based cohesion with previous sentence.	Calculated by "classifying" whether s and p appear to be related by our models for Cause, Contrast and Adjacent RSR relations. Combining the results of these classifications is discussed in Subsection 4.3.2. Dynamically adjusts as P is built.

Table 4.1: Features used in DefScriber's sentence selection algorithm. Features are evaluated for a candidate sentence s to be added to an under-construction summary P with last sentence p. C is a clustering over S such that each s belongs to exactly one cluster $c \in C$; s is a member of exactly one cluster c in a set of clusters C which exhaustively and exclusively partition the full set of relevant input sentences S. Note that the value of most features is recalculated dynamically as the in-progress answer is formed.

4.3.1 Common Feature Framework

The broadest extension to DefScriber is the implementation of a common framework for all of the features listed in Table 4.1. We define our features such that when considering a candidate sentence c for addition to an in-progress response, c can be represented by a single vector of feature values. Combining all of our features in a single framework enables us to more easily train DefScriber using a supervised learning approach, as described in Section 4.4.2.

In this way, we make DefScriber amenable to supervised training for the integration of any new features, including RSR- and non-RSR features alike. While training of summarizers/response generators in this way goes back to Kupiec et al. (Kupiec et al., 1997), it has recently shown to be useful for the DUC 2005/6-type question-focused summarization task by Daumé and Marcu (Daumé III and Marcu, 2005) and Fisher and Roark (Fisher and Roark, 2006).

4.3.2 RSR Features

The first feature to add using this framework is one which utilizes our TextRels RSR models. We decide to implement an RSR Cohesion feature which leverages the binary classifiers built for our evaluations in Chapter 3. These classifiers determine the relative likelihood that a given text span pair is related by one versus another RSR. More specifically, we decide to use the three binary classifiers which measure the likelihood of a given span pair under {Cause, Contrast, Adjacent} versus NoRelSame RSR models. Our intuition here is that in the context of DefScriber answers, we are primarily concerned with finding strong RSR links, rather than maximizing our confidence about the precise nature of a relation. That is, we are focused at this point on determining whether *some* relation holds, i.e. making "something versus nothing" judgments, and not ready to be as picky about what the "something" is.

More formally, our goal is to experiment with the usefulness of a feature which measures whether a candidate sentence s relates to another sentence p in a way that resembles any one of our Cause, Contrast or Adjacent models. Recall that our classifiers allow us to estimate this by measuring the relative likelihood of whether two sentences appear to be related by, e.g., Cause, or more closely resemble a baseline model of weaker topic-relatedness.

This is done in our classifier by comparing $P(p, s|Cause)$ and $P(p, s|NoRelSame)$, using individual word pair probabilities to approximate the sentence pair probabilities.[1] We can use the difference in these estimated probabilities to compute a normalized value from zero to one which expresses how Cause, Contrast and/or Adjacent-like a given sentence pair is. In addition to these probabilistic calculations, we also give full confidence to the Adjacent hypothesis for the rare case where s and p are actually adjacent in an original source document and for Cause and Contrast if they are joined by the appropriate cue phrase.

In our experiments with these features, we also consider several ways to compute a single value for the RSR Cohesion feature value as listed in Table 4.1, namely by allowing separate free parameters for:

Cause, Contrast, Adjacent weights Three separate parameters which multiply the base zero-to-one value calculated by comparing $P(p, s|R_k)$ and $P(p, s|NoRelSame)$ where R_k is one of the three RSR relations. We use separate parameters to enable learning how to interpret the probabilities of each model independently.

Combine mode Parameter which determines how to combine the three individual Cause, Contrast and Adjacent scores; can take on two settings, to either take the mean or maximum of scores; meant to learn relative importance of a single RSR model being very strongly matched, versus several RSR models all having a medium-strength match.

We experiment with how to set these parameters as part of our system training in Sections 4.4.2 and 4.5.4.

4.3.3 Other Features

In addition to this RSR Coherence feature, DefScriber uses several additional features to assess the "goodness" of a candidate sentence with respect to an in-progress response. The

[1] For our experiments in this chapter, we use the TextRels model versions which achieve the best classification scores for the "X vs NoRelSame" classifications in Chapter 3. These are the models trained on topic-segment constrained instance data, using training sets 1+2, and other model parameters as optimized in Section 3.5.1.

Centroid, Coverage and Lexical Cohesion features are essentially unchanged from previous versions of DefScriber and are summarized in Table 4.1. However, we also add two new features:

Query Score This new feature quantifies the degree to which a given sentence occurs in an area of dense query terms. Whereas we previously identify a relevant "NSD" sentence as a binary decision, and then determine centrality based only on Centroid score, this score adds a scalar dimension based on how close a sentence is to the question. This is distinct from the Centroid score, which determines how close a sentence is to the set of NSD sentences as a whole. This feature is especially targeted at the DUC 2005-style long-form questions, where there can be a high degree of variability in how much a given sentence overlaps a question. In addition, for these topic-focused questions, we dynamically adjust this score so that terms which are already "covered" in the in-progress answer are DE-weighted in assessing a candidate sentence's Query score. For biographical questions, the score remains constant (i.e., because we do not expect that because a persons name is mentioned in their biography already that it becomes "covered").

Document Cohesion This new feature reflects the cohesion boost which we assume is likely when we reproduce a sentence ordering from an original document. The actual score assigned is inversely proportional to number of sentences between the candidate sentence s and the last sentence in the in-progress summary.

Note that the features described in Table 4.1 vary in how dynamic they are with respect to an in-progress response. For instance, the Centroid score for a candidate sentence is completely static and depends only on the set of NSD sentences as a whole, which do not change as the response is built. The Coverage score, on the other hand, is dynamic with respect to the response as a whole, since it considers which clusters have been covered in the entire in-progress response; the Query score is dynamic in this way for topic-focused questions, but static for biographical ones. By contrast, the Cohesion features (RSR, Lexical and Document) are defined so as to depend only on the last sentence in the in-progress response. We discuss in our future work section how these different levels of dynamism might be reconsidered in future approaches.

4.3.4 Revised Sentence Selection Algorithm

Using our common feature-based framework for old and new features alike, we can re-implement our sentence selection algorithm, which we imaginatively call ChooseNextSentence. As in the previous implementation, we use an iterative algorithm which, at each iteration, greedily chooses a "next" sentence for the summary by maximizing a combination of the values in our common feature framework. While in principle, any combination function can be used, for simplicity we currently use a weighted sum of feature values. DefScriber's answer summaries are created by repeatedly choosing a "next" sentence until a length criterion has been met (or no input sentences remain). However, we create our answers within a beam search framework, where the most promising n summaries (in terms of total feature values of added sentences) are kept at each iteration in order to counteract the possibility of local maxima which can result from greedy choices in the presence of several order-dependent features. In the experiments reported here, we use beam width $n = 8$, derived as a reasonable level which appears to improve response quality without degrading performance in development experiments.

Note several important points in DefScriber's operation which precede the call to the ChooseNextSentence algorithm:

- The set S of "Non-Specific Definitional" (NSD) sentences, i.e. sentences which have any broad relevance to the topic being described/defined, are identified and separated out from the (possibly) larger set of all input document sentences.

- The sentences in S are clustered into a clustering C such that each $s \in S$ belongs to exactly one $c \in C$.

- "Genus-Species" (GS) sentences which provide a category-differentiator (or "is-a") statement for the topic being described, are identified from among S using lexical-syntactic patterns.

- The first sentence in the summary is chosen as the highest- ranked (by Centroid feature) GS sentence, or simply highest-ranked sentence in S if no GS sentences were found. This sentence is passed into the algorithm as the initial, single-sentence summary P. (Within the beam-search framework, we construct n initial responses P, using GS sentences in order of ranking, and then non-GS sentences in order of ranking.)

Following these initial steps, DefScriber repeatedly selects the next sentence to add to the in-progress summary P by repeated calls to the ChooseNextSentence() algorithm listed as Algorithm 1. In the next section, we conduct supervised training to determine good settings for the weight vector W which is one of the algorithm input parameters.

Algorithm 1 ChooseNextSentence(P, C, F, W)

```
INPUT:
$P$ partial summary with $> 0$ sentences
$C$ set of candidate sentence clusters
$F$ set of features to evaluate for candidate sentences
$W$ set of weight parameters defined over all $f \in F$

OUTPUT:
the best candidate sentence $n$ to choose for extending $P$

// GetBestUnused will extract from each cluster in $C$
// the sentence with highest Centroid score not in $P$
$B \leftarrow GetBestUnused(C, P)$
for all $b \in B$ do
  $Score[b] \leftarrow 0$
  for all $f \in F$ do
    $Score[b] += CalcFeatScore(f, b, P) * W[f]$
  end for
end for
return $b \in B$ with maximum $Score[b]$
```

4.4 Content Selection Experiments

Our first set of experiments focus on using DefScriber's updated feature set to produce answers which contain maximally relevant content. We use data from DUC conferences to learn how to combine the RSR and other features when constructing answers for both biographical and topic-focused questions.

4.4.1 Data

We use two sets of data for carrying out our experiments. The first is from the DUC 2004 biography-focused summarization task, where the system is given a set of documents which contain information about a given person and asked to create a summary which implicitly answers a "Who is X?" question. There are 50 topics in this task, focusing on various individuals from John F. Kennedy, Jr. to Sonia Gandhi to Steven Hawking. The data from this task which we have at our disposal include the summaries created by human annotators (four per topic) and by other peer systems.

Title VW/GM Industrial Espionage

Question Explain the industrial espionage case involving VW and GM. Identify the issues, charges, people, and government involvement. Report the progress and resolution of the case. Include any other relevant factors or effects of the case on the industry.

Title Threat to Wildlife by Poachers

Question Where have poachers endangered wildlife, what wildlife has been endangered and what steps have been taken to prevent poaching?

Title Fertile Fields

Question Discuss making and using compost for gardening. Include different types of compost, their uses, origins and benefits.

Table 4.2: Some example topics from the DUC 2005 test set (d311i, d407b, d694j).

The second data set is from the DUC 2005 topic-focused question task, where the task is to create a response to broad, multi-part questions structured as shown in Table 4.2. There are 50 topics from this task as well, and the data we have include the summaries created by human annotators (at least six per topic) and by other peer systems.

For both data sets, we randomly segment into 4 development topics, 8 test topics, and 36 training topics.

4.4.2 Training

In these experiments, our goal is to train the set of feature weights W to input to the ChooseNextSentence algorithm, which will be applied over the six features listed in Table 4.1 (Centroid, Coverage, Query, Lexical Cohesion, Document Cohesion, RSR Cohesion). In order to simplify the problem, we combine the feature values in a straightforward weighted sum. Still, if we consider only five possible weights for each, training over possible permutations would involve 5^6 settings. Since our system takes several minutes to summarize a single DUC input set, an exhaustive search was not a feasible approach. Moreover, it is not even a clearly desirable approach, as it would likely result in overtraining.

Instead, we would like to find a good, if not necessarily optimal weighting using a supervised training approach. In this scenario, we can consider the feature weighting W and resulting ROUGE score for a set of responses using that weighting as our input-output pairs. Since these pairs are relatively expensive to generate in terms of CPU time, we decide to use a guided search to intelligently explore the space of possible weightings. We use a hill-climbing algorithm for this purpose, using random restarts to avoid local maxima.

In order to have an effective hill-climbing search, we need an objective function which tells our algorithm whether a given point in the search space, i.e. a given weighting W, appears "uphill," or better than another. Fortunately, we can use the automated ROUGE evaluation method to evaluate the weighting/state given the set of responses which it produces.

We begin the process with a manually estimated weighting W, which assigns a scalar weight w_f to each feature f in our feature set F. We proceed in our hill climbing by iteratively generating and scoring a set of responses at the current weight W, and then considering a move to a new weighting W'. We create W' by selecting at random a feature f and incrementing or decrementing[2] w_f. We move from W to W' if it is an "uphill" or

[2] To bound the problem, we consider only the set of possible weights in $\{0, 1, 2, 4, 8\}$. Note that as described in Table 4.1, our features are each normalized so that their unweighted values fall between zero

"sideways" move, which may require generating and scoring the set of responses score for W' if it has not yet been visited. Once we reach a local maximum, i.e. a state/weighting from which we can only travel "downhill", we can either return that weighting or randomly restart the process. In practice, our stopping criteria for random restarts is partly bound by CPU time considerations (we run some hill climbs for several days) but also based on the number of restarts done without finding a better local maximum (we return if 5 consecutive restarts do not find improvement). While this procedure is not guaranteed to find an optimal weighting, it explores a large search space with relative efficiency.[3]

We run this algorithm four times for four different learning conditions. First of all, we separate the training using the DUC04 (biography) and DUC05 (topic-focused question) data. There are several motivations for this. First, we simply feel that since the two data sets are based around different questions, they may require different strategies. Moreover, the response length requirement in DUC04 is 100 words, whereas in DUC05 it is 250 words, which we believe may effect tradeoffs in coverage versus cohesion-focused features. Also, the Query feature is calculated somewhat differently for the two question types, as described earlier in Section 4.3.

For each of the two data sets/question types, we perform a two-step training process. The first training learns a weighting over non-RSR features only, and only adds in the RSR Coherence feature after training the other features. We train the non-RSR features first partly to establish a strong baseline so that any improvement achieved via these features will be in addition to an already-optimized score. Moreover, as described earlier in Section 4.3.2, we actually consider a number of different ways for calculating the RSR Coherence feature's value, and optimizing these RSR component weights at the same time as the five non-RSR features makes our search space overly large.

Thus, in the second training we use a fixed best weighting for the Non-RSR features found in the first training phase, and then conduct a second pass which holds those weight-

and one. Table 4.3 shows the final weightings learned.

[3] This description is purposely high-level, and elides some smaller details of the implementation. For instance, we require differences in ROUGE score to reach a certain magnitude to be considered "downhill" or "uphill" moves, i.e. minute differences are simply considered "sideways" moves.

ings constant while varying the RSR Coherence feature and component weights. While learning the weightings in this sequence makes the problem more tractable, we do acknowledge that it inhibits our ability to observe and adjust for dependencies between non-RSR and RSR features.

Setting	**Centroid**	**Coverage**	**Lexical Cohesion**	**Query**	**Document Cohesion**	**RSR Cohesion**
DUC05 best	2	2	1	4	4	2
DUC05 median	0	2	0	1	0	-
DUC05 baseline	1	1	1	1	1	1
DUC04 best	2	2	0	1	1	2
DUC04 median	2	4	0	2	2	-
DUC04 baseline	1	1	1	1	1	1

Table 4.3: Weightings found by hill-climbing experiments described in Section 4.4.2. Since the RSR Coherence weight is learned on top of the "best" settings for the other features, there is no RSR weight associated with the "median" hill-climbing setting. Feature descriptions are given in Table 4.1.

Thus, we altogether have four training settings. We begin with the two separate trainings over the DUC04 and DUC05 data, learning weightings for the Non-RSR features. Then for each of these, we train weightings for the RSR Coherence feature. Table 4.3 summarizes the learned weightings in each setting. While the hill-climbing algorithm cannot claim to find optimal values, we do note that we visit a substantial number of weightings in each case and thus can claim to have found non-trivial local optima. For instance, in the DUC05 non-RSR feature optimization, we try 391 weightings, finding 8 local maxima in the process. In all training settings, we observe that the difference between the lowest and highest ROUGE-SU4 recall scores achieved is significant using ROUGE confidence intervals ($P < .05$). We discuss in the next section whether these improvements carry over to the test set; we note them here only by way of saying that the different settings we explore do appear to have significant effect on content selection quality as measured with ROUGE.

We also observe that the learned weightings show some intuitively interesting differences in the best settings found for DUC04 versus DUC05 to be of interest. For instance, the

Query feature appears more significant for the broad queries of DUC05 than for DUC04, where it provides more fine-grained information about relevance.

Another difference is that Centroid and Coverage features dominate the DUC04 best settings. We theorize this to be due also to shorter summary length, since in shorter summaries, the Lexical/DocumentCohesion features can cause too much summary content to be devoted to a relatively narrow sub-topic. In a limited experiment with longer-length summaries for DUC04 document sets, we indeed found that these coherence-based features appeared to be more useful.

We do not show in Table 4.3 the component weights learned in our second training pass for computing the RSR Coherence, but rather only the overall feature weight (which has the same value, 2, for both DUC04 and DUC05 best settings). The reason for this is that, unexpectedly, in both settings our learning quickly finds that a setting which weights the component models of Adjacent, Contrast and Cause equally (at any positive weight) performs best for both the DUC04 and DUC05 data sets. The only difference is in the function used to combine the component model scores; the DUC04 training set performs slightly better when we take the feature value as the maximum of these component scores, and the DUC05 set slightly better when we take the mean.

This result is somewhat surprising, given that a primary reason for implementing the Adjacent RSR model in the first place was our intuition that it would be more valuable than the more semantically specific Cause or Contrast model for the task of assessing inter-sentence coherence in this context. There are a number of possible reasons for this. However, most fundamentally, we note that the detection of Adjacent-ness versus NoRelSame achieved substantially lower scores in the classification experiments as reported in Chapter 3, so the lack of differentiability between Adjacent sentence pairs versus same-topic pairs may be the critical issue. Nonetheless, we can intuit that the Adjacent component contributes positively and similarly as the Contrast and Cause models, since our learning assigns it equal weight as those models in the overall computation.

4.4.3 Results

Tables 4.5 and 4.4 show the results for our content selection experiments over the DUC04 and DUC05 sets. For the non-RSR settings, the RSR Cohesion feature is effectively turned "off" by being given a weight of zero. We also show in the table selected points from the parameter search over the non-RSR settings carried out by our hill-climbing algorithm, namely the best and median settings visited, as ranked by ROUGE score. We also show the score which results from adding the RSR Coherence score (calculated with its best setting) to the best combination of non-RSR features. In addition the table lists the scores for an evenly-weighted baseline (without the RSR Coherence feature), as well as the scores for the official test run submitted in the given year's DUC competition for our original DefScriber system and the best-scoring peer system (in DUC04, our system is the best-performing system).

Additionally, note that we provide the score of each setting on the training as well as test topics; this is primarily so that we can see whether overfitting appears to be a critical issue (i.e., whether our learned "best" weightings are too-well adapted to the training topics). It appears here that this is not the case, i.e. that the difference in our scores between training and test sets is broadly consistent. Further, we can see that the differences between training and test scores using our best setting correlates with differences using our baseline setting, indicating that the difference may primarily due to differences in average topic difficulty between the two sets[4].

In the non-RSR experiments over the DUC04 data, our best learned setting outperforms our original system and the baseline on the test set ($P < 0.05$). By adding the RSR Coherence feature to the best hill-climbing setting learned by the other, non-RSR features, we generally find small but positive increases on ROUGE scores; while the difference with the non-RSR score is not statistically significant, we increase our margin over the baseline and original results.

For the DUC05 data, our best setting with non-RSR data outperforms both the baseline and our original system's scores on the test set according to ROUGE confidence intervals

[4] See, e.g., Dang's overview (Dang, 2005) for further support for the notion that topic difficulty has a significant measurable effect on ROUGE score.

Weighting	**Score (Training)**	**Score (Test)**
Best hill-climb+RSR	0.1282	**0.1273**
Best hill-climb (non-RSR)	0.1233	0.1268
Median hill-climb (non-RSR)	0.1167	0.1165
Baseline (non-RSR)	0.1152	0.1223
DefScriber original / DUC04 best peer	0.1177	0.1209

Table 4.4: DUC 2004 data set performance measured by ROUGE SU-4 recall scores using the feature weightings learned in experiments described in Section 4.4.2. The settings correspond to the learned weights shown in Table 4.3.

Weighting	**Score (Training)**	**Score (Test)**
Best hill-climb+RSR	0.1362	0.1326
Best hill-climb (non-RSR)	0.1351	0.1300
Median hill-climb (non-RSR)	0.1325	0.1285
Baseline (non-RSR)	0.1324	0.1260
DefScriber original	0.1285	0.1236
DUC05 best peer system	0.1290	**0.1337**

Table 4.5: DUC 2005 data set performance measured by ROUGE SU-4 recall scores using the feature weightings learned in experiments described in Section 4.4.2. The settings correspond to the learned weights shown in Table 4.3.

($P < 0.05$). Given that our system had the fourth-highest ROUGE-SU4 Recall score among the 32 scored systems in DUC05, this significant improvement is non-trivial. Adding the RSR Coherence feature improves the score further, although not significantly so. However, the boost in score from adding the RSR Coherence feature does move us closer to the best peer score, to the point where there is no significant difference between our scores using ROUGE confidence intervals ($P < 0.05$).

We list an encouraging example of the type of improvement we can achieve in the biographical response example in Figure 4.1, which shows a response to “Who is Stephen Hawking?” There are two critical improvements here. First, the sentence about “Mr. Big Bang Theory” moves upward in the summary to be the second sentence (after the initial Genus-Species sentence). We find in our analysis that this sentence is boosted up mainly

Manually Estimated Weights, No RSRs	Trained Weights and RSRs
Hawking holds the Cambridge University post once held by Sir Isaac Newton and is the author of “A Brief History of Time.” I’m not sure if that’s what the devoted followers of Dr. Stephen Hawking call themselves, but they should. Stephen Hawking remains confident that physicists will prove string theory – a so-called “theory of everything” to explain the universe – but said today it might take longer than he had expected. Hawking, Mr. Big Bang Theory, has devoted his life to solving the mystery of how the universe started and where it’s headed. Stephen Hawking, hailed as a modern-day ...	Hawking, 56, is the Lucasian Professor of Mathematics at Cambridge, a post once held by Sir Isaac Newton. Hawking, Mr. Big Bang Theory, has devoted his life to solving the mystery of how the universe started and where it’s headed. Stephen Hawking remains confident that physicists will prove string theory – a so-called “theory of everything” to explain the universe – but said today it might take longer than he had expected. Hawking’s odds on proving the theory are the same, but he now says it could take another 20 years. Stephen Hawking, the Cambridge theoretical physicist who wrote “A ...

Figure 4.1: Example answers produced by DefScriber before and after training and addition of RSRs for the biographical question, “Who is Steven Hawking?” The last sentence in both answers is truncated as it would be in the official DUC submission to comply with the 100 word limit.

because of its high Centroid score and the high weight learned for the Centroid feature in the DUC04 training. The second key change involves the two sentences about proving string theory, which are placed together even though they occur in an original document separated by three intervening sentences. In analyzing the data we find that all of our Cohesion features play a role in promoting this sentence; it has good Document and Lexical Cohesion, but it also scores highly according to our RSR Cohesion model; in particular, the terms *today* and *years* are strongly linked in the Contrast model, while *proving* and *theory* are strongly linked by Cause. Lastly, note that the effect of implementing the beam search is visible in the change in initial Genus-Species sentence. This occurs because the selection algorithm finds that it reaches a higher total score by using the second GS sentence at the start of the response, and deferring the mention of a sentence from the cluster which essentially groups together “A Brief History of Time” mentions until later in the summary (this is the last sentence in the second summary, which is truncated).

4.5 Content Ordering Experiments

In addition to content selection, we are also interested in measuring the effect of our general updates and RSR-based features in particular on the quality of DefScriber’s content ordering. In this section, we explain our implementation of an additional post-processing algorithm which reorders the sentences in a response in order to increase cohesion, and then describe an experiment to measure the content ordering performance of DefScriber both before and after this algorithm is applied.

4.5.1 Reordering Responses

In manual examinations of some output responses over our development topics, we observe that even when good sentences are selected, the ordering of those sentences which results from the iterative calls to ChooseNextSentence is not always ideal. We thus decided to implement a post-processing sentence reordering algorithm which aims to improve cohesion and make our response more coherent overall.

To this end, we implemented an algorithm that examines the multi-sentence answer produced using ChooseNextSentence(), and looks for opportunities to reorder which improve local cohesion of the response as a whole, measured by the sum of Document, Lexical and RSR Cohesion features of adjacent sentences, features which can be recalculated on the fly for any proposed reordering.

Of course, these features are already considered within ChooseNextSentence itself, when the original sentence ordering is produced. Yet to understand why we consider them again in a post-sentence-selection pass involves considering how these cohesion features are used within the ChooseNextSentence algorithm. In ChooseNextSentence, i.e. when response content is being chosen, coherence features are weighted with respect to other features which consider other aspects of content “goodness,” e.g. the Centroid feature, which measures topic-level centrality. Determining the proper weighting for these different features is the subject of our hill-climbing experiments in the previous section. In some sense, within the ChooseNextSentence content selection phase, the role of cohesion features is to inform the system about what sentences are "close" to sentences which are already judged "good," on the intuition that this can itself be a clue to content importance.

However, by implementing a reordering step after the response content has already been selected, these features can be used used purely to determine if, within the selected sentences, we can achieve a better ordering. By increasing cohesion locally, the goal is to make the summary more fluent and also to increase its overall comprehensibility.

As we discuss later, this begs the question of whether and how Cohesion features should be used in ChooseNextSentence in the first place. However, we implement this separate post-selection reordering as an interim step which does not require us to rework the way these features are used in ChooseNextSentence at this point, and defer more fundamental separations of the content selection and ordering steps.

4.5.2 The Reordering Algorithm

The reordering algorithm takes as input the N sentences selected by ChooseNextSentence in the order of their selection, which is equivalent to the final summary produced by the system, modulo possible truncation of the final sentence to conform to DUC word-length limit. The algorithm then proceeds to greedily swap the sentence pair which yields the highest resulting Cohesion measures of the response as a whole, as long as:

- Overall cohesion of the newly adjacent sentences is increased. This overall measure combines Document, Lexical and RSR Cohesion features. Note that these features are considered separately in the reordering algorithm than in sentence selection, i.e. they can be weighted differently for reordering than for in sentence selection.

- Various ordering heuristics are maintained. For example, Genus-Species sentences cannot be moved out of the first position.

Again, we use a beam search (beam width $n = 4$) to counteract the possibility of local maxima in this greedy algorithm.

4.5.3 Experiment and Data

In order to verify that this reordering algorithm was indeed improving our output summary order, we carried out the following experiment to separate the task of content selection from content ordering: We scrambled sentences in reference summaries, and left our system the task of recreating the original, human-selected order.

In order to do this, we prepared four augmented versions of each training set from the DUC04 and DUC05 data sets, each containing an (different) additional pseudo-document which is actually a model summary for that set. Then, we implemented a constrained mode of DefScriber in which only sentences from this pseudo-document can be included in the final answer, but without using any information about the original order of its sentences. This allows us to simulate the kind of background information which DefScriber would typically have, i.e. a cluster of documents which relate to a given query, while seeing how it uses this information to recreate the ordering of a human-quality summary of this set. This is distinct from the task of merely trying to infer the ordering of an arbitrary text document, since we do not have the question and background document set in that case. Before running DefScriber in this mode, we must also disable the Document Cohesion feature for these experiments, since all source sentences come from the same "document", so following within-document ordering information captured by Document Cohesion would make the task trivial. Thus, we rely on Lexical Cohesion and RSR Cohesion features to make reordering decisions.

Using this experimental setup, our goal is to compare the outputs from the system with and without the reordering algorithm, and when using the reordering post-process to measure the specific contribution of RSR Coherence to its performance.

4.5.4 Training

We conduct another hill-climbing experiment, where in this case our objective function for determining the value of a given weighting is the Kendall's Tau score. In addition to having been used in other document-ordering evaluations (Lapata, 2003; Barzilay and Lee, 2004), a desirable property of this measure is its ease of interpretation. Values range from -1 to 1, reflecting inverse to perfect ordering, respectively. Interpretively, one can view a score as

Threatened mammals include golden lion tamarins in Brazil; African elephants poached for their ivory tusks; black and white rhinos hunted for their horns; Indian tigers hunted to meet the demand for tiger bone; jaguars and black Howler monkeys in Belize; blue and humpback whales; Arabian Oryxes; black-footed ferrets; tigers in Sumatra; bison in Poland; wild dogs in southern Africa; African manatees; giant Pandas; lemurs in Madagascar; harp and hooded seals; bears in the Pyrenees; and orangutans in Indonesia. (1,2)

Wildlife in danger of extinction includes species of mammals, birds, reptiles, and amphibians.(2,1)

Threatened birds include red kites, British barn owls, and North American whooping cranes; threatened reptiles include sea turtles in waters from the Caribbean to the Atlantic and from the Mediterranean to the Pacific; and threatened amphibians include South American caimans and Majorcan toads.(7,3)

Programs seeking to protect wildlife from extinction can be grouped together according to the nature of the methods they use.(3,4)

Another group deals with breeding endangered species, such as whooping cranes and tamarins, in captivity.(4,7)

Another group of programs is concerned with establishing conservation areas, such as the preserve in Belize for jaguars and monkeys and the whale sanctuary around Antarctica where hunting is prohibited.(5,6)

One group of programs is concerned with banning or restricting activities, such as trading in ivory, poaching of rhinos, commercial whaling, hunting turtles, and importing baby sealskins; and with punishing those engaged in such activities, such as embargoing imports from countries failing to stem trade in rhino and tiger parts.(6,5)

Figure 4.2: DefScriber's ordering of sentences from a human-written response for the question, "Wildlife in Danger of Extinction: What general categories of wildlife are in danger of extinction world-wide and what is the nature of programs for their protection?" The ordering is performed as part of the experiments in Section 4.5. Numbers after each sentence indicate DefScriber's position for the sentence before invoking reordering, and original position in the human-written response, respectively. So the first sentence above, marked (1,2), was also the first sentence in DefScriber before reordering, the second sentence in the original human-written response.

proportional to the probability that a given pair of sentences within a summary is ordered as in the original model; a score of 0, for instance, reflects an even probability that any pair will be in correct versus inverse order; 0.5 means that 75 percent of the sentence pairs are correctly ordered with respect to the model. It is calculated as:

$$\tau = 1 - \frac{2(\text{number of inversions})}{N(N-1)/2}$$

Where N is the number of items to be ordered. In our training, we take the Tau score for a given weighting as the mean Tau score across all of the constrained summaries produced from the human-abstract-augmented training document clusters.

We use this hill-climbing experiment to estimate the weights used in reordering when combining the Lexical and RSR Coherence features. While we could attempt to manually estimate the weights used for reordering between Lexical and RSR Cohesion features, the RSR probabilities returned by the different models for Cause, Contrast and Adjacent can be difficult to interpret and combined in different ways. Thus, in this experiment, we again use an automated hill-climbing to set these weights, adjusting feature parameters with the goal of improving the Kendall's Tau score. We do not need to separately estimate weightings for the non-RSR baseline, since the Lexical Cohesion feature is the only feature used in that case and thus does not need to be combined with anything else.

4.5.5 Results

Setting	**Kendall's Tau (Train)**	**Kendall's Tau (Test)**
DUC05 Lex+RSR	0.084	**0.087**
DUC05 Lex Only	0.075	0.079
DUC05 No Reorder	0.066	0.069
DUC04 Lex+RSR	0.267	**0.259**
DUC04 Lex Only	0.238	0.242
DUC04 No Reorder	0.230	0.228

Table 4.6: Mean Kendall's Tau scores using Lexical and RSR Cohesion, Lexical Cohesion only, and no reordering.

Table 4.6 shows the results of these experiments with post-selection reordering. We include the Tau scores for both the non-reordered baseline, as well as the Lexical Cohesion only baseline, i.e. the reordering performance without the RSR Cohesion feature. We observe that in all cases, the addition of the reordering step using Lexical Cohesion only improves scores, and that scores further improve with the addition of the RSR Cohesion feature. However, using a one-way ANOVA for correlated samples, the overall improvement for the DUC05 test set (i.e. the improvement from no reordering to reordering with Lexical and RSR Cohesion) approaches but does not reach statistical significance ($P < 0.09$). The improvement for DUC04 test set is not statistically significant either ($P < 0.12$).

Additionally, we note that in the DUC05 case, the Kendall's Tau scores do not rise above 0.1, reflecting that we have difficulty recreating the model summary orderings at a rate much above random. On the other hand, the scores are significantly higher in the DUC04 case. We theorize that, at least in part, this may be due to the fact that in biographical summaries such as that of Figure 4.1, we describe a single entity and tend to follow a clearer more-to-less importance ordering. In this case, the high value of the learned Centroid weighting for biographical (DUC04) responses may account for the relatively high Kendall's Tau value even before any reordering is done. Moreover, we may benefit from the likelihood of having a Genus-Species sentence in the first position of biographical responses.

However, another issue is that DUC05's longer (in terms of word length) responses may have more complex structure. That is, the original version of the human abstract which is shown in its reordered version in Figure 4.2 (the human-written ordering can be recovered using the second set of numbers after each sentence in the Table) follows a kind of statement-example schema, making general statements in sentences 1 and 4 (about kinds of endangered wildlife and kinds of programs to help these species), and following each with several examples in sentences 2-3 and 5-7, respectively. Our approach of maximizing local cohesion measures cannot fully recover an ordering of this kind.

A final point here is that because of the limitations of our experimental setup we are unable to use the Document Cohesion feature to help our ordering decisions (since it would render the experiment trivial, by giving us the initial ordering of sentences from the human abstract "pseudo-document"). While it is difficult to estimate exactly what effect this feature will have outside of the ordering experiment, our learning experiments in the previous section for both the DUC04 and DUC05 data find that it contributes to identifying useful content. Outside of the ordering experiment, we speculate based on anecdotal evidence that leaving this feature “on” (using a manually estimated weight) during the post-processing reordering can help us to some degree in finding good orderings, since it takes advantage of the implicit information that human authors give us by ordering two sentences in a certain way in the same original document.

4.6 DUC 2006 Result Overview

In the process of completing the above-mentioned experiments, we participated in DUC 2006, where the task was to produce topic-focused responses to questions formatted as in DUC 2005. As our findings from this chapter were not yet complete at the time we produced our official DUC06 run, our system had not yet been fully optimized and refined. Nonetheless, the system as run for DUC06 included early versions of the main updates mentioned in the previous sections, including the reordering algorithm and best learned weights for the non-RSR features. In addition, we used an early implementation of the RSR feature, but since the RSR-specific experiments were not complete at that time our use of the RSR Coherence feature parameters was not entirely optimal.

As we focused our main effort on the above-mentioned experiments, which were predicated on the data sets for DUC04 and DUC05 available at that time, we have not yet performed an extensive analysis of the DUC06 results. However, our preliminary examination shows our system performing above the median in all of the evaluated scoring categories, including manual and automatic measures. In particular, among 35 system peers evaluated, our scores were in the top 10 for ROUGE recall, mean linguistic quality, and mean responsiveness.

Figure 4.3 shows an example answer from our submitted response. Note that the first sentence selected is a Genus-Species sentence; the sentence which follows it is adjacent to it in the original document, but its selection is influenced here by the a high RSR-based score based on our Cause model as well. Also, note that we insert paragraph breaks in

Title Global Warming

Question Describe theories concerning the causes and effects of global warming and arguments against these theories.

Answer Global warming is the change in the climate thought to occur because human activities add to a buildup of greenhouse gases such as carbon dioxide, methane and nitrous oxide. Those gases trap heat, and although there is ongoing dispute about how that in turn affects the climate, there is no doubt that global temperatures are rising.
The scientists said that the warming effect, especially rising temperatures in winter, will result in a series of changes to nature and human society.
The findings suggest that global warming blamed on human industry is manifesting itself not just by causing a global rise in temperatures but by also favoring certain established weather patterns.
Two big questions complicate efforts to predict the course of the earth's climate over the next century: How sensitive is the climate system, inherently, to the warming effect of greenhouse gases?
The cloud – a harbinger of climate change, some call it – appears as global warming research is becoming a major scientific industry in fields ranging from atmospheric physics to biology.
Over the years, skeptics have tried to cast doubt on the idea of global warming by noting that measurements taken by earth satellites since 1979 have found little or no temperature rise in large parts of the upper atmosphere. The satellites' all-encompassing coverage yields more reliable results than temperature samplings showing a century-long warming trend at the earth's surface, they argued.
Sea levels would rise as the Antarctic ice sheet melted because of higher temperatures caused by global ...

Figure 4.3: An example DUC 2006 topic (d641e) and DefScriber's answer.

our summary between most sentences, but not between those with high cohesion (in this case, the first and second, as well as the seventh and eighth). However, remember that in this work we do not explicitly recognize that the question in this example is itself causal in nature; rather, we are using our Cause model only to help assess cohesion between sentences which we select for the answer based on other criteria. (In the next chapter, we explore how we can use our Cause model to identify relevant sentences directly by their relationship to explicitly causal questions.)

Based on this initial analysis, we observe that our system remains a robust and competitive one in this task. While our new research did not vault us to the top of the rankings, we observe that many other participants appear to have been hard at work as well; for instance, our ROUGE scores in 2006 would have been significantly "winners" in 2005.

4.7 Conclusions and Future Work

In this chapter, we begin with the goal of adding a Rhetorical-Semantic Relation-derived cohesion feature to DefScriber. In the process, we also make important extensions to DefScriber which allow us to integrate new features more efficiently and empirically, using a supervised learning approach to take advantage of the training data which has become available through the last few years' Document Understanding Conferences (DUCs). Overall, we are pleased to make gains in two aspects of system performance, content selection and content ordering.

In our first set of experiments, we evaluate content selection using the ROUGE metric. We find that by implementing a new, more easily trainable framework for sentence selection, we are able to significantly outperform our previous results on a test set of biographical (DUC 2004) questions and a second test set of topic-focused (DUC 2005) questions. In both cases, the addition of an RSR Cohesion feature improves ROUGE scores, although in neither case is the improvement over peak non-RSR performance statistically significant.

In a second set of experiments, we evaluate content ordering by asking our system to recreate the ordering of a human-authored abstract. We find that the addition of RSR Cohesion measures increases performance in this task as well, but the increase does not reach the level of statistical significance. However, even with the addition of a reordering step which reviews the final answer globally for opportunities to increase local cohesion, we find that our performance, as measured by the Kendall's Tau statistic, is relatively low. Particularly in the case of the DUC 2005 topic-focused questions, we find the order-inducing task challenging, which we hypothesize has to do with the nature of those summaries themselves as well as limitations in our system.

Perhaps the broadest limitation involves the current overlap in DefScriber between methods for content selection and content ordering. Currently, the initial response generation phase which selects content (via the ChooseNextSentence algorithm) uses features which consider both global importance, like Centroid, and local cohesion, like Lexical Cohesion. Then, once all sentences are selected, we look globally for opportunities to increase average local cohesion using a post-selection reordering algorithm, making a series of decisions based on local cohesion measures alone. We believe that future work should consider how to make these pieces fit together more intelligently.

While we advance DefScriber's sentence selection process in this chapter to include a unified feature representation and a straightforward approach for training a linear feature combination, we would like to experiment with other frameworks or learning methods.

In terms of framework, a key issue involves the large granularity of our training examples, which currently take as an objective function scores based on the entire response produced by a given feature weighting. Instead, a finer-grained approach might separate the problems of: (1) estimating intrinsic sentence relevance, e.g. by individual sentence ROUGE score, (2) estimating the strength of combining pairs or larger groups of sentences and (3) re-estimating intrinsic sentence relevance given a partial summary. Two recent papers report success in (1) (Daumé III and Marcu, 2005; Fisher and Roark, 2006); Nenkova focuses on analogous issues to (2) and (3) in her work on general-purpose (as opposed to topic-focused) multi-document summarization (Nenkova, 2006).

In terms of learning methods, those which consider more complex models could be applied to consider more nuanced analysis of feature interaction and effect. For instance, Estimation Maximization (Dempster et al., 1977) can be used to learn a Gaussian mixture model for weighting, and rank-based methods like perceptron ranking (Crammer and Singer, 2001) are advocated by Fisher and Roark (Fisher and Roark, 2006) to avoid overfitting on training data; nonetheless, other competitive systems have reported success with methods similar to our hill-climbing approach (Daumé III and Marcu, 2005). Moreover, sequence-aware models which consider ordering beyond the local, sentence-pair level should be considered when learning sentence combination/ordering rules. These can help address non-local coherence issues of the type we find in our reordering example in Figure 4.2.

While we point out in that example that a tree-like structure pertains, even methods which do not fully induce such a structure can still be helpful by considering new features, e.g. average cohesion over larger sentence groups within a potential response. A general issue to keep in mind, however, is that the adaptability, robustness and intuitiveness of DefScriber's approach have often been an asset in adapting quickly to tasks where training data is sparse or the domain of potential questions broad. Whatever future changes we make to the system, especially those which aim to take advantage of a supervised training setting, should be mindful of this and try to make any such improvements in such a way that they can be useful even when this kind of training data is not available.

Another issue with respect to any learning we attempt to perform is the choice of metrics which we use as the objective function in our experiments. In our content selection experiments, we use ROUGE (Lin, 2004) to evaluate the content of our responses with respect a set of human-created abstracts. In our content ordering experiments, we use Kendall's Tau in a synthesized task which aims to recreate the ordering of some of those same abstracts. While ROUGE has been shown to correlate with human judgments of summary quality, and Kendall's Tau has a straightforward intuitive justification, the key reason we use these metrics is our ability to train automatically where the creation of human judgments would be expensive and create its own ambiguities. While we feel that these metrics are entirely reasonable for studying our problem with the data we have, we feel that it is important to keep in mind their limitations with respect to representing "real" ground truth.

While there is no "magic bullet" alternative, one possibility would be to attempt a semi-automated scoring of responses using the Pyramid metric (Nenkova and Passonneau, 2004), in order to determine if that metric's use of human-annotated semantic chunks helps us better understand system performance in content selection. While automating the Pyramid scoring process completely is difficult, it might be practical for use as a post-hoc procedure to compare best/median/baseline settings which are first determined by ROUGE.

With respect to the content-ordering experiments, rather than necessarily changing the Kendall's Tau metric, we might instead experiment with other tasks beyond our synthesized order-recreating task. One possibility here would be to extract the available human quality judgments for each peer answer on the "coherence/structure" quality question which is asked in DUC. Then, for experimentally-generated orderings, we could evaluate their coherence based on how the same or a similar response had been rated by a human. However, even judgments for each peer systems' answer (20 to 30 systems have participated over the last few years), this data is likely to be relatively sparse given the vast possible set of answers which can be created, although this is somewhat tempered by the fact that most systems use extractive techniques. Nonetheless, it would require a method to adjust scores based on an estimate of response similarity, which brings us back to a problematic issue.

These ideas address issues for improvement in our general approach, but what of the fact that adding RSRs only improves our results modestly, despite all of our efforts to build TextRels? First, we feel that some of the already-mentioned ideas for future work, including an architecture which separates content selection and ordering decisions more cleanly, could help the RSR Cohesion feature have a stronger effect. Moreover, representational variations may be useful, such as using the singular value decomposition (SVD) approach used in LSA (Deerwester et al., 1990) to compress and abstract our model dimensions. Evaluating LSA itself as a comparative/complementary cohesion feature would also be worth pursuing, although as we mention in the related work section, other studies have not found ways to make LSA consistently effective in this role.

Another issue concerns whether using our RSR models to assess cohesion is even the best way to apply them in these long answer QA tasks. In this chapter, we use them to assess sentence-pair cohesion, a way that is perhaps most natural given that we have already implemented classifiers to do this. However, as we explain in the next chapter, we can also find alternative ways to use the knowledge captured by our RSR models, for instance in a lexical expansion role to help assess relational relevance between question and answer.

Finally, another important factor which may be limiting our performance in DefScriber is the lack of an approach for sentence simplification or compression. In recent years, researchers have identified capable methods for simplifying long sentences (Siddharthan, 2003; Galley and McKeown, 2007) while maintaining important syntactic and semantic structure. Others have reported that using these kinds of compression techniques can improve summarization results (Siddharthan et al., 2004; Daumé III and Marcu, 2005), including in the most recent DUC where the top-performing system on a number of benchmarks makes use of sentence compression (Vanderwende et al., 2006). While the performance of DefScriber is still strong among its peers, the time has probably come for us to integrate a sentence compression component in order to remain competitive.

Chapter 5

Applying RSRs for Relation-Focused Questions

5.1 Introduction

We initially pursue the problem of RSR modeling in Chapter 3 motivated by the desire for a robust, automatic technique that can improve the quality of DefScriber's descriptive answers using rhetorical and semantic concepts. And in Chapter 4, we find that the ordering and content of these answers can be improved to a degree using our TextRels RSR models. Yet we believe that the relation knowledge captured in these models can also be leveraged more directly for answering a class of questions which explicitly call for explanations and comparisons.

Looking back at our experiments in Chapter 4, we use RSRs to help create descriptive answers responding to queries for information on a broadly stated topic of interest. In such cases, the relation between question and answer is one of general relevance, i.e. inasmuch as the answer should be relevant to the question, but beyond that it is not clearly defined. In that context, we integrate RSRs into our algorithm for sentence selection by including features which take account of, e.g., how "Cause-like" a given candidate sentence appears to be with respect to the previously chosen sentence. That approach uses the RSR models as an heuristic for a good arrangement of sentences, on the intuition that answer quality will improve if we arrange our sentences so as to be tightly linked according to our various

RSRs.

That is certainly a logical way to use our RSR models, i.e. to gauge how close a given arrangement of sentences is to reflecting the discourse phenomena on which we train. It essentially follows the functionality which we evaluate in Chapter 3, where we classify the relation of two text spans which occur together in a document.

Yet even while our RSR models are trained and evaluated in a framework which focuses on the relationship of text spans which occur in the same document, their usefulness can extend beyond attempts to mimic such a context. Instead of using TextRels to *discriminate* between relational strength across two input text spans which may be placed together in a document, we explore in this chapter an approach where we use the models to *generate* relevant terms for a particular question. Using this new approach, we can seek out relevant text more efficiently, and gain flexibility in using the models outside of a strict classification paradigm. From both a development and user perspective, the generated terms provide an intuitive window for understanding the kind of knowledge which is acquired by the model. This is not as clearly the case in our classification-based approach, where the result of consulting our model is a probability of a given relation across a sentence pair, and decomposing that probability to come to an intuitive understanding is a non-trivial task.

We analyze the contribution of this method in the context of answering questions which address the Cause and Contrast RSRs directly. These "relation-focused" questions call directly for explanations (e.g., "Describe causes of the espionage conviction of Edmond Pope.") or comparisons (e.g., "Compare Korea and France with respect to economic outlook."). Because these questions explicitly address relational concepts of Cause and Contrast, we can use our RSR models to help assess the intrinsic relevance of a potential answer sentence with respect to a question. This is distinct from the scenario in the previous chapter, where the intrinsic relevance of a candidate answer is based only on its textual overlap with a question. In that setup, RSR models are used only after the set of question-relevant sentences is identified, to assess the strength of links between candidate answer sentences already determined to be intrinsically relevant.

For instance, consider the problem of answering the question, "Describe the causes of Taiwanese Premier Tang Fei's resignation." Assuming we have a means for identifying *res-*

ignation as the question's focus, a classification-style approach might amount to classifying possible answer sentences with regard to how Cause-like they are with respect to the term *resignation*. However, this is both inefficient in terms of the number of classifications done, and, depending on the number of documents we are willing to sift through, may also miss key sentences.

Instead, we would like to be able to directly search for those sentences which include terms for which we have strong evidence that they are causes of *resignation*. As we explain in this chapter, we can consult our Cause model and find out that, for instance, terms like *illness, fail, bribery, pressure, harassment, conflict, dispute* etc. are such likely causes. We can then use these terms to direct an efficient search for sentences containing these terms (emphasized in bold) e.g.:

> After months of **failing** health, Prime Minister Tang Fei, resigned Tuesday amid mounting **pressure**.

Causally-linked terms can be used in this way to find question-relevant sentences in documents retrieved using keyword queries which concatenate question words. Furthermore, one can imagine that in some cases causally-related terms might help to find relevant information which predates the event in question. In this case, a sentence or document may not even mention *resignation*, but can still be inferred to be relevant:

> Tang has been in the job for less than five months, but his tenure thus far has been uninspired, marked by **illness** and frequent **disagreements** with President Chen Shui-bian.

Likewise, when answering a comparative question like "Compare Korea and France with respect to economic outlook.", we can consult our Contrast model to identify language commonly used when making comparisons in the domain of *economic outlook*, and focus our search for sentences containing relevant comparative terms like *stagnant, trend, term, caution*, etc.

Questions with a causal or comparative dimension represent an area only sparsely explored in previous work, and an area whose place in the QA community is still evolving. For example, the consensus opinion in the ongoing experimental Information Distillation task being conducted under the DARPA GALE program[1], has been to omit causal questions to this point because of their perceived difficulty. While our exposition therefore acknowledges the broad scope of these new question types and the associated vast search space of potential problems and solutions, we persevere in defining a subset of problems which we can concretely explore. By discussing our findings in implementing and evaluating a system for answering relation-focused questions, we hope to provide useful initial results in this developing research area.

As a whole, this chapter addresses the third research question which we address in the introduction to this thesis, namely: How can we apply this rhetorical-semantic model to enhance the performance and scope of our question answering techniques? In particular, we focus here on expanding the scope of our capabilities, by exploring this new area of causal and comparative questions.

The remainder of the chapter is organized as follows: We begin with a review of related work in QA, but also in other areas of NLP such as text generation (Section 5.2). We then describe the notion of relation-focused questions, broadly situating the problem, attendant challenges and research areas (Section 5.3) before narrowing our focus to concentrate on two specific kinds of relation-focused questions, Explanations and Comparisons (Section 5.4). In Sections 5.5 and 5.6, we describe our implementation of CCQ, a system for answering these questions which uses relevant-term generation from our RSR models to identify answer material, and reuses several DefScriber modules to assemble relevant information into a multi-sentence answer. We evaluate CCQ's performance via a user study, and find that adding RSR-derived information achieves the top results in all cases, and significantly improves performance in some scenarios (Section 5.7).

5.2 Related Work

Work which explicitly focuses on questions with rhetorical or semantic "relation focus" is fairly sparse. However, if we take a slightly broader view, we find related threads not

[1] An overview of the GALE program is available at: `http://projects.ldc.upenn.edu/gale/`)

only in QA research, but also in work on information retrieval, text generation and lexical semantics.

While causal and comparative relation-focused questions *per se* have not been widely studied, broad-participation exercises such as TREC (Voorhees and Buckland, 2005) have included a small number of questions which relate to these concepts. In TREC, the number of explicit cause questions is small; only 12 of approximately 3,000 questions in the 2000 through 2005 question sets are "Why" questions (e.g., "Why is Jane Goodall famous?"); over the same span, there are perhaps five explicit comparison questions (e.g., asking "how much stronger" one material is compared with another). While a somewhat broader set of questions might be viewed as implicitly, or at least plausibly, causal or comparative[2], techniques which specifically address such questions from a relation perspective have gained little traction. Instead, the approach of many systems is to identify the expected type of answer from within a taxonomy of known types (e.g. the Webclopedia QA typology (Hovy et al., 2001)), often using learned rules to map a given question to its expected answer type, as in Li and Roth (Li and Roth, 2002). This type-centric method can be integrated with an information-retrieval component in the manner of "predictive annotation" (Prager et al., 2000), whereby potential answers are annotated in the corpus using the question typology, for more efficient question-answer matching at runtime.

These approaches can achieve high performance in finding factoid answers to certain cause- or contrast-type questions to the extent that similar questions are seen or hypothesized in advance. In this sense, the question "How fast is X?" is analyzed as "What is the SPEED of X?" and "How did X die?" as "What is the CAUSE-OF-DEATH of X?". On the answer-finding side, such a system relies on annotations which identify phrases denoting units of speed or fatal diseases. However, such techniques do not generalize well over unseen question types; that is, their method for matching a cause-of-death question to an answer tells them nothing about matching a question which asks about causes of, e.g., a dispute between countries or the resignation of a politician.

[2] For instance, from the approximately 1,200 questions in the 2001-2 sets, there are two "How did X die?" questions which could be interpreted as asking for causal explanations, and four "How fast is X?" as asking for a comparison of the speed of X with that of another thing.

In order to add a dimension of sentential semantics to address some shortcomings of the type-centered approach, Narayanan and Harabagiu (Narayanan and Harabagiu, 2004) use semantic frames to guide the search for answers beyond merely looking for a specific named-entity type, using frame representations in the style of PropBank (Palmer et al., 2005) to identify the roles that different question elements should have with respect to one another. This approach does avoid some of the pitfalls of an entity-centric approach, but is limited to some extent on the match between questions and available frames; nonetheless it is a potentially powerful complement to a completely entity-driven method.

Another thread of work which is closer to our ultimate approach involves the task of answer-containing passages (which can be sentences, paragraphs or arbitrary-length windows), rather than attempting to identify exact answers based on a strong type hierarchy. Tellex et al. (Tellex et al., 2003) compare various algorithms for passage retrieval, finding that techniques which use density of query terms as a primary measure tend to be successful. Tiedemann (Tiedemann, 2005) suggests a flexible way to extend the terms used in passage retrieval using linguistically-motivated techniques which rely on syntactic annotation of question terms to weight a corresponding query, and finding significant improvement over a test set based on the CLEF QA evaluation. In a broader context, Dumais (Dumais, 1996) demonstrates the efficacy of semantically motivated measures of relevance, implemented via Latent Semantic Analysis (LSA) techniques (Deerwester et al., 1990), to filter information retrieval results based on semantic "closeness" to specific topics of user interest. Whereas the LSA technique uses a global, static notion of semantic space derived offline over an arbitrarily large corpus (in principle, the larger the better), pseudo-relevance feedback (Xu and Croft, 1996) offers a dynamic method for finding related terms, but can suffer from so-called "query drift" inasmuch as a new query based on found terms can "drift" away from the original topic.

Work focusing on cause-specific semantics for QA is fairly sparse. Expanding on early work (Garcia, 1997) on causal analysis of action verbs, Girju performs a detailed manual analysis of causality in verb phrases (Girju, 2003). This analysis is used to build a classifier which determines whether causally-ambiguous verbs, e.g. *produce*, carry causal semantics in a given context[3]. While this approach is useful for finding explicit statements of causation, it does not build a concept-based model of causation. That is, Girju demonstrates how the model can be used to answer the question, "What are the effects of acid rain?", via causal analysis of the verb *contribute* in the candidate answer sentence, "Acid rain is known to *contribute* to the corrosion of metals.", rather than any model of a causal connection between *acid* and *corrosion.* Thus it is not effective for seeking our implicit causal statements like, "With the increase in acid rain, we saw substantial metal corrosion."

Taking a somewhat broader view, Bethard (Bethard, 2005) suggests an approach to answering causal and other questions which centers on relations between "events" rather than strictly verbs as in Girju. This approach has the benefit of abstracting the causal relation away from limitations of particular scope (i.e. lexical, propositional, sentential). Moreover, it can potentially use other kinds of event logic (e.g. about temporal or location properties of an event) to support causal reasoning. However, this work is still in the developing stages.

Verberne et al. describe collecting a set of short-answer "*why* questions" to augment the low coverage of such questions in the TREC and other collections (Verberne et al., 2006). Their approach involves manual elicitation of questions and answers from a set of annotators given a small document collection, namely the RST Bank corpus (Carlson et al., 2001). In companion work (Verberne, 2006), Verberne proposes a procedure for using syntactic criteria – some of which are similar to Girju's verb-centered analyses – classifying these questions into four subtypes: *Cause, Motivation, Circumstance* and *Purpose.* In recently submitted work (Verberne et al., 2007), they use this classification as part of an end-to-end approach for identifying answers from their set of *why* questions within the RST Bank corpus. This work uses the manually annotated RST Bank corpus (Carlson et al., 2001) to test a procedure for answering short-answer *why* questions, using an approach which matches a question topic lexically to text in an RST nucleus and selecting its satellite as an answer. While this approach is relatively successful when applied in the context of a small

[3] The training data they derive for a semi-supervised learning approach uses, in part, the "Cause-To" relation which is annotated in WordNet (Fellbaum, 1998); however, as they note, this relation is annotated only for a small number of verbs (approximately 500)

corpus with human-quality RST annotations, its applicability over a larger corpus with automatic annotations remains an open question. Another finding in their evaluation of this procedure is that, even with human-annotated RST features, approximately 20 percent of questions require what they call "world knowledge" to be answered. For instance, for the question "Why is cyclosporine dangerous?" they note that their system is unable to find an answer because it cannot causally link *dangerous* to phrases like *renal failure, morbidity, nausea*. This points out a clear possibility for the contribution of RSR models like ours, i.e. as a resource for finding these kinds of inferred relational links.

Overall, Verberne et al.'s work shares several important elements with our own, including a focus on causal questions and an approach which aims to utilize elements of discourse theory. However, we feel that their work is complementary with our own, in that (1) they focus on using RST relation *structure*, or form, to find statements of causal links in text, whereas our approach (detailed later in this chapter) uses RSR models to focus on relation *content* (2) their approach is firmly in the short-answer mold, and on a single type of question (*why*), whereas our work examines questions related to models of cause and contrast, and is in a long-answer framework.

Other work focuses on different ways in which explicit causal statements can be identified. Khoo et al learn cause-associated patterns, e.g. those anchored by cue words like *because* (Khoo et al., 2000). In the semantic role-labeling task (Gildea and Jurafsky, 2002; Chen and Rambow, 2003; Hacioglu et al., 2004) as well as in sentence-level discourse parsing (Soricut and Marcu, 2003), researchers map the structure of entire sentences within a larger set of relations which include concepts of causality, relying heavily on the syntactic relationship of sentence constituents.

Again, what these approaches share is a focus on relation *form* rather than *content*. That is, they identify causally related constituents based on how they are joined by text, structurally, lexically or otherwise. This is distinct from Marcu and Echihabi's (Marcu and Echihabi, 2002) approach, which builds a model of relation content, focusing on the constituents themselves, rather than the connections between them. All of this gets back to a primary motivation for Marcu and Echihabi's work, namely the identification of implicit relations. As we discuss later in the chapter, when we consider the challenges of relation-focused questions, models like Marcu and Echihabi's can also help us detect the implicit connections between question and answer.

Causality has also been studied in NLP applications beyond QA. For instance, in text generation, both Mittal and Paris (Mittal and Paris, 1995) and Lester and Porter (Lester and Porter, 1997) explore the problem of explanation generation. Mittal and Paris focus on contextual generation to provide textual explanations of logical reasoning in an expert system, whereas Lester and Porter's work centers on issues of content planning in generating academic explanations of biological processes. However, neither of these systems focus on inferring or deriving information that is causal, relying instead on detailed ontologies which explicitly encode notions of causality and concept hierarchies. An interesting finding in Lester and Porter is that it is possible (and useful) for causal explanations to be flexible to different "views" of an event, e.g. a view of *photosynthesis* as either a process with input and output materials, or alternatively as a transduction between different types of energy. When considering alternate content plans, one can therefore choose the view which is optimal with respect to other constraints. This observation is important with regard to answering and evaluating relation-focused questions, inasmuch as it supports the idea that there can be multiple valid causal descriptions, and that choosing the "right" one can depend on contextual considerations.

The concept of contrast has consistently been included in studies of discourse and rhetoric. In Chapter 3 we discuss various such theories and list in Table 3.1 some of the proposed relations which touch on the concept of contrast. Others have focused more specifically on uses and linguistic manifestations of contrast. For instance, Umbach (Umbach, 2001) studies the ways in which contrast is used in question-and-answer dialog, identifying uses which are both rhetorical ("denial-of-expectation") and semantic ("semantic opposition") in nature.

In terms of applications, particularly in QA, work which specifically addresses questions with a contrastive or comparative focus is mostly speculative. Burger et al. (Burger et al., 2001) identify "comparative questions" as an important future direction in QA. Interestingly, they categorize these questions under a class of questions requiring implicature; for instance in answering, "How rich is Bill Gates?" they consider together the challenges of

(1) determining that a term like *ownership* is relevant to a comparison of *rich*-ness and (2) inferring that *ownership* can pertain over a company, and if that company is large and profitable (like Microsoft), ownership implies wealth. However, while (2) may require deep semantics and reasoning, (1) seems amenable to shallower methods. Harabagiu and Maiorano (Harabagiu and Maiorano, 1999) identify contrastive questions as one of five broad question types (in this proposed taxonomy, TREC-style factoid questions fall into a single type), but do not explore answer techniques for such questions.

Looking beyond QA, other NLP applications can also provide output which is in some sense implicitly comparative, for instance in classification of positive versus negative opinion statements on a given topic (Yu and Hatzivassiloglou, 2003), or polarity classification of consumer product reviews (Popescu, 2005). That is, if we run a positive-versus-negative opinion classifier on a set of news articles about, e.g., welfare reform, the resulting sets of positive and negative opinions can be viewed as an answer to the question, "Compare opinions on welfare reform." Information extraction approaches in domain-template scenarios as in the Message Understand Conference (MUC) (Grishman, 1997) can also support implicit comparisons. For instance, if we consider the MUC-4 conference (Sundheim, 1992), in which systems identified template information in the domain of terrorist attacks, one can imagine that the results of information extraction could be used to support comparisons of different attacks based on points of domain interest such as perpetrator, number of casualties, etc. However, a critical limitation of using these methods to answer comparative questions is that the domains of interest and/or points of comparison must be known *a priori*, and thus are less useful in a dynamic setting, where we may be asked to compare new classes of objects, and/or compare along different axes.

5.3 Relation-Focused Questions

We conceive of *relation-focused questions* as a broad class of questions which we, as humans, may answer by referencing intuitive models of RSR-like relations. We do not attempt to define these questions more precisely here; rather, we specify only that they are questions which can potentially be answered more accurately using models of relations such as Cause, Contrast, and other rhetorical and semantic relations included in the various theories of rhetoric and semantics discussed in the related work section of Chapter 3. We attempt in this section only to sketch the outlines of this *relation-focused* concept as a more abstract notion; then, in the remainder of the chapter, we focus on a specific subset of these questions and the development and analysis of techniques for answering them.

Relation	**Explicit Question**	**Implicit Question**
Cause	What caused the Great Chicago Fire of 1871?	How did the Great Chicago Fire of 1871 start?
Contrast	Compare the economies of the Korea and France.	How strong are the economies of Korea and France?
Whole-Part	What nations are part of OPEC?	What is the structure of OPEC?
Sequence-Temporal	What preceded the Bronze Age?	When did the Bronze Age begin?

Table 5.1: Implicit and Explicit relation-focused questions for several relation types.

In her study of causal questions, Girju (Girju, 2003) notes that a question can reference the concept of causality either explicitly or implicitly. As she points out, questions which include a causal element may not necessarily begin with *Why*, or use keywords like *cause* or *effect*; rather, they may be realized in many alternative ways.

Extending this notion, we list in Table 5.1 questions which either explicitly or implicitly have a relational aspect with respect to various relations, including our Cause and Contrast. We also list questions formed around other relations, namely the Whole-Part and Sequence-Temporal relations listed by Hovy and Maier (Hovy and Maier, 1993), to demonstrate that the relations around which such questions may be focused are not limited to those we happen to model.

Clearly there are any number of research issues to be addressed in answering such questions in the general case. In the remainder of this chapter, we explore a specific sub-area of this problem; however, before narrowing our focus, we briefly consider some of the big-picture issues in answering such questions:

Question Parsing and Typing Most basically, how can we identify which questions should be analyzed as relation-focused, and determine which relations are relevant, and to which parts of the question?

Furthermore, even if we assume that we are able to parse a question like "When did the Bronze Age begin?" into a form that recognizes it as an implicit request to situate the concept of *Bronze Age* in a time sequence, it is still not clear that a relation-focused approach is the correct one. That is, an approach which invokes a Temporal-Sequence relation model may not yield our best chance at finding a good answer. Even if we might, as humans, invoke some notion of time sequences to answer that question (e.g., by considering what other *ages* we know about, and thinking about whether *stone* most likely comes before or after *bronze*), we may be better served from a system perspective to view such a question from a type-centric viewpoint, searching our corpus annotations of a TIME-DATE type near the terms *Bronze Age* and *begin*.

At some level, the reasoning as to whether a relation-focused approach is desirable may center on the type of response sought. Some users may be looking for a short, factoid style answer, i.e. "before 3000 B.C." For others, a multi-sentence, long-answer style answer which summarizes information from a relational perspective, i.e. with regard to historical developments before, after or during the Bronze Age, will be preferable.

Answer Finding: Document, Passage and Factoid Identification Assuming a question analysis which identifies some relation-focused component to a question, how should this carry through to the tasks of identifying relevant information at document, passage and factoid levels? If we determine that a question is asking about causes of a *fire*, how can we apply a model of Cause at each stage in an information filtering pipeline for optimal results? That is, should we look only for clauses which, at some deep semantic level, have a propositional content which asserts something like *cause(X, fire)*? Or at the other end of the spectrum, we can make the contribution of a relation-aware component at the document retrieval level, e.g. by adding document search terms which are linked in a semantic model as causes of *fire*, e.g. *gasoline* or *wind*.

Relation Modeling Assuming we have a strategy for using a "generic" relation model to somehow assist the QA task, what system, what actual model implementation techniques will fit best with such a system? That is, in our work in Chapter 3, we evaluate, train and test TextRels based on related text spans taken from within document context, and model our relations in a word-pair based framework which achieves high coverage but does not take into account potentially important issues such as polysemy, syntactic roles, or multi-word terms. Any of these or many other factors may be useful or even critical for a relation model to be used in this task.

In the next sections, we explore only a small subset of these issues, in order to work with a more tractable problem. In particular, we focus on passage retrieval within the "answer finding" step discussed above, using RSR models to help identify relevant information. For many of the other hurdles, we either simplify the problem (for instance, by using question templates which do not need to be parsed), or hold other variables constant (for instance, by exploring the use of only one kind of relation model, i.e. our TextRels model).

5.4 Narrowing Our Focus: Two Question Types

From a potentially very broad set of questions which might benefit from the application of RSR knowledge, we narrow our focus to include two types of questions only, *Explanation* and *Comparison* questions. The motivation in selecting these two question types is to use knowledge in our RSR models for Cause and Contrast to enhance our answers for explanation and comparison questions, respectively.

5.4.1 Explanation Questions

The template for explanation questions is: *Describe [causes | effects] of [Focus] in [Topic].*

The three bracketed sections of the question indicate structured slots to be filled as follows:

- In the first slot, we can specify the question subtype as either *cause* or *effect.* We refer to the respective question subtypes as Explanation-Cause and Explanation-Effect.

- The second slot, the question *Focus*, specifies the core concept for which we are seeking an explanation.

- The third slot, the question *Topic*, specifies broadly the general area of interest in which the *FOCUS* should be interpreted and its explanation pertain.

Using this template, we can form questions like: "Describe [causes] of [conviction] in [Edmond Pope Convicted for Espionage in Russia]." The interpretation of such a question is that we seek material not only on the Topic of Pope's conviction, but specifically on *causes* of it. Thus, for instance, information on jury deliberations would be relevant, whereas information on Pope's family history or the prison sentence Pope received would not be. (Note that if we change the directionality of the explanation to ask about effects, rather than causes, the relevance is different, i.e. the jury is not relevant but the prison sentence is.)

5.4.2 Comparison Questions

The template for comparison questions is: *Compare [Topic-A] and [Topic-B] with respect to [Focus].*

- In the first and second slots, the Topic fields specify the two primary entities or issues which are to be compared.

- The question Focus specifies the key axis of comparison around which we are meant to compare the two Topics.

The comparison template can be used to construct questions like: "Compare [Korea] and [France] with respect to [economic outlook] ." In interpreting such a question, relevant information should not only be generally about the economy in Korea or France, but should highlight facts which facilitate a *comparison* of the two economies. In this sense, information on growth predictions or employment rates (i.e., common metrics for assessing economic outlook) would be more relevant for this answer than, say, the appointment of a new treasury secretary in France.

As we discuss in more detail in Section 5.7, assessing the “comparability” of a single fact atomically can be somewhat difficult; that is, while GDP growth is a far more typical benchmark on which to compare two economies, a comparison *could* be made between France’s new treasury secretary having green eyes, and Korea’s brown. Whether comparisons motivated by a model of “typical” comparative dimensions are in fact deemed preferable by users is an empirical question which we do not answer directly, but on which our system evaluation may at least shed some light.

5.5 Implementation: CCQ

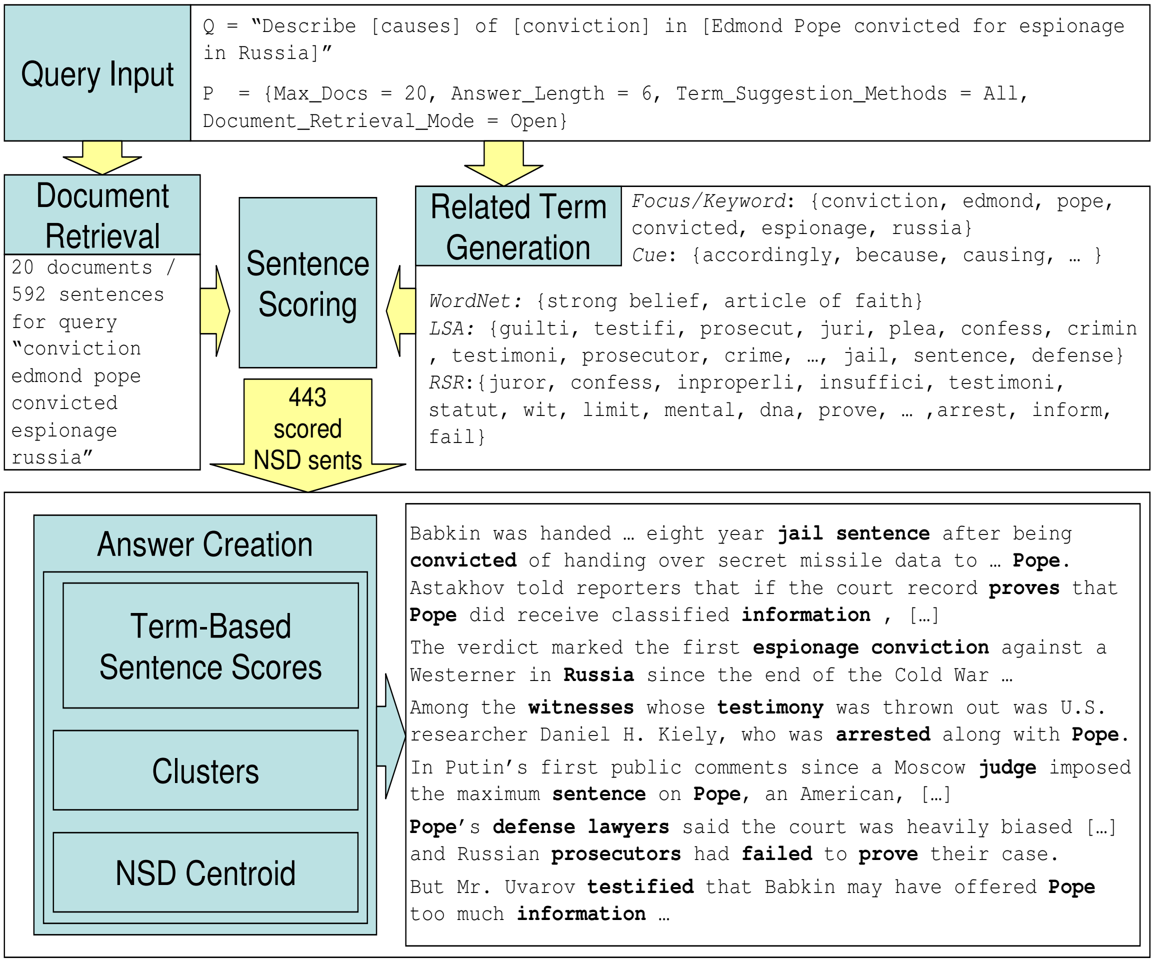
We implement a pipeline for answering these Explanation and Comparison questions in the long-answer style which we have explored in earlier chapters, i.e. by presenting a multi-sentence summary of the relevant information. We call our implemented system CCQ (“Cause and Contrast Questions”).

In CCQ, we implement new components for query input, document retrieval and the application of RSRs and other techniques for identifying and scoring relevant answer sentences. We then extend several DefScriber modules which we repurpose to create an answer from these scored sentences. We first present a high-level description of the CCQ architecture, tracing an actual system run, and provide further detail about the methods and parameters used in each module in Section 5.6.

5.5.1 CCQ Pipeline Overview

Figure 5.1 illustrates a module-level view of our CCQ pipeline running for the explanation question “Describe *causes* of *conviction* in *Edmond Pope convicted for espionage in Russia* .” Our processing proceeds in the following stages:

Query Input The query includes the question Q, which is entered according to the comparison or explanation template. In addition, the query specifies values for a set of parameters P which guide the answering process, including the number of documents from which to extract an answer (in this case, 20) and length of the desired answer in sentences (6).



Q = "Describe [causes] of [conviction] in [Edmond Pope convicted for espionage in Russia]"

P = {Max_Docs = 20, Answer_Length = 6, Term_Suggestion_Methods = All, Document_Retrieval_Mode = Open}

20 documents / 592 sentences for query "conviction edmond pope convicted espionage russia"

Focus/Keyword: {conviction, edmond, pope, convicted, espionage, russia}
Cue: {accordingly, because, causing, … }

WordNet: {strong belief, article of faith}
LSA: {guilti, testifi, prosecut, juri, plea, confess, crimin , testimoni, prosecutor, crime, …, jail, sentence, defense}
RSR:{juror, confess, inproperli, insuffici, testimoni, statut, wit, limit, mental, dna, prove, … ,arrest, inform, fail}

Babkin was handed … eight year **jail sentence** after being **convicted** of handing over secret missile data to … **Pope**.

Astakhov told reporters that if the court record **proves** that **Pope** did receive classified **information** , […]

The verdict marked the first **espionage conviction** against a Westerner in **Russia** since the end of the Cold War …

Among the **witnesses** whose **testimony** was thrown out was U.S. researcher Daniel H. Kiely, who was **arrested** along with **Pope**.

In Putin's first public comments since a Moscow **judge** imposed the maximum **sentence** on **Pope**, an American, […]

Pope's **defense lawyers** said the court was heavily biased […] and Russian **prosecutors** had **failed** to **prove** their case.

But Mr. Uvarov **testified** that Babkin may have offered **Pope** too much **information** …

Figure 5.1: Module level processing for relation-focused question "Describe *causes* of *conviction* in *Edmond Pope convicted for espionage in Russia* ," as described in Section 5.6. Words in the answer which match terms from related term generation are shown in **bold**.

Document Retrieval In document retrieval, we retrieve a set of query-relevant documents from which we will extract an answer. We retrieve documents from a locally-indexed collection of newswire articles, using a probabilistic search-engine to return documents for a keyword query formed from a concatenation of fields from the input question template instance. In our example, we retrieve 20 documents, or 592 total sentences, for the query "conviction edmond pope convicted espionage russia."

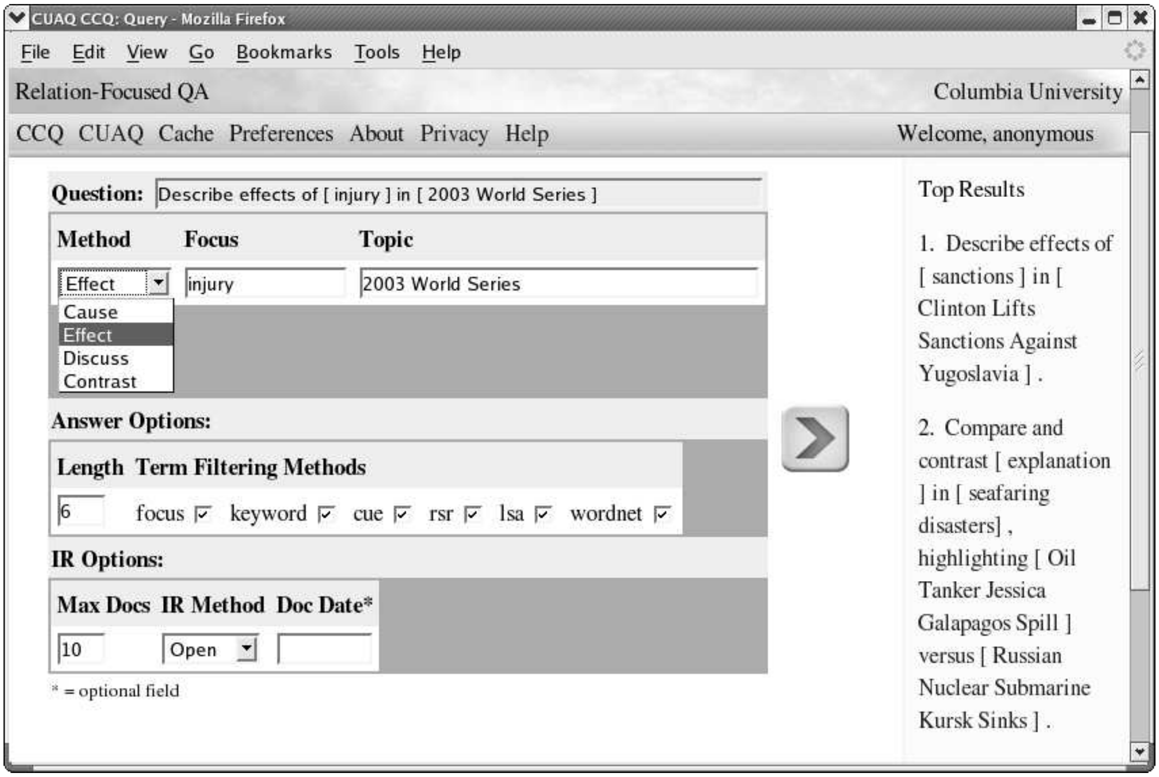
Related Term Generation In this step, we generate lists of question-relevant terms, which will then be used in the following step to seek out relevant sentences in the documents returned from Document Retrieval. Note that this step occurs in parallel with Document Retrieval; it relies only on the input query itself. We implement six methods of term generation which vary in complexity from methods which simply echo back keywords from the input question template, while other methods dynamically query our TextRels RSR models to suggest related terms (*RSR*). In the example run in Figure 5.1, all six methods are used (in the figure, there are only five lists since *focus-* and *topic-keywords* terms are shown in the same list). For the LSA and RSR methods, we show the top ten suggested terms in order, as well as any other suggestions which match terms in the answer (the other methods produce unordered lists). For a given query, its parameters may specify whether to use all term generation methods, or only a select subset.

Sentence Scoring This step scores each sentence in each document retrieved in Document Retrieval based on overlap with the term lists generated in Related Term Generation. The output of this scoring is a ranked list of sentences with non-zero scores which we call "NSD" sentences since they form the broad set of question-relevant material from which we will extract an answer, and in this way are analogous to the "Non-Specific Definitional" (NSD) sentences of DefScriber. In the example run, 443 of the 592 input sentences are retained in our NSD list.

Answer Creation In this step, we form an answer by integrating several modules adapted from DefScriber. In particular, we compute cluster and centroid data over the set of NSD sentences. We then use the feature-based sentence selection framework developed in Chapter 4 to construct an answer which considers both the term-based scores from Sentence Scoring, alongside cluster- and centroid-based sentence scores. Figure 5.1 shows the system answer, with terms which match in the lists from Related Term Generation in bold (some sentence parts are truncated for length).

5.5.2 Web Interface

In order to support ad-hoc experiments with the CCQ system pipeline, we integrate it within the Web-based interface to the CUAQ system, which integrates several QA-related


CUAQ CCQ: Query - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Relation-Focused QA
Columbia University
CCQ CUAQ Cache Preferences About Privacy Help
Welcome, anonymous
Cause
Effect
Discuss
Contrast

Question: Describe effects of [injury] in [2003 World Series]

Method	Focus	Topic
Effect	injury	2003 World Series

Answer Options:

Length	Term Filtering Methods
6	focus ☑ keyword ☑ cue ☑ rsr ☑ lsa ☑ wordnet ☑

IR Options:

Max Docs	IR Method	Doc Date*
10	Open	

* = optional field

Top Results

1. Describe effects of [sanctions] in [Clinton Lifts Sanctions Against Yugoslavia] .

2. Compare and contrast [explanation] in [seafaring disasters] , highlighting [Oil Tanker Jessica Galapagos Spill] versus [Russian Nuclear Submarine Kursk Sinks] .

Figure 5.2: Query input screen for the Web-based interface to our CCQ relation-focused QA pipeline.

projects jointly developed at Columbia and the University of Colorado. Figure 5.2 shows the query input screen for CCQ as displayed in a browser.

While development of the interface *per se* is not a focus of this thesis, we do use screenshots from it in several of the following sections to help display the functionality of particular CCQ modules.

5.6 CCQ Modules

With the pipeline overview as context, we now provide in this section a more detailed view of each module in the CCQ architecture.

5.6.1 Query Input

Our system accepts input as a pair (Q, P), composed of Q, the explanation or comparison question, and P, a set of parameters which guide the answering process.

The question must follow one of the two structured templates defined in Section 5.4, i.e. for Explanation or Comparison questions. The question can thus be one of three types, Explanation-Cause, Explanation-Effect, or Comparison. The question also contains several fields which contain unrestricted (but not empty) text, namely the Topic and Focus fields, where Explanations have a single Topic and Comparisons have Topic-A and Topic-B.

As we mention earlier, our primary motivation for using template-style questions in our system is to reduce problem dimension and avoid implementing a question-parsing step to deal with question format variation. However, in our interface, while users enter question content in a field-by-field manner to avoid malformed queries, we automatically display the natural language interpretation of the question in order to minimize any effect of the template restriction on query comprehension. For example, in Figure 5.2, we see that the user has selected “Effect” as the query type and entered “injury” and “2003 World Series” as the Focus and Topic respectively, which causes the Question field to show the interpreted question, “Describe effects of [injury] in [2006 World Series].”

In addition to the question itself, the query specifies values for a set of query parameters, P, which guide the answer process:

Answer length The desired number of sentences in the answer.

Term filtering methods For each of the implemented methods in the Related Term Generation step (detailed below), term generation can be enabled or disabled.

Maximum documents For the Document Retrieval step (described below), the maximum number of documents to fetch.

IR Mode For the Document Retrieval step, this parameter specifies retrieval in Open, Assisted or Oracle mode (described below).

Document Date Parameter for Document Retrieval in Open mode specifying the approximate publication date of documents of interest.

Topic ID Parameter for Document Retrieval in Assisted or Oracle mode, specifying a topic ID number for documents of interest (where documents in the collection have been pre-annotated with topic numbers).

5.6.2 Document Retrieval

We retrieve documents in one of three possible modes, namely *Open*, *Oracle* and *Assisted*. The mode to use is indicated as one of the query parameters, as is the maximum number of documents to retrieve.

In the open mode, an information retrieval query is issued to an instance of the Indri search engine (Metzler and Croft, 2004)[4] indexing the TDT-4 and TDT-5 text document collections[5]. The query issued for Explanation questions simply concatenates the terms of the question Topic and Focus. By default, query terms are sent to Indri as optional terms in the vector query sense.

For Comparison questions, two distinct queries are issued. Both start from the concatenated query used for Explanation questions, i.e. the concatenation of Topic and Focus terms. Then, for each of the two Topics, a modified query is formulated which emphasizes one Topic by requiring its term(s), and de-emphasizes the weight of the other Topic's term(s). Table 5.2 shows examples of question-query mapping for two example queries[6].

Additionally, for Open queries, we implement the option to include a date field (or in the case of comparison questions, one date field for each subtopic) as an additional query term when we wish to retrieve documents published on or about that date. This option takes advantage of a feature in our Indri index which includes document date as a special

[4] Available from: `http://www.lemurproject.org`. We use the Indri search engine for the experiments reported in this chapter. In principle, any keyword-based search engine which implements query operators for required terms and term weighting can be used in the manner we describe.

[5] An overview of the Topic Detection and Tracking (TDT) project is described by Wayne (Wayne, 2000). A number of project resources are also available online from `http://projects.ldc.upenn.edu/TDT/`.

[6] The queries shown in this table are written for general comprehensibility rather than in the actual Indri query language; for example, the second query for the comparison question in Table 5.2 is rendered in the actual Indri language as: `#filreq(#uw(france) #weight(0.9 #combine(economic debt outlook france) 0.1 #combine(korea)))`

Question	Query	Query 2
Describe causes of [conviction] in [Edmond Pope Convicted for Espionage in Russia]	conviction edmond pope convicted espionage russia	-
Compare [Korea] and [France] with respect to [economic outlook].	+korea economic outlook −france	+france economic outlook −korea

Table 5.2: Example question-query mappings for CCQ explanation and comparison questions run in Open document-retrieval mode. Two queries are generated for Comparison questions; only one for explanation questions. The plus (+) sign indicates that a term is required in any returned document; the minus (-) sign indicates a term the search engine should weight below other query terms.

Topic ID	Topic Description
40002	UN Climate Conference - The Hague (Netherlands)
40015	Floods in China's Hainan Province
40048	Train Fire in Austrian Tunnel
41007	Parliamentary Elections in Egypt
41015	Clinton Lifts Sanctions Against Yugoslavia

Table 5.3: Several example topic descriptions from TDT. Typical topics include accidents and natural disasters, as well as political and diplomatic happenings.

indexing term; however, we have not experimented significantly with this feature and do not discuss it further here.

We also implement an Oracle mode in order to experiment with the effect of real versus ideal IR performance on the overall pipeline. This is meant primarily as an option to be used in evaluation circumstances, as we do in the evaluation section of this chapter. In the Oracle mode, we use human-annotated relevance judgments to return only relevant documents for a given TDT topic. We have implemented this ability for the TDT-4 and TDT-5 topic sets, so that the user can specify any of the 330 TDT topics as an additional parameter, in which case any documents annotated by humans as being relevant to the topic are returned, rather than running a keyword query against Indri. Table 5.3 lists several example TDT topic statements.

Note that for comparison queries in oracle mode, one topic number is specified for each subtopic, and two sets of documents are retrieved. If, for any TDT topic, the number of human-judged relevant documents exceeds our maximum documents parameter, the maximum is enforced by randomly truncating the full set of documents.

In Assisted mode, we generate and issue queries just as in open mode, but augment the document list retrieved by Indri by prepending any available Oracle-based documents. We implement this mode to have a simple way of adding additional documents to the human-marked TDT relevance sets, which are in some cases incomplete. However, we have not evaluated the performance of this mode and do not discuss it further.

An important note is that regardless of query mode, we retrieve two sets of documents for Comparison questions, with one set associated with each question Topic. In the output of Document Retrieval, we pass forward both document sets, maintaining pointers to remember which documents are associated with which Topic.

5.6.3 Related Term Generation

In this step, we generate lists of question-relevant terms which will be used to score and filter the sentences contained in the documents which are returned from Document Retrieval. We implement six such methods overall; the query parameters may specify to use all methods, or a select subset only. (In our evaluation in Section 5.7, we experiment with three distinct subsets.)

5.6.3.1 Keyword and Cue Terms

The first two Term Generation methods, *Focus-Keywords*, *Topic-Keywords* simply "generate" by echoing back the terms in the Focus and Topic fields of the question. Using Focus and Topic terms to evaluate the question-relevance of a given sentence is clearly important, since these terms are by definition relevant to the question. As we discuss in earlier chapters, using the density of question terms in a passage as a gauge of sentence-question relevance proves successful in DefScriber for identifying NSD sentences; others have also found this to be a robust technique (Tellex et al., 2003).

The next method we use, called simply *Cue*, uses a static list of cue words which depend on the question type itself. That is, it generates a list of terms including *because* and *therefore* for Explanation questions, and a list including *however* and *nevertheless* for Comparison questions. These lists are manually compiled from the cue-phrase patterns around which we extract Cause and Contrast instances in Chapter 3. This method is meant to be a coarse-grained method for finding explicit relational statements and a complement to the Keyword methods above. That is, while the Keyword methods help find sentences generally relevant to the question Topic/Focus irrespective of the relational question aspect, the Cue terms help find sentences expressing notions of the relevant relation (i.e. Cause or Contrast).

5.6.3.2 RSR Terms

While the Keyword and Cue-based terms can help to identify Topic, Focus and relation-relevant sentences individually, we do not wish to rely on finding a sentence which contains the perfect mix of Keywords and Cue terms. That is, recalling our earlier example question, "Describe causes of [conviction] in [Edmond Pope Convicted for Espionage in Russia]," the Keyword and Cue methods alone could help identify the relevance of a sentence like, "Pope was convicted for espionage because of eyewitness testimony which proved the case." Yet without the explicit marker *because*, we will miss the equally relevant sentence, "Eyewitness testimony proved the prosecution case against Pope." Moreover, we may find sentences which use causal cues and Topic/Focus terms, but in an unconnected fashion, e.g. "Because of the holiday, the decision on whether to convict Pope will wait until Friday."

For these reasons, we turn to our RSR models for a method which directly seeks out those terms which are relationally linked to the question Focus by the relevant model, i.e. the Contrast model for Comparison questions and the Cause model for Explanation questions. Using this technique, we can directly determine that terms like *prove* or *testimony* in our example above are of interest. We do this by querying our Cause model to find which terms are likely to occur in a cause relation with *conviction*. This allows us to identify relevant answer material whether or not any of the terms from our causal Cue list, or even the Focus term itself (i.e., *conviction*) are present.

We implement the *RSR* term suggestion method over our TextRels RSR models. Given the seed term(s) from the question Focus field, we generate a list of the most strongly linked words in the relevant relation model. Throughout the chapter we list examples of such suggestions for various question types and Foci, including in Figure 5.1.[7]

Applying Likelihood Ratios To generate a list of strongly linked terms, we use the log-likelihood ratio calculation set forth by Dunning (Dunning, 1994). We use log-likelihood in preference to χ^2 or pointwise mutual information because of its reliable properties with regard to estimating relative likelihoods for rare events[8].

Given a seed term s and a possibly related term t, we compare the likelihood of seeing the pair (s,t) in the context of our relation of interest (i.e. Cause or Contrast) versus simply seeing s and t in the same document (i.e. under the NoRelSame model). For instance, consider the case where we wish to determine whether the seed term stem *convict* is related by the Cause relation to the candidate term stem *testimoni.* Using the framework of Dunning, we view the Cause and NoRelSame models as parameter spaces Ω_1 and Ω_2 in which the occurrence of terms t in pairs of the form (*convict, t*) are distributed according to parameters p_1 and p_2 respectively. The hypothesis we test is whether these parameters are in fact the same; i.e. whether there is some p such that $p_1 = p_2 = p$.

The likelihood ratio for this test uses the binomial distribution to estimate the probability of k successes in n Bernoulli trials, which are often referred to as "idealized coin-flips". The information in our word pair models in Chapter 3 can be mapped onto these binary trials as follows. Imagine we wish to measure the association between the term stems *convict* and *testimoni.* We can consider our set of trials (n) to be the cases where we observe a relation instance between two text spans, and one of the spans contains a term with the stem *convict.* We can consider our successes (k) to be the subset of those trials where the other text span, i.e. the other "half" of the relation instance, contains a term with the stem *testimoni.* We thus have k_1 and k_2 as the frequency of the word pair (*convict, testimoni*)

[7] In some cases, terms in the Topic field might also be appropriate seeds. However, by making these fields distinct, we purposely allow for the Topic to contain background terms such as names or other identifiers which are needed to establish the general topic of the question, but would not necessarily be appropriate seeds.

[8] See Dunning (Dunning, 1994) and Manning and Schütze (Manning and Schütze, 1999, pp.172-4,178-81) for a detailed discussion.

in our NoRelSame and Cause models, respectively, and n_1 and n_2 as the total frequency of word pairs containing *convict* in those same models.

As explained in Dunning, the likelihood ratio λ reduces to:

$$\frac{\overset{\max}{p}\ L(p, k_1, n_1)L(p, k_2, n_2)}{\overset{\max}{p_1, p_2}\ L(p_1, k_1, n_1)L(p_2, k_2, n_2)}$$

Where $L(p, k, n)$ is the binomial probability if k successes in n trials where the probability of success in any one trial is p, i.e. $p^k(1-p)^{n-k}$. We can read off maximum likelihood estimates for p, k and n from our Cause and NoRelSame models as:

$$p_1 = p_{Cause} = \frac{k_1}{n_1}, \quad p_2 = p_{NoRelSame} = \frac{k_2}{n_2}, \quad p = \frac{k_1+k_2}{n_1+n_2}$$

In the case of the term pair (*convict, testimoni*), we have:

$$p_{Cause} = \frac{91}{34{,}853} \quad p_{NoRelSame} = \frac{96}{279{,}794}$$

In practice, we use these values to compute the quantity $-2\log\lambda$ based on the likelihood ratio λ given above, since the former is asymptotically χ^2 distributed, and we can thus use the critical values for the χ^2 test to evaluate the resulting ratio. For (*convict, testimoni*), we find $-2\log\lambda = 128.94$, well above the critical value for χ^2 with one degree of freedom[9], which is 6.64 for $p < 0.01$, or 10.83 for $p < 0.001$.

Parameters and Implementation Using these likelihood ratios, we can make term suggestions based on our RSR models. However, there are several key parameters in our RSR Term Generation implementation which we note here.

RSR Model Training As discussed in Chapter 3, there are a number of parameters which modulate the training of our RSR models. Our general principle in choosing which model parameters we use to construct the models we use for Term Generation is to use those which perform best on the relevant classification tasks, i.e. Cause vs NoRelSame and Contrast vs NoRelSame. Based on this criterion, we use the models trained on topic-segment constrained instance data as described in Section 3.5.2,

[9] The degrees of freedom are determined by the difference in the number of parameters, in our case one since we are considering a single parameter for each of two models.

which achieves the overall performance on these classifications, and using other model training parameters as optimized in Section 3.5.1.

In addition, we train the models using the full set of instance data (i.e. the "Extracted Instances" counts in Table 3.2). Note that this adds additional training data ("training set 3") to models evaluated in Chapter 3 (which are trained on the training sets 1 and 2 only). By using the full data we aim to make the models as robust as possible, noting that our classification-based evaluations support the idea that adding data at this scale increases the differentiability of the models (see, e.g., Figure 3.3).

Using RSR Directionality The careful reader may have noticed above that in our earlier exposition of using log-likelihood for term suggestion, we discuss using the Cause model to derive term suggestions for an Explanation-Cause type question, yet failed to explain how the process would differ for an Explanation-Effect type question. To do this, we utilize the directional nature of the Cause patterns, which is maintained in the aggregated instance counts of our models.

In the patterns used to extract Cause examples, we not only capture two spans of text, but we also store the resulting term pairs according to a cause-effect semantic order associated with the pattern from which they were extracted. For example, in the single-sentence pattern "W_1 because W_2", the first matched span is understood semantically to be the effect, and the second the cause. (Each of the Cause patterns listed in Appendix B is annotated to indicate its associated directionality.) When we store the word pairs which are extracted for a given Cause instance, we reflect the semantics of the extraction pattern in the ordering of the pair, always storing terms from the cause text span in the first position, and terms from the effect span in the second. Thus, we add to the frequency count for (*hungry, ate*) when we see the sentence "He ate because he was hungry.", as well as if we see "He was hungry, and therefore ate." This is because while *hungry* and *ate* appear with opposite surface ordering here, their semantic ordering is the same.

This ordering in the model can easily be translated to the term suggestion process. Specifically, when looking up probability estimates for an Explanation-Cause ques-

candidate position	**suggested terms**
combined	postpon, partli, price, melt, cancel, jam, tide, shortag, harvest, tax, delai, traffic
cause	melt, tide, rainfal, inch, torrenti, dump, warm, lash, drainag, fell, high, drain
effect	postpon, price, partli, cancel, traffic, delai, jam, harvest, shortag, ton, shut, unabl

Table 5.4: Top-ranking term stems generated from the Cause RSR model for the seed term *flood*. The combined row shows terms likely to appear in either semantic direction; the cause versus effect distinction is made by looking for likely terms only in the appropriate semantic position as described in Section 5.6.3.2.

tion, we can read off the likelihood estimates from the Cause model using only the frequencies for which the seed term is in the effect position (since we are looking for candidate terms in the cause position). For Explanation-Effect questions, we do the opposite. We find that these procedures can indeed capture the difference between the two "directions" of the Cause relation. For example, in Table 5.4, we observe that by generating terms likely to appear in the cause versus effect position of our model, we can generate lists which separate the two concepts with respect to the idea of a *flood*, as we would like to do when answering the cause versus effect versions of, "Describe (causes—effects) of flooding in China's Hainan Province." This is an important feature of our RSR-based term suggestion: we are able to operationalize the intuition that these questions should have different answers, by applying the directionality of an Explanation question.

On the other hand, our Contrast model and its associated patterns are symmetrical, as is our NoRelSame model. In these cases, the model simply records the surface ordering of the pair and we combine frequencies from term pairs in either direction when estimating the likelihood of a seed-candidate term pair.

Identifying Candidate Terms Thus far, we have explained the procedure for determining if a single candidate term, t is linked to a question's Focus term s. But how does this translate into coming up with a set of terms T which we return in the context of the CCQ Related Term Generation module? Because the vocabulary of our training

Focus Term(s)	RSR terms suggested for Explanation-cause question
environmental	hurt, fear, oil, leak, contamin, gasolin, water, land, tanker, threat, lack, coal, coast, 1989
regulation	interst, supplement, toxic, competit, act, combin, commerc, statut, concern, effect, heavi, contamin, problem, lack
environmental regulation	interst, supplement, toxic, hurt, fear, contamin, lack, oil, competit, threat, heavi, pose, concern, leak

Table 5.5: RSR Cause term suggestion examples for multi- versus single-word Focus terms. We show the first 14 suggestions which are generated for an Explanation-Cause question with the given Focus, ordered by decreasing $-2 \log \lambda$ score (all scores are above the significance threshold of 10.0).

model contains only 12,800 term stems, we can actually perform the likelihood ratio test for all terms t which co-occurred with our seed term using less than five seconds of CPU time even in the worst case (i.e., when the seed term is very common and the list of candidate terms is long). Then we can simply return the list of candidate terms for which our $-2 \log \lambda$ score is greater than our threshold. The threshold we use here is 10.0; this is born out as a reasonable threshold in manual analysis, but also can be justified more formally given that it is in the range of critical χ^2 values discussed earlier.

Multiword Seed Terms In the TextRels RSR models, we maintain frequency counts only for unigrams. Yet, we would like to be able to ask questions which focus on various subjects, including those best described with multiword terms. This flexibility will allow questions with a focus like *environmental regulation*, e.g. "Describe causes of *environmental regulation* in India." Of course, *environmental regulation* is a collocation, where neither component used alone represents the whole meaning, as are many multiword question Foci which we would like to be able to use[10] Nonetheless, we prefer

[10] An exception, which we have found useful in some development experiments, is to use a multiword focus to disambiguate a polysemous term, e.g. using *fire dismissal* or *fire burning* to disambiguate a question Focus of *fire* to the verb versus noun meaning.

using our unigram model to implement an approximate solution rather than omitting the ability to include such questions altogether.

Given that our TextRels models do not store frequencies for bigram pairs, we implement an approach which simply combines the observed frequencies for these terms when estimating the k and n frequencies in the log-likelihood calculation. This amounts to a slight generalizing of the question which our ratio test is asking, so that it can now be understood as, "If we see some seed term $s \in S$ in one half of a relation, are we more likely to see candidate term t in the other half based on one RSR model (e.g., Cause) versus another (e.g. NoRelSame)?" Here, the single word seed case can be seen as the case where S merely contains one word.

The resulting suggestion lists will be similar to those which we would get if we combined separate term suggestion lists for each component term in the Focus, e.g. by interleaving; however, using this procedure results in a ranking which accounts more easily for overlap between the lists, i.e. cases where the same term might be suggested with different ranking/score for two separate components of the Focus.

Table 5.5 shows the result of our RSR term suggestion for an Explanation-Cause question with the Focus *environmental regulation.* We can see that suggestions for causes of *environment* clearly discuss environmental issues. Suggested term stems for *regulation* describe some environment-relevant causes (e.g., *toxic*), and others which would appear to relate more to corporate or market regulation, (e.g., *commerc* and *combin*). We can see that in the suggestions for the multiword seed *environmental regulation*, stems which apply to the combined meaning, e.g. *contamin*, remain high in the list, whereas those which apply more clearly to general market regulation like *commerc* and *combin*, fall off the list. In addition, the new term *leak*, which seems highly relevant to *environmental regulation*, emerges from the combined evidence even though it was not rated as highly in relation to either of the components of the multiword term.

We also note that several terms in these lists seem problematic in certain ways such as overspecificity (e.g., *1989* , which we believe is in the list because it was the year of the infamous Exxon Valdeez oil spill) or underspecificity (e.g., *heavi*). We discuss these and other general issues with our RSR-based term suggestion process at greater length in Section 5.8.1.

5.6.3.3 LSA and WordNet Terms

In addition to the RSR term suggestion method, we implement two other semantically-motivated methods, *LSA* and *WordNet*. We add these methods in order to complement the RSR-based method, and also to be used as points of comparison in our evaluation.

The *LSA* method uses an implementation of Latent Semantic Analysis (LSA) (Deerwester et al., 1990) to find a set of terms which are semantically close to our focus term(s) using measures of distributional similarity. We implement this method by training on a large corpus of newswire data (a subset of the Gigaword corpus), computing distributional contexts at the document level. As is typically done in LSA, we use singular value decomposition to collapse the vectors into a smaller dimensionality space.[11] While choosing the "optimal" dimensionality for this task is an open problem, we find in practice that collapsing to a 100-dimensional orthogonal vector representation appears to yield reasonable results[12]. Given the seed term(s) from the Focus field, we can compute cosine distance between LSA vectors of seed and candidate terms. This yields a list of candidates which we can rank by decreasing latent semantic similarity with the seed (we use a manually estimated cutoff to truncate the list). Table 5.6 shows the term suggestions generated for the seed *flood*. While at least some of the generated terms do seem related, we note that as opposed to our RSR-derived list, we do not have any information on how the words in the list relate to flood, only that they are similar distributionally.

The *WordNet* method forms a list of related terms using various relationships in the WordNet (Fellbaum, 1998) database. As in the other dynamic methods, the seed term(s) from which the suggestion list is generated are the term(s) in the Focus field of the ques-

[11] We use the SVDPACK package available from `http://www.netlib.org/svdpack/`

[12] This choice is generally within the range of values used by others for NLP tasks like spell-checking (Jones and Martin, 1997), relation inference (Padó and Lapata, 2003) and document coherence measures (Barzilay and Lapata, 2005)

Method	**Suggested Terms**
LSA	blanket, shore, nearbi, arriv, mountain, rain, railwai, southern, northern, soil, southeast, surround
WordNet	deluge, flood, inundate, swamp

Table 5.6: Top-ranking term stems generated via the LSA and WordNet methods for the seed term *flood*, as described in Section 5.6.3.3.

tion. We use the following procedure for generating an (unordered) list of related terms: (1) look up the WordNet "synset" for each term in the question Focus[13] (2) query WordNet's database for terms linked to the synset by either a cause-to, antonym, synonym or entailment relationship (3) for explanation questions, return the list of cause-to, entailment and synonym terms; for comparison questions return the list of antonym and synonym terms. Table 5.6 shows the terms generated by this method for the seed *flood*.

Lastly, note that multiword seed terms are supported both for LSA and WordNet-based related term generation. For LSA, we use mean vectors computed over each component seed terms; for WordNet we first check if the multiword term has its own WordNet entry (WordNet does have entries for some collocations); if not we run term suggestion separately for each component in the multiword term and union together the resulting lists.

5.6.4 Sentence Scoring

In the Sentence Scoring module of CCQ, we use the term lists from Related Term Generation to score each sentence in each document returned from Document Retrieval. This is broadly similar to the part of DefScriber where we identify the set of Non-Specific Definitional (NSD) sentences. However, the critical difference is that in CCQ we not only score sentences based on the density of terms from the question itself, but also terms derived from the various

[13] Especially given the template style of our question, inferring the correct part-of-speech category for a focus term is ambiguous, since the terms are not necessarily in a parseable sequence. We default to a verb interpretation, which allows us to use a greater number of the relevant semantic relation types in WordNet, including entailment and cause-to, which are annotated only for verbs, and back off to other parts of speech only if no valid verb form exists in WordNet. Where multiple synsets exist, we select the first (synsets in WordNet are ordered based on occurrence frequency in corpus analysis).

semantically-based term generation methods.

We derive an intrinsic score for each sentence based on its lexical overlap with these related term lists, and then create an overall score which is a weighted sum of this intrinsic score and the intrinsic scores of other sentences from the same document which appear within a two-sentence window. More precisely, we compute the intrinsic score for a given sentence S composed of words $w_1 ... w_j$, given a set of related term lists R each of which consists of a set of term-score pairs (t, s) as:

$$\sum_{w \in S} \sum_{r \in R} \sum_{(t,s) \in r} \begin{cases} s & \text{if stem\_equal}(w,t) \\ 0 & \text{otherwise} \end{cases}$$

Where the *stem_equal*() routine returns true if the stemmed form of a word in the sentence equals the given term suggestion (all suggested terms are stemmed).

We primarily determine the s scores, i.e. the scores associated with each suggested term stem in each related term list, based on Inter-Document Frequency (IDF) values computed over a large corpus. The initial score for all of the terms in our term lists is set this way, with the exception of the Cue terms, which each have the same score (equivalent to the IDF score of a fairly frequent term, or more precisely one which occurs once in every ten documents). This is because we do not consider a hit on a rare cue phrase more significant than a hit on a common one, since both tell us the same thing, i.e. that the sentence contains some Cause/Contrast assertion.

On top of the initial IDF-based score, we apply a linear scaling factor to the values for the RSR and LSA methods according to the relative confidence/similarity score of the terms in the suggestion list. The intuition here is that, since these suggestion methods generate a scored list, we wish to take advantage of that information to emphasize those terms which are suggested with high confidence/similarity. The scaling maps the range of the confidence/similarity scores linearly onto the range $[1.0, 2.0]$. Thus the lowest-similarity suggested term simply keeps its IDF-based score, the highest-similarity term has its score doubled, and the intervening terms scale linearly.

Once these individual term scores are tallied for each word in each sentence as described above the end result is an overall intrinsic score for each sentence. As in DefScriber, a second pass computes an overall *query score* for each sentence, by adding its intrinsic score to a fractional amount of the intrinsic score of other nearby sentences, in order to reward those sentences in passages which have high term density. Finally, those sentences which have a positive intrinsic score become our NSD set of relevant sentences and are passed down the pipeline (along with their overall query scores) to the Answer Creation module.

5.6.5 Answer Creation

In the answer creation phase, we use the bottom-up modules of DefScriber to form a concise answer from the scored NSD sentence set identified by the Sentence Scoring module.

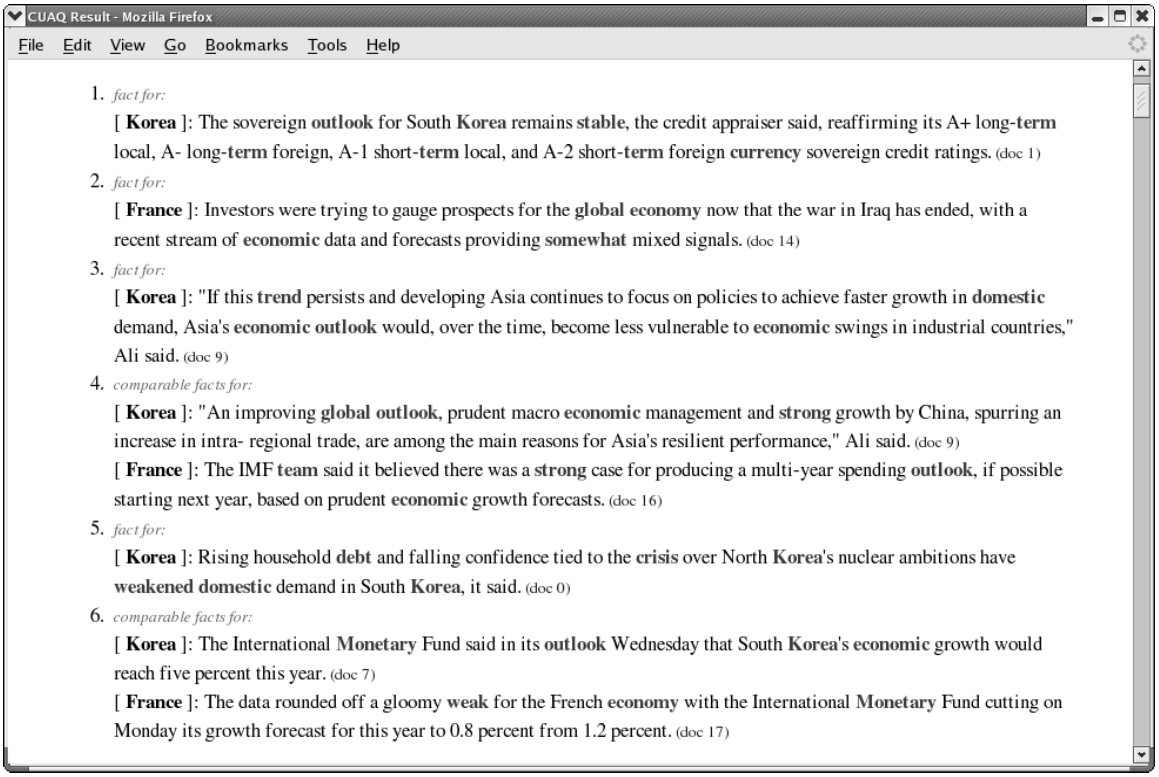
Taking the set of NSD sentences as input, we apply the *clustering* module from DefScriber to group sub-topics of redundant or strongly related information[14]. Additionally, we compute a single NSD *centroid* as a term-frequency vector which aggregates the term counts from all of the NSD sentences. We then calculate the centroid score of each sentence as its cosine similarity with the NSD centroid. (The implementation of these processes follows the descriptions of DefScriber's bottom-up modules in Chapter 2.) For Comparison questions, we create two clusterings and two centroids, one each for the sentence sets associated with each subtopic.

Thus each sentence is assigned a cluster label and centroid score, in addition to its query score as computed using the suggested terms described in the previous subsection.

For Explanation questions, we then select the first answer sentence as the sentence with the highest combined centroid and query score, using a manually estimated weighting to combine the scores. We then iteratively produce the remaining sentences up to the answer length limit using a version of the ChooseNextSentence() algorithm presented in the previous chapter. In this case, the algorithm's parameters are set such that the three features considered when evaluating next-sentence candidates are the query score, centroid score, as well as the cluster-based score (which penalizes sentences based on the number of already-selected sentences from the same cluster). We currently disable the coherence-based

[14] The clustering algorithm does not have access to the related terms lists discovered via the procedure described in Section 5.6.3. However, integrating this additional knowledge as a clustering cue would be an interesting subject for future work.

features because we are trying to focus on the efficacy of the related term-based query scores and wish to limit the number of factors influencing sentence choice in order to understand the impact of the related terms more clearly.



1. *fact for:*
   [**Korea**]: The sovereign **outlook** for South **Korea** remains **stable**, the credit appraiser said, reaffirming its A+ long-**term** local, A- long-**term** foreign, A-1 short-**term** local, and A-2 short-**term** foreign **currency** sovereign credit ratings. (doc 1)
2. *fact for:*
   [**France**]: Investors were trying to gauge prospects for the **global economy** now that the war in Iraq has ended, with a recent stream of **economic** data and forecasts providing **somewhat** mixed signals. (doc 14)
3. *fact for:*
   [**Korea**]: "If this **trend** persists and developing Asia continues to focus on policies to achieve faster growth in **domestic** demand, Asia's **economic outlook** would, over the time, become less vulnerable to **economic** swings in industrial countries," Ali said. (doc 9)
4. *comparable facts for:*
   [**Korea**]: "An improving **global outlook**, prudent macro **economic** management and **strong** growth by China, spurring an increase in intra- regional trade, are among the main reasons for Asia's resilient performance," Ali said. (doc 9)
   [**France**]: The IMF **team** said it believed there was a **strong** case for producing a multi-year spending **outlook**, if possible starting next year, based on prudent **economic** growth forecasts. (doc 16)
5. *fact for:*
   [**Korea**]: Rising household **debt** and falling confidence tied to the **crisis** over North **Korea**'s nuclear ambitions have **weakened domestic** demand in South **Korea**, it said. (doc 0)
6. *comparable facts for:*
   [**Korea**]: The International **Monetary** Fund said in its **outlook** Wednesday that South **Korea**'s **economic** growth would reach five percent this year. (doc 7)
   [**France**]: The data rounded off a gloomy **weak** for the French **economy** with the International **Monetary** Fund cutting on Monday its growth forecast for this year to 0.8 percent from 1.2 percent. (doc 17)

Figure 5.3: Example response presentation in the CCQ web interface for the Comparison question, "'Compare [Korea] and [France] with respect to [economic outlook]."

For Comparison questions, we follow a similar procedure, but rather than selecting one sentence at a time, we select sentences in pairs. Each pair includes one sentence from the document set retrieved for each Topic in the comparison. The sentence selection order within each Topic is essentially the same the same as for Explanations. For each sentence pair, if the lexical similarity of the two sentences is above a manually estimated threshold, we present them in the answer output as "comparable sentences"; otherwise, we interleave sentences from each subtopic and present them adjacently but do not explicitly group them as comparable. Figure 5.3 shows an example of how both comparable and interleaved facts are presented in a sample answer.

5.6.6 Parameter Estimation

Before turning to our evaluation, we note an overall issue which impacts many of the modules described above. Lacking a "gold-standard" evaluation corpus, we manually estimate various parameters to the modules described above. These include cutoffs used in various methods of Related Term Generation, as well as the weightings used for combine scores in Sentence Scoring and Answer Creation. Given limited resources and time for evaluation, we focus primarily on identifying and implementing these parameters, and manually estimate suitable values for various parameters. While this ad hoc procedure cannot claim to derive empirically optimal settings, we nonetheless feel that it derives reasonable baselines for this initial experiment. Furthermore, in the evaluation which follows, we do use these estimates in creating our responses. Thus, our results can be seen to some extent as an evaluation of how well the settings have been chosen.

5.7 Evaluation

We evaluate the performance of CCQ on both Explanation and Comparison questions with a user study in which we collect relevance and preference judgments from human evaluators.

5.7.1 Creating Test Questions

We use a set of 20 Explanation and 10 Comparison questions as test questions for our experiments. Several example questions are listed in Table 5.7.1; the full question set is listed in Appendix C, along with example responses.

We design this set of questions in a semi-automated manner, using topics from the TDT corpus. Before beginning development, we identify 104 TDT-4 or TDT-5 topics which have at least two relevant English-language documents in the corpus. We then randomly select a subset of 80 topics for possible use in test questions, and use the remainder during system development. After implementing and tuning the CCQ system, we proceed to create our test questions as follows:

- We select at random a topic from the held-out test set of 80 TDT topics. Then, based on manual examination of the topic, we decide whether it is suitable for an

Question Type	**Question**
Explanation-Effect	Describe effects of [sanctions] in [Clinton Lifts Sanctions Against Yugoslavia] .
Explanation-Effect	Describe effects of [fire] in [Train Fire in Austrian Tunnel] .
Explanation-Cause	Describe causes of [losing games] in [Australian Open Tennis Championship]
Comparison	Compare [Popocatepetl Volcano Erupts] and [Earthquake in India Gujarat State] with respect to [deaths injuries casualties] .
Comparison	Compare [Oil Tanker Jessica Galapagos Spill] and versus [Russian Nuclear Submarine Kursk Sinks] with respect to [explanations].
Comparison	Compare [UN Climate Conference] versus [Arab League Summit meeting in Cairo] with respect to [diplomatic strategy] .

Table 5.7: Example questions from the test set in our evaluation of the CCQ system.

Explanation-Cause, Explanation-Effect, and/or Comparison question (in several cases we ask more than one question about a topic).

- To create an Explanation question, we use the topic description from the TDT corpus to fill in the Topic field of the question (we edit to remove punctuation only). We then manually decide on an appropriate Focus for which we can ask an Explanation-Cause or Explanation-Effect question. In choosing the Focus term(s), we attempt where possible to pick a term which is used in the question topic itself, although we sometimes introduce a new term if none of the question words seems to encapsulate a plausible causal Focus. The decision of whether to form an Effect or Cause question for any individual case is subjective, but we impose the constraint that, in total, we must come up with an even number of each type so that our test set is balanced.

- Creating a Comparison question requires more human intervention. In this case, given the first (randomly selected) Topic, we manually scan the list of test TDT topic descriptions for a second topic which shares some significant dimension with the randomly selected topic. For instance, if the initial, randomly selected topic concerns an earthquake, we might manually choose a second topic about a different earthquake or natural disaster. These two topics are slotted into the Topic A/B fields of our Comparison question template. Then, as in Explanation questions, we manually choose a reasonable term to serve as the Focus of the comparison.

We create questions centering around these TDT topics for three primary reasons. The first is to avoid introducing undue bias into our questions based on our experience in developing the CCQ system. The second reason is pragmatic. Namely, we have integrated this first version of CCQ with only one search engine, which indexes a limited set of documents, namely the relevant TDT corpora. By using TDT topics in our questions, we can be confident that the document collection we access has at least a Topic-level overlap with our questions.

The third reason for using TDT topics is to be able to factor out the impact of Document Retrieval performance in returning Topic-relevant documents. While our method for creating questions ensures that question-relevant documents exist in the collection, that does not make the task of finding them trivial, since there are over 500,000 English-language documents in the collection, of which only 2,076 are marked as relevant to any of the TDT topics. By having a TDT topic ID associated with each question Topic, we have the option of using the CCQ Oracle-mode retrieval to gauge system performance in an ideal Document Retrieval situation.

To expand on this motivation, we note that in our ad hoc experiments with development questions, we did indeed observe wide variations in Document Retrieval performance carrying through, unsurprisingly, to affect end-to-end performance. Given that our CCQ term-based filtering uses expansion based on the Focus term(s) to filter retrieved documents, the potential impact of imprecise document retrieval can be magnified in this system. Consider the example question in the bottom row of Table 5.7.1, "Compare [UN Climate Conference] versus [Arab League Summit meeting in Cairo] with respect to [diplomatic strategy]." Our RSR-based term suggestion methods correctly identify that term stems like *aggress, bargain, conced* and *confront* are relevant here; yet if we match these terms in documents about another conference or meeting, they are of no help.

Of course, in some of the earlier evaluations of DefScriber, we face the issue of input document relevance too. While the DUC-based evaluations provide a set of relevant documents as input, the survey-based evaluation we conducted for DefScriber definitional answers in Section 2.4.1 required us to run our own document retrieval. However, in that case, we did not find that irrelevant documents were a major problem. There are several primary reasons for this, including (1) In DefScriber, we have the built-in advantage of having a short search phrase which encapsulates our topic of interest very well, namely the term we are trying to define. Building a good query for CCQ questions is not as straightforward. (2) The sentence-level filtering in DefScriber, where we identify NSD sentences, uses density of the definition target term as a primary criterion, thus acting to increase sentence-level precision further; as we have just explained, the term expansion approach of CCQ can have a negative impact here, especially with off-Topic documents. (3) In DefScriber, our IR is run through a Web search engine, giving us access to an enormous range of documents, and often finding many highly relevant documents in many cases. In CCQ, on the other hand, we run IR over a much smaller document collection, and the probabilistic search engine we use can sometimes return only peripherally relevant documents.

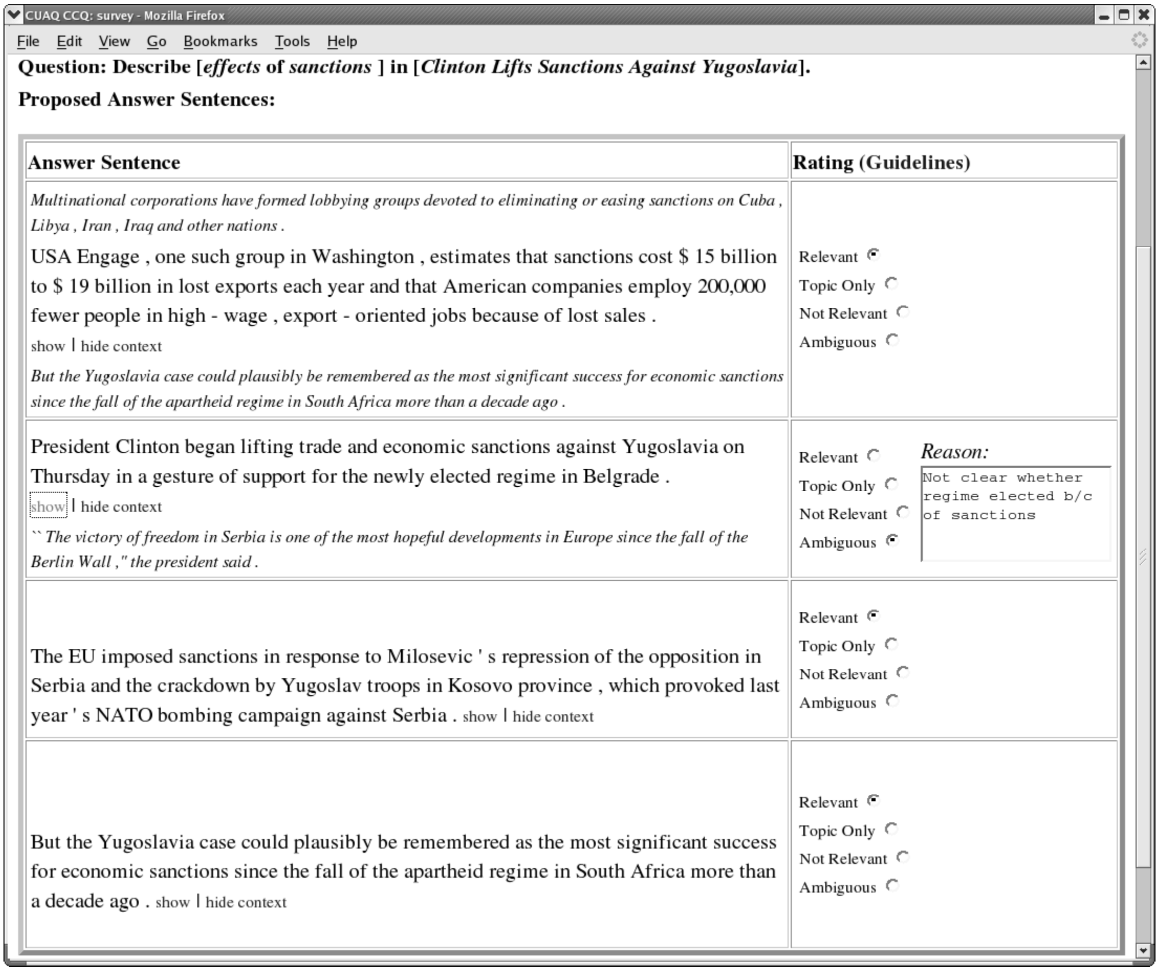
As we report in the next section, we ultimately experiment with both Oracle and non-Oracle document retrieval in different parts of our evaluation.

5.7.2 Evaluation Setup

We evaluate our system's performance on the test questions by collecting human judgments of answer quality. While there are a number of aspects of our system which could be evaluated, we are most concerned with measuring the contribution of using RSR model-derived terms for finding question-relevant sentences. With this in mind, we initially design an evaluation interface where users must rate each individual sentence in an answer into one of the following four categories[15]:

1. **Relevant** The sentence discusses the correct Topic (or, in the case of Comparisons, at least one of the two Topics) *and* gives information which is relationally relevant given the Focus and question type, i.e.:

[15] This summary of the categories is for the reader of the thesis who is already familiar with question types and the concepts of Topic and Focus; the actual user instructions, included in Appendix C, are more comprehensive.


CUAQ CCQ: survey - Mozilla Firefox
File Edit View Go Bookmarks Tools Help

Question: Describe [*effects* of *sanctions*] in [*Clinton Lifts Sanctions Against Yugoslavia*].

Proposed Answer Sentences:

Answer Sentence	**Rating (Guidelines)**
Multinational corporations have formed lobbying groups devoted to eliminating or easing sanctions on Cuba , Libya , Iran , Iraq and other nations . USA Engage , one such group in Washington , estimates that sanctions cost $ 15 billion to $ 19 billion in lost exports each year and that American companies employ 200,000 fewer people in high - wage , export - oriented jobs because of lost sales . show | hide context *But the Yugoslavia case could plausibly be remembered as the most significant success for economic sanctions since the fall of the apartheid regime in South Africa more than a decade ago .*	Relevant ● Topic Only ○ Not Relevant ○ Ambiguous ○
President Clinton began lifting trade and economic sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade . show | hide context *`` The victory of freedom in Serbia is one of the most hopeful developments in Europe since the fall of the Berlin Wall ," the president said .*	Relevant ○ Topic Only ○ Not Relevant ○ Ambiguous ● *Reason:* Not clear whether regime elected b/c of sanctions
The EU imposed sanctions in response to Milosevic ' s repression of the opposition in Serbia and the crackdown by Yugoslav troops in Kosovo province , which provoked last year ' s NATO bombing campaign against Serbia . show | hide context	Relevant ● Topic Only ○ Not Relevant ○ Ambiguous ○
But the Yugoslavia case could plausibly be remembered as the most significant success for economic sanctions since the fall of the apartheid regime in South Africa more than a decade ago . show | hide context	Relevant ● Topic Only ○ Not Relevant ○ Ambiguous ○

Figure 5.4: Example evaluation screen for a user rating an Explanation question. The user has clicked to "show context" for the first two sentences.

- For *Explanation-Cause*, provides information about probable causes or contributors to the question Focus.
- For *Explanation-Effect* questions, provides information about probable effects or results of the question Focus.
- For *Comparison* questions, provides information helpful to comparing the Topics with regard to the Focus.

2. **Topic Only** The sentence discusses the correct Topic (or, in the case of Comparisons, at least one of the two Topics) but does not provide causal/contrastive information

with respect to the Focus.

3. **Not Relevant** The sentence is not about the Topic whatsoever.

4. **Ambiguous** The correct judgment is not clear because of lack of context or other reasons. (If this option is selected, the user is prompted to enter a brief explanation, as shown in Figure 5.4.)

In order to measure the specific impact of related term expansion using the RSR-based and other techniques, we evaluate system responses generated using three distinct settings for Related Term Generation module, with each setting using a different subset of lexical expansion methods as follows:

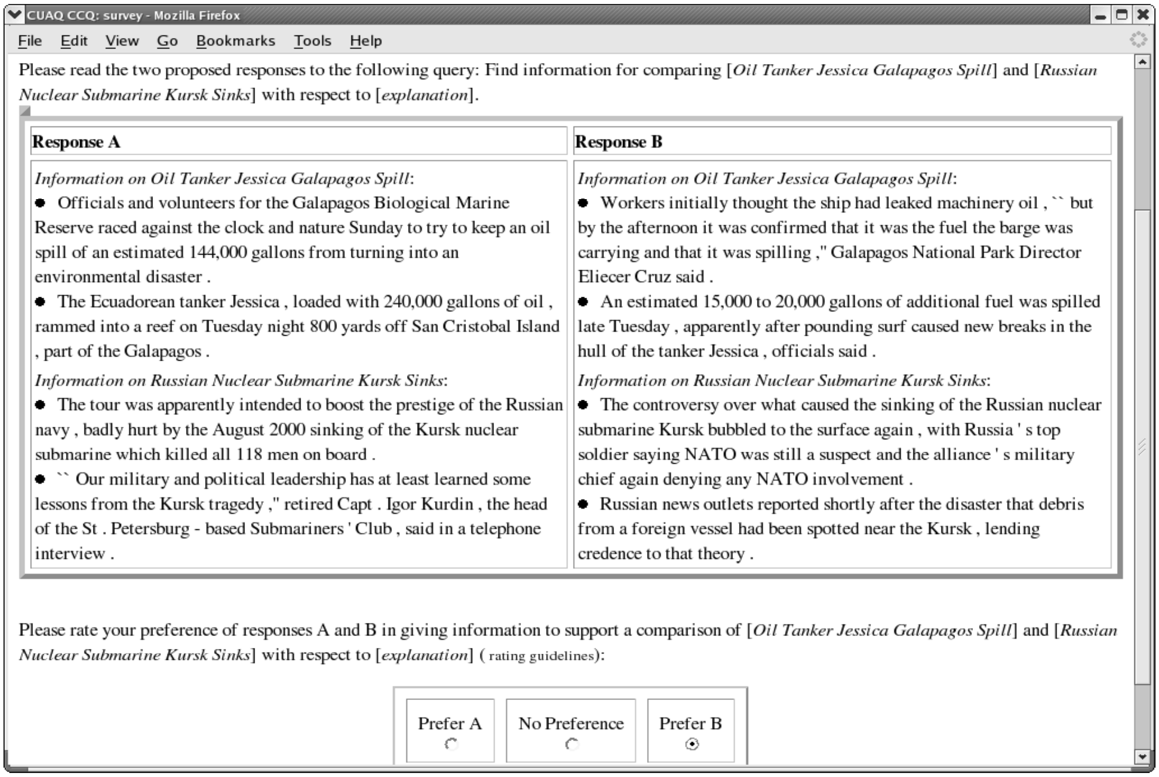
Baseline Using Topic(s) and Focus question terms and static Cue terms only.

Semantic Adding WordNet and LSA-derived terms on top of Baseline.

Semantic+RSR Adding RSR-derived terms on top of Semantic (i.e., using all methods).

Before running our evaluation, however, we had several colleagues give us informal feedback on a pilot version of the instructions and experimental interface (loaded with development questions only). The feedback we received led us to make several key changes to our evaluation setup.

The foremost issue was that our colleagues found it to be difficult to make confident judgments for Comparison answer sentences at the individual-sentence level. More specifically, when "comparable" sentences are automatically matched as in Figure 5.3, they felt that they could make a judgment with regard to such a sentence pair. Yet it was more difficult to make a judgment on the "comparability" of a single sentence. While this feedback points up the importance of pursuing like-sentence matching in future work, we still felt that we did not want our evaluation to essentially focus on the matching task (which we implement with a straightforward cosine distance lexical overlap measure), since we are more concerned with whether our RSR-based techniques truly help to find comparison-linked language for a given question Focus. Thus, we decide to change the format for our Comparison question evaluation to concentrate on this point.


CUAQ CCQ: survey - Mozilla Firefox
File Edit View Go Bookmarks Tools Help

Please read the two proposed responses to the following query: Find information for comparing [*Oil Tanker Jessica Galapagos Spill*] and [*Russian Nuclear Submarine Kursk Sinks*] with respect to [*explanation*].

Response A	**Response B**
Information on Oil Tanker Jessica Galapagos Spill: ● Officials and volunteers for the Galapagos Biological Marine Reserve raced against the clock and nature Sunday to try to keep an oil spill of an estimated 144,000 gallons from turning into an environmental disaster . ● The Ecuadorean tanker Jessica , loaded with 240,000 gallons of oil , rammed into a reef on Tuesday night 800 yards off San Cristobal Island , part of the Galapagos . *Information on Russian Nuclear Submarine Kursk Sinks*: ● The tour was apparently intended to boost the prestige of the Russian navy , badly hurt by the August 2000 sinking of the Kursk nuclear submarine which killed all 118 men on board . ● `` Our military and political leadership has at least learned some lessons from the Kursk tragedy ," retired Capt . Igor Kurdin , the head of the St . Petersburg - based Submariners ' Club , said in a telephone interview .	*Information on Oil Tanker Jessica Galapagos Spill*: ● Workers initially thought the ship had leaked machinery oil , `` but by the afternoon it was confirmed that it was the fuel the barge was carrying and that it was spilling ," Galapagos National Park Director Eliecer Cruz said . ● An estimated 15,000 to 20,000 gallons of additional fuel was spilled late Tuesday , apparently after pounding surf caused new breaks in the hull of the tanker Jessica , officials said . *Information on Russian Nuclear Submarine Kursk Sinks*: ● The controversy over what caused the sinking of the Russian nuclear submarine Kursk bubbled to the surface again , with Russia ' s top soldier saying NATO was still a suspect and the alliance ' s military chief again denying any NATO involvement . ● Russian news outlets reported shortly after the disaster that debris from a foreign vessel had been spotted near the Kursk , lending credence to that theory .

Please rate your preference of responses A and B in giving information to support a comparison of [*Oil Tanker Jessica Galapagos Spill*] and [*Russian Nuclear Submarine Kursk Sinks*] with respect to [*explanation*] (rating guidelines):

Prefer A ○ No Preference ○ Prefer B ●

Figure 5.5: Example evaluation screen for a user rating a Comparison question.

In particular, we purposely choose *not* to present "matched" information in the evaluation interface, even when when we do identify it. Rather, we present two sentences about each question Topic, and explain in the survey instructions that these sentences are meant to *support* a comparison of the two topics, rather than to *be* a comparison. We then ask for a user's preference with regard to the former, wording the question carefully: "Given the task of creating a comparison of the two topics with respect to the focus, which response provides more useful information?" Figure 5.5 shows a sample evaluation screen.

Again, the idea here is to concentrate the evaluation on whether we are successfully finding the comparison-linked language, rather than the matching of statements which share that language across both Topics. In some sense this goes to a key assumption with regard to these questions, namely that information can be comparative, even if it does not result in a point-for-point comparison, and in that sense leaves some of the work of the comparison task to the user. For instance, in the right-hand side of Figure 5.5, we do not get a point-by-point comparison of the explanations for the two seafaring disasters. Yet, our RSR model identifies terms like *thought, officials, caused, controversy* and *deny* as relevant when making comparisons with respect to *explanations*. Thus, while we do not explicitly draw the comparison in the presentation of our answer, we provide comparative information which would allow the user to formulate a statement like, "In the Jessica disaster, officials said that pounding surf caused additional oil to spill, whereas in the Kursk sinking, Russian officials blamed a collision with a NATO vessel (which NATO officials denied)." Of course, in other cases, we may find comparative information which is more obviously matched; the point is not to over-emphasize the automatic matching aspect of our system and to let users decide whether useful comparisons can be made based on the information we find.

While the initial feedback motivated us to evaluate Comparison responses holistically, our colleagues reported that for Explanation questions, sentence-level relevance judgments were much easier to make. In addition, they told us that having the original-document context of an answer sentence available (as pictured for the first two sentences in Figure 5.4) via an expandable "show context" option was often useful in increasing judgment confidence and avoiding unnecessary Ambiguous judgments.

The other common feedback we received was that it was difficult to keep two kinds of questions in mind and switch back and forth between instructions/contexts for evaluating the two types. Thus in our final evaluation study, we assigned individual users to rate only one response type, either Explanations or Comparisons.

We show examples of the final version of the evaluation interface for each question type in Figures 5.4 and 5.5.

5.7.3 Explanation Questions

For each of the 20 test questions in our test set, we compare answers from our CCQ under six distinct conditions. These six response conditions result from using three different settings for Related Term Generation and two settings for Document Retrieval, or $3 \times 2 = 6$ distinct conditions.

The three settings we evaluate for Related Term Generation are the Baseline, Semantic, and Semantic+RSR settings described in the previous subsection.

The two settings we use for Document Retrieval, are the Oracle and Open settings.

Finally, in all cases, responses are four sentences long.

We collect human judgments from 16 judges overall, each of whom rated 12 answers. This means that each of the 120 total responses (6 conditions × 20 questions) was rated by at least one judge, and 72 were rated by two judges. The judges in our evaluation are primarily graduate students in computer science. None of them was briefed on the goals or methods of our experiment other than as described in the survey instructions. Appendix C lists the survey instructions precisely as given to subjects.

5.7.3.1 Interannotator Agreement

We first analyze the data to determine how closely our judges agree on the categories to which they assign individual answer sentences. This is especially important given the exploratory nature of our questions, as a window to understanding whether the thing which our system is meant to identify, i.e. information which helps answer Explanation questions, can be reliably recognized by people.

To calculate interannotator agreement, we use Cohen's Kappa statistic κ, which is commonly used to calculate the degree to which annotator agreement exceeds chance agreement (Carletta, 1996). Kappa is calculated as:

$$\kappa = \frac{P_a - P_e}{1 - P_e}$$

Where P_a is the average agreement among annotators and P_e is the expected chance agreement. This statistic has a value of 1.0 under exact agreement; a value at or near 0.0 is expected under complete independence. The interpretation of what value indicates "significant" agreement is more debated. Di Eugenio's (Di Eugenio, 2000) informative discussion explains how a number of different standards for interpreting Kappa have been used. As she notes, Krippendorff's (Krippendorff, 1980) standard is the strictest, discounting values below .67, assigning values above .67 and below .80 "moderate" reliability, and values of .80 and above "high" reliability. But she points out that more forgiving standards, which consider values as low as .41 acceptable, may be applicable depending on various factors. One such factor which she recognizes is whether the probability of chance occurrence, P_e, factors in distribution skew. This practice, which we do follow, can make high κ values harder to achieve. However, calculating in this way seems more sound given that annotator agreement can otherwise appear falsely high if we do not recognize that in the aggregate, some categories may be chosen more frequently than others.

We find that over the 72 responses rated by two judges, we have actual agreement $P_a = 0.729$ versus chance agreement of $P_e = 0.399$, for $\kappa = 0.549$. Note that chance agreement is not simply .25 because the calculation we use takes account of the fact that the distribution of judgments over categories is uneven. Thus we can say we have agreement well above what would be expected by chance, although we only approach the stricter reliability measures for interannotator agreement.

Looking more closely at the judgments, we find that a substantial number of the disagreements concern sentences which one rater has marked "Ambiguous" and the other as something else. This happens in 19 of 288 total cases (72 double-marked responses × 4 sentences per answer). For example, for the question, "Describe [effects] of [publishing] in [Tiananmen Papers Published]", one answer contains the following sentence:

> Peter Osnos, publisher and chief executive of Public Affairs, described the book's author as a "compiler who had been specifically sent abroad with material to find a way to make it available."

One annotator rates this sentence with "Topic Only" relevance, possibly having used the "show context" option. The context of the sentence does not explicitly say that the "book" in question is the "Tiananmen Papers" book, but could be argued to make that interpretation highly likely (the answer sentence falls between these two sentences where we have marked bracketed ellipsis):

> The book's author and his associates, they say, hope the documents will spurt debate on reform by embarrassing hard-liners who encouraged the use of violence. [...] Osnos said a Chinese language compilation of the documents would be published in a few months.

However, the second annotator applies an Ambiguous rating, adding the note "the context doesn't determine if the sentence is on topic or not The Book may refer to Tiananme"

Term Generation (Document Retrieval)	**Ambiguous**	**Not Relevant**	**Topic Only**	**Relevant**
Baseline (Oracle)	0.12	0.16	0.42	0.30
Semantic (Oracle	0.03	0.20	0.38	0.39
Semantic+RSR (Oracle)	0.06	0.17	0.45	0.47
Baseline (Open)	0.16	0.33	0.30	0.21
Semantic (Open)	0.12	0.29	0.36	0.23
Semantic+RSR (Open)	0.11	0.28	0.38	0.23

Table 5.8: Proportion of answer sentences rated in each category for Explanation responses grouped by Term Generation Method and Document Retrieval mode. (Each individual answer is four sentences long; their are 20 test questions.)

[*sic*]. While we do attempt in our instructions to explicitly encourage users to assume topic relevance in cases where it seems "substantially more likely than not," this kind of disagreement is at some level inevitable in a subjective evaluation.

Issues such as this leave us slightly short of the interannotator agreement levels we would like. Still, the level of agreement we do achieve is certainly well above random, and within commonly used ranges for "moderate agreement" as discussed above. Furthermore, we believe that more iterations of the instructions and a larger data set might increase agreement to meet the more stringent reliability standards proposed by some.

5.7.3.2 Comparing Methods

Once we establish the relative reliability of our categories, we proceed to examine the impact of the parameters used in creating a response on the relevance judgments assigned to that response.

By examining the proportion of answer sentences in each category assigned to each category, we can quickly observe that there is a strong impact based on the Document Retrieval mode used. Table 5.8 summarizes these proportions. (For purposes of this table, we combine ratings for questions where there are disagreements, we simply assign a "half" frequency to each category for those sentences.) As we would expect, there are substantially more Not Relevant judgments when using the Open IR method, as well as an increase in the proportion of Ambiguous judgments. Based on this observation, we examine the two cases separately.

In the case of **Oracle** mode Document Retrieval, each document retrieved by CCQ has been judged by a TDT annotator to contain information relevant to the TDT topic which we use as our question Topic. However, TDT judgments of topicality are made at the document level. Thus, we often have only a small number of sentences in a longer article which are topic-relevant. For instance, for the topic "2000 Nobel Peace Prize", many of the relevant articles discuss all of the Nobel prizes awarded in that year, so that there may be relatively few sentences which actually mention the peace prize. This helps account for the fact that even in the Oracle setting, a non-trivial number of sentences (between 15 and 20 percent) are rated as Not Relevant, i.e. not about the Topic whatsoever.

Nonetheless, we feel that the key challenge for our system when using Oracle mode is to find sentences which express cause/effect information with respect to the Topic. In our collected judgments, the success of the system in doing this is reflected by the number of Relevant sentences, not simply sentences which are about the topic (i.e. judged as Topic Only). Thus, we evaluate the difference between Oracle configurations based on their success in finding sentences judged as Relevant. To that end, we score each configuration by how many of its response sentences are rated as Relevant by our judge(s) (again, we combine ratings for response sentences where two judges disagree, assigning a "half" point if one of two judges rates a sentence as Relevant). Figure 5.6 shows three such answers and their associated scores. (The mean scores using this calculation are: Baseline, 1.2; Semantic, 1.5; Semantic+RSR 1.875.)

We can use the scores for the responses as the dependent variable in a one-way ANOVA test for correlated samples (Lowry, 2006), where the independent variable is the Term Generation method used (Baseline, Semantic or Semantic+RSR). In this analysis, we have three correlated samples resulting from the response to each question by each of the three Term Generation methods, and each response has a score as explained above. Using the ANOVA test, we find that the Term Generation method used has a significant effect on the score with $P < .05$. However, this tells us only that the effect is significant in the

Method (Score)	Answer Text
Semantic + RSR (3.5)	USA Engage, one such group in Washington, estimates that sanctions cost $15 billion to $19 billion in lost exports each year and that American companies employ 200,000 fewer people in high-wage, export-oriented jobs because of lost sales. (**R,R**) President Clinton began lifting trade and economic sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade. (**R,X**) Sanctions almost certainly contributed to the fall of Slobodan Milosevic, who had survived the loss of four civil wars and a bombing by NATO. (**R,R**) But the Yugoslavia case could plausibly be remembered as the most significant success for economic sanctions since the fall of the apartheid regime in South Africa more than a decade ago. (**R,R**)
Semantic (3)	The sanctions also included restrictions on air travel and oil sales to Yugoslavia, though both of those sets were eased even before Milosevic's fall. (**R**) The NATO allies imposed a variety of sanctions on Yugoslavia beginning in 1992, during the war in Bosnia. (**X**) President Clinton began lifting trade and economic sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade. (**X**) Sanctions almost certainly contributed to the fall of Slobodan Milosevic, who had survived the loss of four civil wars and a bombing by NATO. (**R**)
Baseline (2)	President Clinton began lifting trade and economic sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade. (**R**) The NATO allies imposed a variety of sanctions on Yugoslavia beginning in 1992, during the war in Bosnia. (**X**) The sanctions also included restrictions on air travel and oil sales to Yugoslavia, though both of those sets were eased even before Milosevic's fall. (**R**) However, other sanctions were imposed again in 1998 after Milosevic launched a brutal crackdown on ethnic Albanians in Kosovo. (**X**)

Figure 5.6: Three answers from our evaluation for the question, "Describe effects of [sanctions] in [Clinton Lifts Sanctions Against Yugoslavia]," and their associated scores, as described in Section 5.7.3.2. All answers use Oracle Document Retrieval, but differ as indicated in the methods used for Related Term Generation. The annotation(s) after each sentence indicate whether the human judge(s) marked a sentence Relevant (R) or not (X).

aggregate, not that any individual configuration has a higher rank than any other. To analyze whether the pairwise differences in score are significant, we can use the Tukey post-ANOVA test (Lowry, 2006). Using Tukey, we find that Semantic+RSR scores significantly higher than Semantic alone ($P < .05$), and that both Semantic and Semantic+RSR score significantly higher than Baseline ($P < .05$).

With **Open** mode document retrieval, we see in the bottom three rows of Table 5.8 that the system finds fewer Relevant sentences in all configurations, and that the differences in proportion of Relevant sentences at the aggregate level is greatly reduced. Performing the same ANOVA analysis just discussed confirms that there is no significant effect on Term Generation method on Relevant sentence score in this case.

However, if we consider that in this task there is a greater difficulty in simply finding Topic-relevant sentences, we can analyze this case slightly differently. Whereas earlier we score a response based on the number of judged Relevant sentences, we instead compute the score based on the number of sentences judged Relevant *or* Topic Only (using the same "half-point" procedure for response sentences with two different judgments). The intuitive reason for considering this score is that even when the Semantic/RSR methods do not find cause or effect information, they may sometimes guide an answer toward Topic-relevant sentences. In fact, in terms of pure efficiency, the Semantic and Semantic+RSR methods actually are less efficient than the Baseline at finding Relevant sentences from the total number of Relevant and Topic Only sentences rated, but we hypothesize that this reflects that while these more sophisticated methods are able to reach deeper for recall of sentences which are Topic Only, it is simply more difficult to find Relevant sentences.

Using this broader scoring metric, we do find with an ANOVA test that there is is significant effect on score depending on method ($P < .05$). However, the Tukey post test finds that while the Semantic and Semantic+RSR methods both clearly have a higher mean than Baseline, that there is no significant pairwise difference between these methods and the Baseline.[16] In addition, there is no significant pairwise difference between Semantic and

[16] Looking closely at the ANOVA data, we see that this is largely because of the high variance in these scores, which makes significance harder to achieve. We attribute the high variance in part to the increased unevenness in document quality introduced by using Open retrieval.

Semantic+RSR methods.

5.7.4 Comparison Questions

As we describe earlier in this section, feedback from several colleagues indicated that responses for Comparison questions can be difficult to judge at the sentence level. On this basis, we reconfigured the evaluation of these questions to use a preference-based, side-by-side protocol, shown in Figure 5.5.

Another difference with the Explanation question evaluation is that we only evaluate Comparison responses generated using Oracle Document Retrieval settings. The reason for this decision is primarily practical, owing to the difficulty in collecting a large number of user judgments.

Therefore, for each of the 10 Comparison questions in our test set, we compare CCQ answers under three distinct conditions, namely the three settings for Related Term Generation described earlier, Baseline, Semantic and Semantic RSR. As shown in Figure 5.5, the responses presented for judgment are always four sentences in length, using two sentences to describe comparative facets of each question Topic.

We collect human judgments from six judges overall, each of whom rated ten answer pairs. For each question, there are $\binom{3}{2} = 3$ unique pairs of answers for which to collect preference judgments, or 30 pairs overall. Thus we collect two judgments for each answer pair.

5.7.4.1 Interannotator Agreement

As we do for Explanation questions, we begin our data analysis by assessing the interannotator agreement of our judges. Again, we use Cohen's Kappa to measure agreement. In this case, we have chance agreement $P_e = 0.346$ and actual agreement of $P_a = 0.650$, and $\kappa = 0.464$. As described earlier, this is within the range of the more lenient standards for moderate reliability in κ, although considerably below stricter standards.

5.7.4.2 Comparing Methods

In order to compare the frequency with which the different methods result in a user-preferred response, we transform these pairwise preference judgments into scores as follows. For each Comparison question, we take the six pairwise judgments for the question (2 annotators × 3 responses), and assign a "point" to a given response for each judgment which prefers it. For example, for the question "Compare [Kobe charged with sexual assault] and [Edmond Pope Convicted for Espionage in Russia] with respect to [prosecution]," the Semantic and Semantic+RSR methods create the same answer, and the Baseline answer is slightly different (Figure 5.7, shows the two answers in the same style as they are shown in the evaluation interface; see Figure 5.5 for an actual screenshot of the interface). Both raters prefer the Semantic/Semantic+RSR answer over the baseline, but (unsurprisingly) rate the identical Semantic/Semantic+RSR answers as equal. Thus, for this response, the Baseline gets a zero score, whereas the Semantic and Semantic+RSR methods each get two points (one from each of the two judges who prefer it over the Baseline response).

The mean scores are, for Baseline 1.4, Semantic 2.1, and Semantic+RSR 2.1 (the fact that Semantic and Semantic+RSR have the same mean score is not because they always produce the same answer, although this happens in three of the ten test questions). Using these scores, we can apply Friedman's test for correlated samples, which is appropriate in this case rather than ANOVA because the input scores are not interval valued but transformed from preference judgments (Lowry, 2006). We find by Friedman's test that the differences in preference scores are statistically significant ($P < .05$). However, it is clearly the case (and we use post-hoc Tukey tests to confirm) that the pairwise differences are only significant between the Baseline and Semantic methods, and Baseline and Semantic+RSR methods.

5.8 Observations

In addition to the evaluation-based quantitative analysis in the previous section, we also have several qualitative observations about our work in answering Explanation and Comparison questions with the CCQ system.

Baseline Answer	**Semantic and Semantic+RSR Answer**
Information on: [Kobe charged with sexual assault]	*Information on: [Kobe charged with sexual assault]*
• The judge hearing sexual assault charges against Kobe Bryant issued an unorthodox order on Tuesday aimed at protecting the identity of Bryant’s accuser. • LOS ANGELES - Regardless of the outcome of his criminal case , Lakers star Kobe Bryant could foul out in the endorsement world after Friday ’ s sexual-assault charge and acknowledgment he has committed adultery.	• Now he faces a Class 3 felony sexual assault charge that carries a possible sentence of four years to life if he is convicted. • LOS ANGELES - Regardless of the outcome of his criminal case , Lakers star Kobe Bryant could foul out in the endorsement world after Friday ’ s sexual-assault charge and acknowledgment he has committed adultery.
Information on: [Edmond Pope Convicted for Espionage in Russia]	*Information on: [Edmond Pope Convicted for Espionage in Russia]*
• And Pope’s main accuser has recanted his testimony . • The verdict marked the first espionage conviction against a Westerner in Russia in the decade since the end of the Cold War , though both Moscow and Washington have expelled alleged spies from each other ’ s capitals in recent years .	• If convicted in a secret trial , he could be sentenced to up to 20 years in prison. • The verdict marked the first espionage conviction against a Westerner in Russia in the decade since the end of the Cold War , though both Moscow and Washington have expelled alleged spies from each other ’ s capitals in recent years .

Figure 5.7: Responses for the Comparison question, “Find information for comparing [Kobe charged with sexual assault] and [Edmond Pope Convicted for Espionage in Russia] with respect to [prosecution],” using Baseline versus Semantic/Semantic+RSR Term Generation (the latter two methods create the same answer in this case).

5.8.1 Term Suggestion Quality

When we qualitatively examine the kinds of related terms generated by our RSR models, we notice several recurring phenomena in the filtering terms. For instance, consider the question focus *losing games* from our question "Describe [causes] of [losing games] in [Australian Open Tennis Championship]". Table 5.9 shows the top terms suggested from our Cause model for this Explanation-Cause question, as well as the terms which would be generated for an Explanation-Effect question with the same focus. The table shows the terms, along with the associated $-2 \log \lambda$ confidence score (recall that the cutoff for this score is 10.0, so that these terms would all be generated and used in CCQ). We find here, as in many cases, that at least four phenomena typically emerge in the filtering terms.

The first are terms which seem intuitively "correct", i.e. having the desired relation with the Focus. For causes, these would include the various stems relating to injury/illness (e.g., *flu, tendon, viral*), while for effects these appear to include more emotionally-oriented effects (e.g., *shame, sad, angri*).

Second are terms which seem to result from the issue from alternate word senses for one or more of the component terms of our Focus. For instance, the stems *piraci* and *copyright* very likely derive from another sense of *losing* and *games*, namely that software piracy can result in lost sales of video games.

A third phenomenon is the presence of modifier or hedging terms which appear frequently in expressions of causality, but seem less intuitively correct because their bond with the Focus is less tight. In this example, we see a number of such terms in the Explanation-Effect list, namely *partli, mainli, primarili, actual, mostli,*. This can partly be understood as a reflection of the way in which our model includes both rhetorical and semantic aspects of usage, with these indicating the manner in which causality may be expressed rhetorically. We see in this particular example that these hedging terms are observed more frequently with the effect of this particular event than with the cause. Perhaps this indicates less hedging when ascribing blame for a lost game, and more when talking about issues such as resulting shame or anger, which people may discuss more reluctantly.

The last phenomenon which is frequently seen, of course, is simple noise. While many apparently "bad" suggestions can be fairly easily explained based on polysemy or other more typical linguistic issues, we sometimes see related terms whose presence is less clearly explicable. While the lists in Table 5.9 are fairly clean, if we look at another example we can see that this is not always the case.

Explanation-Cause	**Score**	**Explanation-Effect**	**Score**
flu	87.8	partli	95.2
infect	86.0	reluct	71.4
achil	81.6	mainli	59.6
conflict	77.2	primarili	59.2
abdomin	74.8	nervou	58.5
inflamm	67.4	simpli	54.2
stomach	66.8	oppos	52.4
piraci	64.9	shame	48.0
academ	64.6	signific	46.9
tendon	62.8	hate	42.0
frozen	61.4	sad	39.2
spasm	60.8	hurt	38.1
closur	54.1	anxieti	37.5
linger	53.1	actual	36.8
unrest	51.5	mostli	35.6
viral	51.5	gambl	35.3
scandal	50.2	avoid	34.8
chronic	49.5	postpon	34.5
stiff	48.1	angri	33.2
afford	47.3	analyst	31.3
nag	47.0	panic	30.4
merci	45.5	inelig	29.7
fog	45.5	sorri	29.6
ill	45.5	tell	29.5
delai	45.4	shouldn	28.5
tire	44.8	hesit	28.4

Table 5.9: Top-ranking RSR-generated related terms for the question focus *losing games* in Explanation-Cause and Explanation-Effect settings.

Table 5.10 shows the suggested terms from our Contrast model for the question focus *injuries death casualties* from our question "Compare [Popocatepetl Volcano Erupts] and [Earthquake in India Gujarat State] with respect to [injuries deaths casualties] ."

Contrast Term	**Score**	**Contrast Term**	**Score**
figur	52.3	exact	23.3
jolt	51.9	success	23.0
confirm	49.3	actual	22.2
acknowledg	44.3	preliminari	21.9
shook	43.6	bombard	21.2
rock	40.4	earnhardt	20.8
richter	39.6	number	20.7
scale	38.6	autopsi	19.9
count	35.3	necessarili	19.6
measur	32.4	smoke	19.2
rubbl	28.2	griev	19.1
foul	26.4	inevit	19.1
gunfir	24.8	put	18.8
moder	24.7	uncertain	17.8
bodi	23.4	plenti	17.8

Table 5.10: Top-ranking RSR terms for a Comparison question with Focus *injuries deaths casualties*.

If we examine the terms in this table, we recognize some of the same phenomena as in the Cause-suggested terms. In particular, some terms seem to indeed be intuitively correct for comparing injury and death toll. These include terms which deal with the tally in relative or exact terms (*figur, count, measur, moder, exact, preliminari, number*), as well as terms used to compare the manner of deaths (*rubbl, gunfir, autopsi, smoke*).

While polysemy appears to be less of an issue here, we see here that we have another kind of more surprising noise which we sometimes observe as resulting from our model's inability to distinguish between specific instances versus more generally Focus-relation linking terms. Specifically, the suggestion *earnhardt* jumps out from the list as rather odd. The explanation for this term's appearance is that Dale Earnhardt was a race car driver who died in a highly publicized racing accident which took place in the same timeframe in which our training corpus articles were collected. If we go back to our training set, we find extracted Contrast sentences such as:

> Earnhardt was the fourth driver to die in less than nine months, but his death will make the greatest impact.

While we remove the duplication of sentences inherent in newswire stories, nonetheless the coverage of this particular death was sufficient that our model believes that *earnhardt* is a likely term to use when making comparisons about *injuries, death* and *casualties.* Yet while this type of "instance-specific" suggestion may degrade the generality of our generated term lists, they are not generally problematic in our current CCQ implementation when used as post-Document Retrieval filtering terms, because there is little chance of actually encountering such a rare term when comparing death tolls in, e.g., two natural disasters. Nonetheless, it does suggest that identifying instance-specific suggestions, perhaps using logic which accounts for frequencies which are distribute over a short time period, is an interesting direction for future work.

Another phenomenon which we observe in this and other Comparison cases gets back to the combination of rhetorical and semantic usages which our model captures. In Comparison questions, we generally are more interested in capturing the rhetorical usages, inasmuch as they tell us about language used in the context of making comparative or contrastive statements with regard to our Focus. In the *injuries death casualties* example, these would include the suggested terms mentioned above which are used in assessments of the number of dead or injured people and the manner of death. Alternately, we sometimes generate suggestions which are more semantic in the sense of antonyms or sibling terms; for instance, if we use the seed term *dead*, the top related term by the Contrast model is *alive.*

In most of the cases in our test set of Comparison questions, we primarily are interested in the more rhetorical aspects of relations. This is in part because of the eventive nature of TDT topics; in these cases, we want the Contrast RSR model to tell us, in essence, what kinds of terminology people use when discussing, e.g. *prosecutions*, or *diplomatic strategy*, or *achievements.* However, the more semantic side of this relation could be equally interesting in Comparison questions which are formulated differently. For instance, if we were to ask a question comparing the *strength* of two materials, suggestions like *weak* or *fine*, are likely to be useful, even if they fall more on the semantic side. In the end, our model does not distinguish between rhetorical and semantic suggestions, and the distinction can be both subtle and subjective. Still, we mention it here for intrinsic interest and to connect this application back to the more theoretical discussion of our relation models' meaning from Chapter 3.

5.8.2 Term-System Interaction

While the above discussion considers the intrinsic qualities of our RSR model, we ultimately evaluate the impact of these terms on the end-to-end CCQ system. Thus, we also make several observations on the ways in which these term suggestions interact with the CCQ pipeline as a whole.

The primary issue in this regard is probably the way in which we integrate the term generation module in the system currently, namely as a post-document retrieval step. A key advantage of this approach is that it limits any negative effect on document-retrieval precision which might result from some of the term generation phenomena discussed above, e.g. instance-specific term suggestions. Thus, for instance, our Open mode answer for the question: "Compare [Popocatepetl Volcano Erupts] and [Earthquake in India Gujarat State] with respect to [injuries deaths casualties] ." does not retrieve documents about Dale Earnhardt. Then, when our Contrast model generates *earnhardt* as a related term to try and include when discussing issues of *death*, we suffer no harm.

On the other hand, our answers can also suffer as a result of this conservative approach to using our RSR-derived terms. As noted in the previous section, we derive a number of reasonable Explanation-Cause terms for our question, "Describe [causes] of [losing games] in [Australian Open Tennis Championship]." But in our answer for both the Open and Oracle settings, our answer does not reflect this. The left-hand side of Figure 5.8.2 displays the answer produced in our Oracle setting, which contains zero relevant sentences according to our judges.

One reason for this unsatisfactory answer is that the documents returned by Oracle, even though guaranteed to be on-Topic, simply do not contain a high proportion of the Cause-related terms which our model suggests. Of course, there are other issues which lead to our problems, of course, such that the weightings which we use in Answer Creation allow the wrong sentences to be chosen here. Some LSA terms which are only marginally

Test Answer, Oracle Document Retrieval	Answer using modified Open retrieval with RSR keywords
The only unseeded champ in the Open era was Chris O'Neil, who won the Australian Open in 1978, when many top *players* skipped the tournament. The women's wild cards were announced Monday, with six Australians receiving *berths* in the 128-*player* singles draw for the Grand Slam event that begins Jan. 15. Agassi, seeded sixth, needed just 1 hour, 40 minutes to beat Czech *player* Jiri Vanek 6-0, 7-5, 6-3. Rain **delayed** the start of the match by half an hour while the court was dried and the roof closed on Rod Laver Arena.	Safin has been cursed by **injuries** in recent months and had to withdraw from the final of the Barcelona tournament last month with a **stomach** strain. Pete Sampras won **back**-to-**back** *championships* in 1993 and '94, while Todd Martin won it in 1996 and again in 1999. The holder of 29 WTA titles, Venus has seen a drop in form of late culminating in a pulled a **stomach** muscle before the German Open earlier this month. Last year a serious **knee ligament injury** almost forced her out of the game.

Figure 5.8: Two answers for the test question, "Describe [causes] of [losing games] in [Australian Open Tennis Championship]," with RSR-suggested terms in **bold**, LSA-suggested terms in *italics*. The answer to the left is our test answer as produced under Oracle IR; the answer to the right uses the modified Open IR as described Section 5.8.2.

related to the question focus are problematic; some sentences with high centroid-score but few related terms are chosen; our one RSR term (*delay*) does not help us because it cites delay in a way which is not clearly related to a loss.

Nonetheless, we consider a post-hoc experiment to increase the proportion of RSR-suggested terms in input documents. Namely, we include some of our RSR terms in a modified Open-mode Document Retrieval query, so that we can actively skew our input documents toward inclusion of likely answer terms.

In a post-hoc experiment, we find that in this particular case, using the top 40 suggested terms from our RSR model as keywords in the document search (using Indri query operators to weight these terms below the other query terms, and treat them as "synonyms" to reduce impact of the length of the list) yields a more satisfactory Open mode answer, shown on the right side of Figure 5.8.

Of course, this answer is not perfect either. The first and third sentences are about other tournaments (a typical downside of using Open mode retrieval). And the second sentence falls victim to the polysemous nature of *back* (which our model identifies as a cause of losing games, but presumably in the anatomical word sense). Nonetheless, we feel that using our RSR model-suggested terms for directing document retrieval, and not only once retrieval has been done, is a future direction which we wish to explore. We do note, however, that there are significant potential downsides as well, most prominently the precision problems which can hinder any query-time lexical expansion approach.

5.9 Conclusions and Future Work

The experiments reported in this chapter are the most forward-looking in the thesis. We pursue an ambitious and exploratory agenda with regard to our "relation-focused" questions, and in particular Explanation and Comparison questions, presenting a framework, implementation and set of experiments. We are satisfied that our work has provided interesting and informative results across a breadth of issues in this new area, even while there is significant further work to be done.

In terms of framework, our contribution in this chapter centers on the proposal of

Explanation and Comparison questions as two key instances of a broader class of relation-focused questions. We use a template format to constrain the problem space and enforce a level of structure on the kinds of questions which can fall under these two types. Moreover, we consider these questions in the long-answer model, whereby answers are expected to be made up of multiple sentences which provide a comprehensive answer, rather than a short factoid or snippet.

Our implementation of the CCQ extension to DefScriber is perhaps the largest contribution of the chapter. At the system level, we implement new modules for the Query Input, Document Retrieval, Related Term Generation and Sentence Scoring pieces of our pipeline, attaching these components to DefScriber's bottom-up answer creation modules on the back end, as well as implementing a Web interface for ad hoc experimentation. The most innovative feature of these new modules is the integration of a Related Term Generation module which derives terms which are linked to an Explanation or Contrast question by our RSR models.

This application of our RSR models differs from that used for answer coherence explored in the previous chapter, which use the classification paradigm to assess the relational "fit" between two spans of input text. Instead, we generate from our model the specific terms which are most strongly linked to a question "Focus" by a relation of interest. This allows us to use our RSR knowledge more efficiently than in the classification-based scheme. Whereas in that scheme, we would be faced with running our classifier to compare a question Focus with each sentence of interest, we can now explicitly seek out sentences, or even documents, containing terms which our models suggest as strong indicators of relational linkage.

The first important finding of our evaluation experiments is that human annotators are able to agree with moderate reliability on the relevance of individual sentences in Explanation answers. In informal pre-evaluation feedback, we discovered that this agreement was likely to be less strong for Comparison questions, and thus designed an evaluation for these questions which merely seeks judgments of user preference; nonetheless, these judgments have interannotator agreement at the low end of acceptable reliability.

In terms of the quality of responses we produce, we demonstrate that our RSR models can significantly improve the quality of Explanation answers under ideal input conditions.

However, under other conditions, and for Comparison questions, RSR terms do not significantly improve performance over response techniques which use other Semantic methods, such as LSA, to identify related terms.

An important limitation of our results is that we only achieve significant improvement from using our RSR models when using a gold-standard "Oracle" document retrieval process. This clearly points to the need for further research in identifying relevant documents for questions of this type. However, we also note that this limitation, while real, is not unique to the evaluation task we design. For instance, in evaluations such as DUC and MUC, the task centers on extraction of critical information of interest, and systems are provided as input a set of documents of interest (Sundheim, 1992; Dang, 2005).

There are many potential areas for future work. One intriguing possibility is the integration of syntactic or frame-semantic analysis at the sentence level for identifying relevant answer material. In this sense, our approach has thus far been *content*-focused, used only a shallow statistical knowledge model, relying on automatically extracted RSR models to determine likely answer terms. Syntactic and frame-based approaches are very much complementary, employing various *form*-focused methods which we discuss in the Related work section (Girju, 2003; Bethard, 2005; Verberne et al., 2007). In future work, we would like to combine these approaches in a hybrid method that uses syntactic- or semantic role-based analysis to identify likely answer form, and uses the techniques which we consider in this chapter to identify likely answer content from probabilistic relation models. For instance, Girju's method for determining whether a verb like *produce* is being used causally could be combined with knowledge in our RSR model to determine whether one of the arguments of the verb has causal links with a given question.

A second important area involves research into the nature of Comparison and Explanation question themselves. In determining the kinds of questions which our CCQ system supports, as well as in coming up with an evaluation set of questions, we are motivated practically to make a number of simplifications. In particular, we use question templates rather than an open question format, and use TDT-derived topics as bases for our evaluation questions. Yet within these simplifications, whole areas of research are contained; these include question parsing for determining which questions can and/or should be treated as

Explanation or Comparison, as well as an empirical study regarding the topics about which questions "real" users would like to be able to ask. In their recent work, Verberne et al. have made inroads in these areas, collecting a corpus of *why*-questions (Verberne et al., 2006) as well as implementing a question classifier to distinguish among subtypes of these questions (Verberne, 2006).

In this same vein, another interesting study would be to elicit user questions to determine which other kinds of relation-focused questions are of interest. We could use such a study to decide which other relations we might model to support further relation-focused question types beyond Explanation and Comparison. As we discuss in Section 5.3, models of temporal sequence and whole-part structure would both be candidates to support broader use of the techniques we study in this chapter.

Given the informal feedback received in our evaluation, as well as the low interannotator agreement on the Comparison question rating task, another important issue is whether these questions can be refined to make more sense to users. In our informal feedback, several colleagues indicated that the concepts of a "comparative" question/response were intelligible, but tended to imply a point-for-point comparison on "matching" properties. They found the idea of comparative information without this kind of matching harder to grasp, and the low interannotator agreement figures in our evaluation of these questions likely reflect this.[17]

Lastly, an issue on which we touch only briefly is how best to integrate knowledge derived from our RSR models in the system. First, there is the question of training a system to integrate RSR-derived information alongside other features of interest. In the

[17] The process by which humans make comparative judgments has been studied in the psychological literature; some insight into the problem we face here is given by Mussweiler (Mussweiler, 2003), who differentiates comparisons with respect to an implicit class "norm" from comparisons with respect to a a specific class instance. For instance, a comparison of Actor X with a class "norm" might follow along the lines of, "Actor X has appeared in many movies.", where the impicit norm is for actors to get only a modest number of roles. A comparison relative to a class instance might appear more like, "Actor X has been in more movies than Gwyneth Paltrow." Our idea that one can present *a priori* comparative information can in this sense be seen as making an implicit contrast to a so-called "norm," rather than a comparison to a specific class instance.

implementation explained in this chapter, we estimate many parameters in CCQ manually based on ad hoc experiments and our previous experience with DefScriber; going forward, the system would no doubt benefit from formal training. Second, there is also the question of where in the CCQ pipeline to use our lists of relationally relevant terms. Currently, the terms are used to identify question-relevant sentences after document retrieval has been completed. Yet as we mention in the chapter, we observe that in some cases using these terms earlier in the pipeline, i.e. as keywords for document retrieval, may be desirable; still, because of the issues of polysemy and general precision loss that attend any query-expansion process, we have not pursued this approach in general. However, a plausible compromise might involve a variant of pseudo-relevance feedback. In this setting, we can consider that while our RSR models reflect the entire probable spectrum of Cause- or Contrast-related terms for a given Focus, and that we can dynamically filter this list to the relevant terms in a given case by examining the documents from an initial query using only question terms; then, in the style of pseudo-relevance feedback, we might dynamically query for documents which match not only question terms but the filtered set of relation-relevant terms.

Chapter 6

Contributions, Limitations and Future Work

6.1 Summary

Our work contains important results in long-answer question answering, the modeling of rhetorical-semantic relations (RSRs), and the intersection of these two areas. We provide here a concise summary of the major points of the thesis, framed by the research questions we raise in our introduction in Chapter 1.

- *How do we create well-structured, relevant answers to definitional, biographical and topic-focused questions?*

  In our work with **DefScriber and long-answer QA** (Chapter 2), we demonstrate that a hybrid approach which combines goal-driven and data-driven strategies can produce successful descriptive answers for definitional, biographical and topic-focused questions. We implement a top-down component that identifies key *definitional predicate* information, and we adapt and extend methods from multi-document summarization to implement a robust, bottom-up component. In various evaluations, DefScriber performs among the top tier of state-of-the-art systems for this task.

- *How do we learn and refine a model of rhetorical and semantic concepts for use as a resource in answering these questions?*

In our work with **rhetorical-semantic relation (RSR) learning and classification** (Chapter 3), we extend the work of Marcu and Echihabi (Marcu and Echihabi, 2002), and show that methods which incorporate topic segment structure and syntactic features result in improved relation classification accuracy. Our results indicate that further analysis of these features has the potential to be fruitful, particularly with the recent availability of the Penn Discourse TreeBank (Prasad et al., 2006) (PDTB).

- *How can we apply this rhetorical-semantic model to enhance the performance and scope of our question answering techniques?*

  In exploring the intersection of these two areas, we analyzed two methods of **applying RSRs within long-answer QA**. First, we use RSR models to implement a measure of inter-sentence cohesion, but find only incremental improvements with respect to response ordering and content selection measures (Chapter 4). Second, we implement CCQ, a system for answering two "relation-focused" question types, and find that for Explanation questions, our RSR models provide a significant improvement in performance (Chapter 5).

  While we have thus far achieved only measured improvements through the application of RSRs, our work represents important early steps in this new direction. We believe that especially with the availability of PDTB, there will be more research in building and applying RSR-like models, and that our work provides useful initial results in this emerging area.

In the following sections, we review these contributions by chapter, address limitations of our work, and discuss future directions.

6.2 Contributions by Chapter

6.2.1 DefScriber

- We use an innovative, hybrid method which combines knowledge-driven (top-down) and data-driven (bottom-up) techniques for answering non-factoid question types such as definitional, biographical and topic-focused questions. Using a corpus study for motivation, we propose a set of *definitional predicate* knowledge types for the top-down approach, and adapt several methods used in multi-document summarization for the bottom-up component. We find that this method is suitable not only for definitions, but can be adapted successfully to biographical and topic-focused summarization as well.

- We analyze multiple evaluations of system performance in various settings. In a survey-based evaluation of definitional question answers, we find DefScriber outperforms baseline techniques in four of five rated categories; in an automated evaluation comparing system answers to human ones, DefScriber outperforms all of the more than twenty peer systems for a biographical question task (DUC 2004); and in a topic-focused question task our system is consistently among the top third of peers across manual and automated quality metrics (DUC 2005 and 2006).

- We implement and evaluate a novel extension to our clustering module which uses features from the speech-recognition process to improve the integration of speech and text input documents. Our evaluation demonstrates that this method improves results in a document clustering task over a set of mixed speech and text documents.

6.2.2 Rhetorical-Semantic Relations

- Using the framework of Marcu and Echihabi (Marcu and Echihabi, 2002) as a starting point, we implement novel methods for improving RSR model quality using topic segmentation and syntactic filtering, as well as optimization of classification parameters.

- We perform an extensive set of evaluations using both automatic and human-annotated data. For the human-annotated training set, we extract training data from the Penn Discourse TreeBank (Prasad et al., 2006) (PDTB), becoming one of the first research projects to experiment with this new resource.

  These evaluations provide insight into several levels of the RSR modeling and classification problem, including the effect of basic parameters such as vocabulary size and training set size, as well as more complex ones which filter training data using topic segmentation and syntactic filtering. In the case of syntactic filtering, we are able to leverage PDTB to perform evaluation on several levels, evaluating not just the net effect on classification accuracy, but also the performance of the heuristics themselves.

Our findings show that each of our methods for improving model quality results in classification improvements, and furthermore that we can expect improved syntactic heuristics to further refine our models, even when working with imperfect (i.e. automatic) parse data.

6.2.3 Applying RSRs in DefScriber

- We design an RSR-derived cohesion feature which combines information from the RSR models of TextRels to assess inter-sentence cohesion. Leveraging data from the Document Understanding Conferences (DUCs), we perform supervised training to weight this new feature in a combined framework with other, non-RSR-based, features.

- We evaluate the contribution of this feature with respect to content selection and ordering in responses. While the RSR cohesion feature improves selection and ordering in responses only incrementally, we find that the supervised training framework and other updates significantly improve DefScriber's performance in selecting response content for both biographical and topic-focused questions.

6.2.4 Applying RSRs in CCQ

- We propose a novel class of "relation-focused" questions and define two specific instances of such questions, namely Explanations and Comparisons.

- We create a novel application of RSR models as the basis for a lexical expansion approach for finding terms relevant to a relation-aware question interpretation. We use likelihood ratios to suggest related terms from our Cause and Contrast RSR models for Explanation and Comparison questions, respectively.

- We evaluate CCQ, a system which combines these techniques with existing modules from DefScriber to answer Comparison and Explanation questions. Our results suggest that users can more easily agree on the quality of answers to Explanation questions, whereas for Comparison questions, consistent judgments are more elusive. In addition, we find that using RSR-derived terms improves response relevance for Explanation questions, and that the improvement is statistically significant when we use ideal document input.

6.3 Deliverables

DefScriber The DefScriber system, as described in Chapter 2, has been used as a standalone, end-to-end pipeline for answering various long-answer question types, and has been integrated into a Web-based interface to support demonstrations and ad hoc queries. In Chapter 4 we extend DefScriber by adding an RSR-based coherence feature, a post-sentence-selection response reordering algorithm, and we implement a supervised learning experiment to train DefScriber's parameter settings for its sentence selection and ordering algorithms.

In addition, DefScriber modules have been adapted outside of DefScriber both in this thesis (for the CCQ system) and for other research projects. For instance, our clustering module has been used to present the results of an opinion classification system by grouping related themes, and to combine partially overlapping answers within a multi-strategy QA system.

TextRels The TextRels system, described in Chapter 3, produces RSR models from unannotated corpora using cue phrase patterns, and can filter extracted relation instances using segmentation and syntax-based methods. TextRels also implements classification and related-term-suggestion methods over its RSR models. The RSR models themselves are built as relational databases to take advantage of the performance and portability of implementations like MySQL; we have already begun a collaboration where our RSR models are being used by another group within the context of paraphrase detection.

CCQ The CCQ system, described in Chapter 5, produces end-to-end answers for both Explanation and Comparison questions, using related term suggestions generated from TextRels' RSR models and other resources to detect question-relevant material, and modules from DefScriber to compose answers. CCQ is integrated within a Web interface, which allows it to be used for demonstrations and ad hoc queries.

6.4 Limitations and Future Work

Our approach in this thesis is weighted more toward building systems with good end-to-end performance, and less on deep, module-level optimization and analysis. While in some cases this may limit our results, we feel that on balance this is a fundamental matter of tradeoffs. On the one hand, in the span of the thesis we are able to cover a number of applications and evaluations, many of which represent novel research directions. On the other hand, we do not always pursue in-depth analysis which could potentially lead to further insights and improvements within components of these applications. While we are comfortable with the balance we have struck between these competing interests, we feel it is important to acknowledge this facet of our work.

For instance, consider some standard assumptions which we make, such as the use of inter-document frequency (IDF) as a measure of term salience. This measure is used in many of the applications described in this thesis, yet may not be ideal for every task. An investigation of other models of term likelihood, for instance using topic- or query-specific language models, would be an entirely useful and appropriate direction for future research. However, within the course of this thesis, we have tended toward using standard solutions, where they exist, to build robust system components more quickly (e.g., using IDF as a salience function in the centroid-based sentence ranking in DefScriber) and focus our effort on new techniques (e.g., implementing RSR-derived features) in the process of building applications which perform well in end-to-end evaluation.

We have not always struck the balance this way. In some cases, we have done a greater level of in-depth, module-level evaluation. For instance, in our work with speech-text clustering, we examine a solution which looks beyond the standard approach of using the flat text representation of ASR documents for text applications. And our work in Chapter 3 on text classification goes substantially beyond simply using the model of Marcu and Echihabi (Marcu and Echihabi, 2002) "as is," performing a number of experiments which analyze and improve their approach.

At a finer-grained level, there are several important limitations and possibilities for future work in each chapter. Within DefScriber, while we propose a set of many definitional predicates in Chapter 2, we ultimately implement Genus-Species using manually constructed patterns, and Cause and Contrast relations only as cohesion features. That is, for Cause, Contrast and other potential predicates, we do not use them to directly identify question-relevant content in terms of causes or contrasts with respect to the entity or topic named in the question (i.e., the subject of the definition/description itself). On the one hand, we can try to implement some of the content-focused, term expansion approaches as in CCQ, although it is unclear whether they will be successful for questions which are not explicitly relation-focused. Another possibility is to expand our form-focused methods to identify explicit relational statements with respect to question subjects, perhaps building on recent strong results in automated acquisition of relation-indicating patterns (Snow et al., 2006).

In our experiments in Chapter 4, we find that even with the integration of a new RSR-based cohesion feature and supervised learning framework, DefScriber's results on a content ordering task are only marginally better than random. The synthetic aspects of this task may contribute to our poor performance, as we attempt to recreate ordering of a human-written abstract, which may make it more difficult to use the redundancy-based ordering features which are typically helpful in guiding content ordering. Nonetheless, we feel that these experiments reveal important issues in DefScriber's architecture which could be improved. For one thing, DefScriber's basic concept of all-at-once content selection and ordering may be limiting. While we attempt to address this issue with a post-content-selection reordering algorithm, the results using this algorithm do not improve significantly. A more fundamental approach to addressing this problem might involve using ordering models which consider global structure rather than just local (Barzilay and Lee, 2004), or which are more selective in the features that they consider for local cohesion measures, i.e. entities (Barzilay and Lapata, 2005) or limited part-of-speech (Lapata, 2003). Sentence compression is another key feature that other competitive systems have begun to utilize (Siddharthan et al., 2004; Daumé III and Marcu, 2005; Vanderwende et al., 2006) and that we would like to integrate into DefScriber.

In our work with building and refining Rhetorical-Semantic Relation models in Chapter 3, we are especially enthusiastic about further experimentation using the data in the Penn Discourse TreeBank (PDTB) (Prasad et al., 2006). PDTB was released after most of our research had been completed, and the data it contains could be especially useful in refining our syntactic heuristics, as well as for training and testing new approaches.

An important limitation of our approach for RSR modeling is the representation of TextRels RSR models using a full-dimensionality matrix based on actual word stem pairs. While this technique allows us to maintain an intuitive model which can easily be manipulated, we also believe that mapping to a lower-dimensionality model would increase efficiency and conflate individual terms which may have similar distributions with respect to our relations of interest. This kind of dimensionality reduction is typically done with singular value decomposition (SVD) and commonly used in latent semantic indexing; it could be used in an analogous way for our RSR models. The abstractions which are introduced by reduced dimensionality would be especially useful in another possible area of future work, namely the addition of non-surface features to our model, such as part-of-speech or dependency-based features, since these techniques would likely create additional sparsity in our models. In adding syntactic features to our vector-based model, we can build on the results of Padó and Lapata (Padó and Lapata, 2003), who report success in adding such features to a vector-based model for distinguishing between lexical relations.

In our work in relation-focused question answering with CCQ, we see important limitations in several areas. Our current approach focuses primarily on the *content* of response candidate sentences, using lexical expansion from RSR models and other techniques to measure intrinsic sentence relevance for a given question. However, techniques which focus on sentence *form* with respect to syntactic and semantic roles could be a very useful complement, providing insight as to whether relevant terms appear in a form that adds to or subtracts from our relevance computation.

Another limitation of our work with Explanation and Comparison questions is the relatively small scale of our evaluation. We believe that for these new types of questions, where there is not gold-standard evaluation data available, our effort represents an important initial foray into this sparsely researched area. However, additional data is needed to explore the area further and allow for stronger conclusions. An issue which clearly arises from our evaluation is whether the Comparison questions that we implement make sense to users. We find that informal user feedback, as well as the relatively low interannotator agreement in our study, suggest that the idea of "comparative" information may be ill-defined in isolation. These results suggest that, in order to be convincing to users, a Comparison answer may need to match similar information and compare the two Topics of the question in a point-for-point manner.

Appendix A

Cluster Evaluation Metrics

In this appendix we present the derivation of the Homgeneity and Completeness measures for cluster evaluation discussed in Chapter 2. In this derivation we use C to represent the correct set of clusters, and K the set of classes which a clustering algorithm produces, and M_{ij} is the cell in the $|C|$ by $|K|$ matrix which contains the count of items in cluster i which are placed in class j by the classification algorithm.

Homogeneity and Completeness are calculated as:

$$\begin{aligned}
\text{Homogeneity} &= 1 - \tfrac{H(C|K)}{H(C)} \\
\text{Entropy of classes within each cluster} &= H(C|K) \\
\text{Entropy of the distribution of classes} &= H(C) \\
\text{Completeness} &= 1 - \tfrac{H(K|C)}{H(K)} \\
\text{Entropy of clusters containing each class} &= H(K|C) \\
\text{Entropy of the distribution of clusters} &= H(K)
\end{aligned}$$

The conditional entropies can be calculated as:

$$
\begin{aligned}
H(K|C) &= -\textstyle\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{M_{ck}}{n}\log\frac{M_{ck}}{M_{c\bullet}}\\
H(C|K) &= -\textstyle\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{M_{ck}}{n}\log\frac{M_{ck}}{M_{\bullet k}}\\
H(C) &= -\textstyle\sum_{c=1}^{|C|}\frac{M_{c\bullet}}{n}\log\frac{M_{c\bullet}}{n}\\
H(K) &= -\textstyle\sum_{k=1}^{|K|}\frac{M_{\bullet k}}{n}\log\frac{M_{\bullet k}}{n}\\
M_{c\bullet} &= \textstyle\sum_{k=1}^{|K|}M_{ck}\\
M_{\bullet k} &= \textstyle\sum_{c=1}^{|C|}M_{ck}
\end{aligned}
$$

Appendix B

RSR Patterns and Instance Distributions

In this appendix we list text patterns used to extract Cause and Contrast instances in the Instance Mining phase as defined in Chapter 3. *BOS* and *EOS* stand for beginning- and end-of-sentence markers; W_1 and W_2 stand for a string of one or more words, which are the relation spans which are captured by a given pattern. (As described in that chapter, Adjacent and NoRelSame instances use patterns whose only anchors are sentence boundaries. Note that for mining segmented instances, we use the same patterns, but add the constraint that a segment boundary must *not* occur between sentences in our two-sentence Cause and Contrast patterns.

The counts below list the total count of instances for each pattern/relation extracted over the entire Gigaword corpus using patterns without topic segment constraints. We do not provide the per-pattern counts for the topic-segment constrained instances, but do show the segmented/unsegmented count differences in the totals sections.

As described in Chapter 3, we extract less than the total available number of instances for NoRelSame and Adjacent for efficiency/storage reasons, whereas we extract all possible instances for Cause and Contrast. Note also that the actual number of instances used for training our models is generally less than the full number of extracted instances listed here; we also explain this in Chapter 3.

B.1 Cause Patterns

For the **Cause** relation, we use the following patterns. Note that, as discussed in Chapter 5, these patterns have a semantic directionality, i.e. one text span is inferred to be in a *cause* role and the other in an *effect* role; this is indicated by either a C (cause) and E (effect) after the W_1 and W_2 span capturing placeholders in these patterns.

This listing shows each pattern with its associated per-pattern count for the non-segment-constrained patterns only. In the totals section, we give counts for both segment constrained and unconstrained patterns.

```
<BOS> W1(E) because W2(C) <EOS>:        1015047
<BOS> W1(E) because of W2(C) <EOS>:     372950
<BOS> W1(C) in order to W2(E) <EOS>:    89252
<BOS> because W1(C), W2(E) <EOS>:       64035
<BOS> W1(E) as a result of W2(C) <EOS>: 39882
<BOS> W1(C) <EOS> <BOS>as a result , W2(E) <EOS>:       30969
<BOS> because of W1(C), W2(E) <EOS>:    29123
<BOS> W1(C), causing W2(E) <EOS>:       28051
<BOS> W1(E) on the basis of W2(C) <EOS>:        22127
<BOS> W1(C) <EOS> <BOS>thus W2(E) <EOS>:        20523
<BOS> W1(C), thus W2(E) <EOS>:  20421
<BOS> W1(C) <EOS> <BOS>therefore W2(E) <EOS>:   16046
<BOS> W1(E) , due to W2(C) <EOS>:       13114
<BOS> in order to W1(E), W2(C) <EOS>:   10415
<BOS> W1(C) <EOS> <BOS>that is why W2(E) <EOS>: 7546
<BOS> due to W1(C), W2(E) <EOS>:        7469
<BOS> W1(C), thereby W2(E) <EOS>:       7346
<BOS> W1(C), therefore W2(E) <EOS>:     7221
<BOS> as a result of W1(C), W2(E) <EOS>:        6780
<BOS> W1(E) on the grounds that W2(C) <EOS>:    6342
<BOS> W1(C), which is why W2(E) <EOS>:  4670
```

```
<BOS> W1(C), as a result W2(E) <EOS>:   4655
<BOS> W1(C) <EOS> <BOS>consequently W2(E) <EOS>:        3734
<BOS> W1(C) <EOS> <BOS>hence W2(E) <EOS>:       3678
<BOS> W1(C) <EOS> <BOS>for that reason W2(E) <EOS>:      2855
<BOS> W1(E) on the grounds of W2(C) <EOS>:      2663
<BOS> W1(C) <EOS> <BOS>this is why W2(E) <EOS>: 2550
<BOS> W1(C) <EOS> <BOS>accordingly W2(E) <EOS>: 2226
<BOS> W1(C), which resulted in W2(E) <EOS>:     2181
<BOS> W1(C), hence W2(E) <EOS>: 2057
<BOS> W1(E) as evidence of W2(C) <EOS>: 1887
<BOS> W1(C) <EOS> <BOS>because of that W2(E) <EOS>:     1834
<BOS> W1(C) <EOS> <BOS>which is why W2(E) <EOS>:        1751
<BOS> W1(C) <EOS> <BOS>for this reason W2(E) <EOS>:      1415
<BOS> W1(E) on account of W2(C) <EOS>:  1334
<BOS> on the basis of W1(C), W2(E) <EOS>:       1329
<BOS> W1(C) <EOS> <BOS>because of this W2(E) <EOS>:     1315
<BOS> W1(C), that is why W2(E) <EOS>:   1194
<BOS> W1(E) on the basis that W2(C) <EOS>:      1033
<BOS> W1(E) for reasons of W2(C) <EOS>: 978
<BOS> W1(C) <EOS> <BOS>as a consequence , W2(E) <EOS>:  814
<BOS> W1(E) as a consequence of W2(C) <EOS>:    806
<BOS> W1(C) <EOS> <BOS>in so doing W2(E) <EOS>: 644
<BOS> W1(C) in order that W2(E) <EOS>:  601
<BOS> W1(C), consequently W2(E) <EOS>:  491
<BOS> W1(C), with the result that W2(E) <EOS>:  407
<BOS> W1(C), which explains why W2(E) <EOS>:    357
<BOS> W1(C) <EOS> <BOS>that explains why W2(E) <EOS>:   343
<BOS> W1(C) <EOS> <BOS>on that basis W2(E) <EOS>:       321
<BOS> as evidence of W1(C), W2(E) <EOS>:        261
<BOS> W1(C), for that reason W2(E) <EOS>:       240
```

```
<BOS> W1(C) <EOS> <BOS>on this basis W2(E) <EOS>:       218
<BOS> W1(C), this is why W2(E) <EOS>:   213
<BOS> W1(C), because of that W2(E) <EOS>:       211
<BOS> W1(C), as a consequence W2(E) <EOS>:      182
<BOS> as a consequence of W1(C), W2(E) <EOS>:   181
<BOS> W1(C) <EOS> <BOS>this explains why W2(E) <EOS>:   171
<BOS> W1(C), in so doing W2(E) <EOS>:   161
<BOS> W1(C), accordingly W2(E) <EOS>:   136
<BOS> W1(C), that resulted in W2(E) <EOS>:      131
<BOS> W1(C), because of this W2(E) <EOS>:       121
<BOS> for reasons of W1(C), W2(E) <EOS>:        109
<BOS> W1(C) <EOS> <BOS>which explains why W2(E) <EOS>:  103
(17 others with less than 100 instances each.)


TOTALS:


One sentence, cue-E-C patterns: 10447 (same for seg-contrained)
One sentence, cue-C-E patterns: 109389 (same for seg-constrained)
One sentence, C-cue-E patterns: 170494 (same for seg-constrained)
One sentence, E-cue-C patterns: 1478163 (same for seg-constrained)
Two sentence patterns: 99126 (73753 for seg-constrained)

Total count: 1867619 (1842246 for seg-constrained)
```

B.2 Contrast Patterns

For the **Contrast** relation, we use the following patterns. Unlike the Cause patterns listed above, no semantic directionality is inferred across the spans captured.

This listing shows each pattern with its associated per-pattern count for the non-segment-constrained patterns only. In the totals section, we give counts for both segment constrained and unconstrained patterns.

```
<BOS> W1 <EOS> <BOS>but W2 <EOS>:       2495726
<BOS> W1, but W2 <EOS>: 2306869
<BOS> W1 <EOS> <BOS>however , W2 <EOS>: 299435
<BOS> although W1, W2 <EOS>:    185666
<BOS> W1, although W2 <EOS>:    146887
<BOS> W1, though W2 <EOS>:      111061
<BOS> W1 <EOS> <BOS>yet W2 <EOS>:       92309
<BOS> though W1, W2 <EOS>:      77802
<BOS> W1 <EOS> <BOS>instead , W2 <EOS>: 58814
<BOS> W1, even though W2 <EOS>: 46148
<BOS> even if W1, W2 <EOS>:     31444
<BOS> W1, even if W2 <EOS>:     30193
<BOS> instead of W1, W2 <EOS>:  27322
<BOS> even though W1, W2 <EOS>: 26628
<BOS> W1 <EOS> <BOS>nevertheless , W2 <EOS>:    21177
<BOS> W1 <EOS> <BOS>on the other hand , W2 <EOS>:       21061
<BOS> W1, instead of W2 <EOS>:  15787
<BOS> W1 <EOS> <BOS>nonetheless , W2 <EOS>:     12502
<BOS> W1 <EOS> <BOS>by contrast , W2 <EOS>:     11239
<BOS> W1, regardless of W2 <EOS>:       9652
<BOS> W1 <EOS> <BOS>rather , W2 <EOS>:  8353
<BOS> W1 <EOS> <BOS>in contrast , W2 <EOS>:     7973
<BOS> W1, whereas W2 <EOS>:     5655
<BOS> regardless of W1, W2 <EOS>:       4937
<BOS> in contrast to W1, W2 <EOS>:      3438
<BOS> W1 <EOS> <BOS>on the contrary , W2 <EOS>: 2849
<BOS> whereas W1, W2 <EOS>:     2563
```

```
<BOS> W1, in contrast to W2 <EOS>:      2052

<BOS> W1 <EOS> <BOS>conversely , W2 <EOS>:      2046

<BOS> W1 <EOS> <BOS>regardless , W2 <EOS>:      1375

<BOS> W1 <EOS> <BOS>on the other side , W2 <EOS>:       1045

<BOS> in contrast with W1, W2 <EOS>:    295

<BOS> W1, in contrast with W2 <EOS>:    246

<BOS> by contrast with W1, W2 <EOS>:    44

<BOS> by contrast to W1, W2 <EOS>:      38

<BOS> W1, by contrast with W2 <EOS>:    16

<BOS> W1, by contrast to W2 <EOS>:      3


One sentence, W1-cue-W2 patterns:    2674569 (same for seg-constrained)

One sentence, cue-W1-W2 patterns:    360177 (same for seg-constrained)

Two sentence patterns:   3035904 (2674569 for seg-constrained)


Total Count: 6070650 (5318336 for seg-constrained)
```

Appendix C

CCQ Evaluation Materials and Sample Responses

This appendix lists the complete set of test questions, as well as the evaluator instructions, from the user evaluation of our CCQ system reported in Chapter 5. In addition, we include several example responses for both Explanation and Comparison questions.

C.1 Evaluation Questions

C.1.1 Explanation Questions

Table C.1: Explanation questions and corresponding TDT . topics used in evaluation reported in Chapter 5

TDT Topic Description (ID)	**Question Type**	**Question**
Scientists Lose Contact with Mir (40003)	Explanation-Cause	Describe [causes] of [lose contact] in [Scientists Lose Contact with Mir].
Self-immolation Attempt by Falun Gong Followers. (40025)	Explanation-Cause	Describe [causes] of [protest] in [Self-immolation Attempt by Falun Gong Followers] .

President Clinton Issues Pardons (40034)	Explanation-Cause	Describe [causes] of [decision] in [President Clinton Issues Pardons] .
Australian Open Tennis Championship (40037)	Explanation-Cause	Describe [causes] of [losing games] in [Australian Open Tennis Championship] .
Swedish Foreign Minister killed	Explanation-Cause	Describe [causes] of [killing] in [Swedish Foreign Minister killed] .
Sosa ejected cheating suspected (55029)	Explanation-Cause	Describe [causes] of [suspicion] in [Sosa ejected cheating suspected] .
Train Fire in Austrian Tunnel (40048)	Explanation-Cause	Describe [causes] of [fire] in [Train Fire in Austrian Tunnel] .
Iliescu Wins Romanian Elections (41019)	Explanation-Cause	Describe [causes] of [win election vote] in [Iliescu Wins Romanian Elections] .
Edmond Pope Convicted for Espionage in Russia (40055)	Explanation-Cause	Describe [causes] of [conviction] in [Edmond Pope Convicted for Espionage in Russia] .
Russian Nuclear Submarine Kursk Sinks (40004)	Explanation-Cause	Describe [causes] of [sinking accident] in [Russian Nuclear Submarine Kursk Sinks] .
Taiwanese Premier Tang Fei Resigns (40019)	Explanation-Cause	Describe [causes] of [resignation] in [Taiwanese Premier Tang Fei Resigns] .
Oil Tanker Jessica Galapagos Spill (40001)	Explanation-Effect	Describe [effects] of [accident] in [Oil Tanker Jessica Galapagos Spill] .
Former US President Reagan Breaks Hip (40031)	Explanation-Effect	Describe [effects] of [injury] in [Former US President Reagan Breaks Hip] .
Train Fire in Austrian Tunnel (40048)	Explanation-Effect	Describe [effects] of [fire] in [Train Fire in Austrian Tunnel] .
World Economic Forum Protests (40026)	Explanation-Effect	Describe [effects] of [protest] in [World Economic Forum Protests] .

Presidential Power Struggle in Yugoslavia (40007)	Explanation-Effect	Describe [effects] of [struggle] in [Presidential Power Struggle in Yugoslavia] .
US Senate Proposes Easing Cuban Trade Embargo (40009)	Explanation-Effect	Descibe [effects] of [embargo] in [US Senate Proposes Easing Cuban Trade Embargo] .
Tiananmen Papers Published (40033)	Explanation-Effect	Describe [effects] of [publishing] in [Tiananmen Papers Published] .
Clinton Lifts Sanctions Against Yugoslavia (41015)	Explanation-Effect	Describe [effects] of [sanctions] in [Clinton Lifts Sanctions Against Yugoslavia] .
Popocatepetl Volcano Erupts (41028)	Explanation-Effect	Describe [effects] of [volcano eruption] in [Popocatepetl Volcano Erupts]

C.1.2 Comparison Questions

Table C.2: Comparison questions and corresponding TDT . topics used in evaluation reported in Chapter 5

TDT Description (ID), Topic-A	**TDT Description (ID), Topic-B**	**Question**
Earthquake hits India Gujarat State (40038)	Earthquake in El Salvador (40021)	Compare [India Gujarat State] and [El Salvador] with respect to [earthquakes] .
Parliamentary Elections in Egypt (41007)	Ilyescu Wins Romanian Elections (41019)	Compare [Parliamentary Elections in Egypt] and [Ilyescu Wins Romanian Elections] with respect to [voter turnout problems] .

Gao Xingjian Wins Nobel Prize (40049)	The 2000 Nobel Peace Prize (41002)	Compare [Gao Xingjian Wins Nobel Prize] and [The 2000 Nobel Peace Prize] with respect to [achievements] .
Guar Clone Born to Cow (40030)	2000 Nobel Prize in Medicine Awarded (40060)	Compare [Guar Clone Born to Cow] to [2000 Nobel Prize in Medicine Awarded] with respect to [scientific achievements] .
Kobe charged with sexual assault (55047)	Edmond Pope Convicted for Espionage in Russia (40055)	Compare [Kobe charged with sexual assault] and [Edmond Pope Convicted for Espionage in Russia] with respect to [criminal prosecution] .
Oil Tanker Jessica Galapagos Spill (40001)	Russian Nuclear Submarine Kursk Sinks (40004)	Compare [Oil Tanker Jessica Galapagos Spill] and [Russian Nuclear Submarine Kursk Sinks] with respect to [explanations] .
UN Climate Conference (40002)	Arab League Summit meeting in Cairo (41018)	Compare [UN Climate Conference] and [Arab League Summit meeting in Cairo] with respect to [diplomatic strategy] .
Train Fire in Austrian Tunnel (40048)	Car Bomb Explodes in Northern Spain (41016)	Compare [Train Train Fire in Austrian Tunnel] and [Car Bomb Explodes in Northern Spain] with respect to [investigation] .
Popocatepetl Volcano Erupts (41028)	Earthquake in India Gujarat State (40038)	Compare [Popocatepetl Volcano Erupts] and [Earthquake in India Gujarat State] with respect to [injuries deaths casualties] .

UN official killed in attack (55105)	Bombing in Riyadh Saudi Arabia (55106)	Compare [UN official killed in attack] and [Bombing in Riyadh Saudi Arabia] with respect to [deadly attacks] .

C.2 Sample Responses

This section includes sample responses produced by the CCQ system. Some of these examples are included, in part, in Chapter 5, Figures 5.1, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, and Tables 5.4 and 5.10; sample responses include responses from our set of test questions, as well as several other example questions mentioned in the chapter. For all questions, RSR-suggested terms are shown in **bold**, LSA-suggested terms in *italics*, and terms suggested by both RSR and LSA methods are <u>underlined</u>. All answers here are produced with Oracle mode IR, unless otherwise indicated. Answers are shown using the Semantic+RSR, Semantic and Baseline settings as described in Chapter 5.7.2.

C.2.1 Explanation Questions

Q Describe causes of [conviction] in [Edmond Pope Convicted for Espionage in Russia]

A (Semantic+RSR) :

- In February, rocket scientist Anatoly Babkin was handed a suspended eight year *jail sentence* after being convicted of handing over secret missile data to former US naval intelligence officer Edmond Pope.
- Astakhov told reporters on Wednesday that if the court record **proves** that Pope did receive classified **information**, “it is not the problem of a private businessman, but a problem of Russia” to clarify laws on the handling of state secrets.
- The verdict marked the first espionage conviction against a Westerner in Russia in the decade since the end of the Cold War, though both Moscow and Washington have expelled alleged spies from each other’s capitals in recent years.

- Among the witnesses whose testimony was thrown out was U.S. researcher Daniel H. Kiely, who was *arrested* along with Pope.

A (Semantic) :

- In February, rocket scientist Anatoly Babkin was handed a suspended eight year *jail sentence* after being convicted of handing over secret missile data to former US naval intelligence officer Edmond Pope.
- The president encouraged Pope's release on humanitarian grounds because Pope is sick, White House press secretary Jake Siewert told reporters in Nebraska.
- The cars headed straight to Moscow's main international airport and a plane carrying Pope took off for Frankfurt, Germany, shortly before noon, said Pope's lawyer, Pavel Astakhov.
- Astakhov told reporters on Wednesday that if the court record proves that Pope did receive classified information, "it is not the problem of a private businessman, but a problem of Russia" to clarify laws on the handling of state secrets.

A (Baseline) :

- The verdict marked the first espionage conviction against a Westerner in Russia in the decade since the end of the Cold War, though both Moscow and Washington have expelled alleged spies from each other's capitals in recent years.
- Russia's presidential pardons commission recommended on Friday that President Vladimir Putin free U.S. prisoner Edmond Pope, who was convicted of spying earlier in the week and sentenced to 20 years in prison.
- The president encouraged Pope's release on humanitarian grounds because Pope is sick, White House press secretary Jake Siewert told reporters in Nebraska.
- The cars headed straight to Moscow's main international airport and a plane carrying Pope took off for Frankfurt, Germany, shortly before noon, said Pope's lawyer, Pavel Astakhov.

Q Describe effects of [sanctions] in [Clinton Lifts Sanctions Against Yugoslavia] .

A (Semantic+RSR) :

- USA Engage, one such group in Washington, **estimates** that sanctions cost $15 billion to $19 billion in **lost exports** each year and that American companies employ 200,000 fewer people in high-wage, **export**-oriented jobs because of **lost** sales.
- President Clinton began lifting trade and **economic** sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade.
- The EU imposed sanctions in response to Milosevic's repression of the opposition in Serbia and the crackdown by Yugoslav troops in Kosovo province, which provoked last year's NATO bombing campaign against Serbia.
- But the Yugoslavia case could plausibly be remembered as the most significant success for **economic** sanctions since the fall of the apartheid regime in South Africa more than a decade ago.

A (Semantic) :

- President Clinton began lifting trade and economic sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade.
- The EU imposed sanctions in response to Milosevic's repression of the opposition in Serbia and the crackdown by Yugoslav troops in Kosovo province, which provoked last year's NATO bombing campaign against Serbia.
- The ministers rejected a proposal that would have made ending sanctions conditional on the new Belgrade authorities committing to hand Milosevic, who has been indicted for alleged atrocities, over to the international war crimes tribunal.
- But the Yugoslavia case could plausibly be remembered as the most significant success for economic sanctions since the fall of the apartheid regime in South Africa more than a decade ago.

A (Baseline) :

- President Clinton began lifting trade and economic sanctions against Yugoslavia on Thursday in a gesture of support for the newly elected regime in Belgrade.

- President George W. Bush lifted sanctions Thursday on about 45 people from the former Yugoslavia who had repressed democracy or promoted human rights abuses.
- The sanctions also included restrictions on air travel and oil sales to Yugoslavia, though both of those sets were eased even before Milosevic's fall.
- The NATO allies imposed a variety of sanctions on Yugoslavia beginning in 1992, during the war in Bosnia.

Q Describe causes of [protest] in [Self immolation Attempt by Falun Gong Followers] .

A (Semantic+RSR) :

- The official accounts could not be independently verified, but appeared to offer detailed evidence that the protest was undertaken by ardent Falun Gong **believers** who plotted the protest together.
- The self-immolation and very **public** suicide is the most dramatic act so far in the *escalating* 18-month war of wills between the Chinese government and Chinese members of Falun Gong, which the government has labeled an "evil cult."
- Official frustration with the continued defiance has been joined by **fear** that Falun Gong protests could mar the visit by the International Olympic Committee next month to inspect Beijing as a possible site for the 2008 Olympics.
- Last week several newspapers contained long accounts about hundreds of Falun Gong members who had been released from labor *camps* or whose sentences had been reduced, generally after giving up their practice and *denouncing* the spiritual group.

A (Semantic) :

- The self-immolation and very public suicide is the most dramatic act so far in the *escalating* 18-month war of wills between the Chinese government and Chinese members of Falun Gong, which the government has labeled an "evil cult."

- The official accounts could not be independently verified, but appeared to offer detailed evidence that the protest was undertaken by ardent Falun Gong believers who plotted the protest together.
- Last week several newspapers contained long accounts about hundreds of Falun Gong members who had been released from labor *camps* or whose sentences had been reduced, generally after giving up their practice and *denouncing* the spiritual group.
- State media reiterated claims that Falun Gong has caused more than 1,600 deaths and cited a Jan. 1 *statement* on group Web sites in which Li said the crackdown can "no longer be tolerated."

A (Baseline) :

- The official accounts could not be independently verified, but appeared to offer detailed evidence that the protest was undertaken by ardent Falun Gong believers who plotted the protest together.
- The self-immolation and very public suicide is the most dramatic act so far in the escalating 18-month war of wills between the Chinese government and Chinese members of Falun Gong, which the government has labeled an "evil cult."
- Falun Gong spokesmen in New York and Hong Kong immediately distanced the group from Tuesday's event, expressing skepticism about whether the dead and injured were Falun Gong members.
- State media underscored the government's claims it wants to help followers sever ties with Falun Gong and defended the crackdown as a just protection of human rights.

Q Describe effects of [publishing] in [Tiananmen Papers Published] .

A (Semantic+RSR) :

- If genuine, scholars who *translated* and published the papers said, the documents offer a rare glimpse into the motivations and fears behind the communist leadership's decision to **order** troops into Tiananmen Square.

- Peter Osnos, publisher and chief executive of Public Affairs, described the book’s author as a “compiler who had been specifically sent abroad with material to find a way to make it available.”
- The *book* is excerpted in the January/February *edition* of Foreign Affairs magazine, and CBS’ ”60 Minutes” posted portions of an interview with the civil servant, who uses the pseudonym Zhang Liang, on its Web site Friday.
- But, according to the Tiananmen papers, the standing committee was obligated by a **secret** intra-party resolution to refer any stalemate to Deng and the other senior revolutionaries.

A (Semantic) :

- If genuine, scholars who *translated* and published the papers said, the documents offer a rare glimpse into the motivations and fears behind the communist leadership’s decision to **order** troops into Tiananmen Square.
- Peter Osnos, publisher and chief executive of Public Affairs, described the book’s author as a “compiler who had been specifically sent abroad with material to find a way to make it available.”
- The *book* is excerpted in the January/February *edition* of Foreign Affairs magazine, and CBS’ ”60 Minutes” posted portions of an interview with the civil servant, who uses the pseudonym Zhang Liang, on its Web site Friday.
- Orville Schell, dean of the journalism school at the University of California-Berkeley and author of several *books* on China, worked with Nathan and Perry.

A (Baseline) :

- The documents, which number in the hundreds and have been deemed authentic by several experts, appear in “The Tiananmen Papers,” which will be published on Monday.
- The translated collection of documents is being published by PublicAffairs, a non-fiction publishing house in New York.

- The documents reveal deep-seated paranoia that the protests were controlled by unknown anti-communist conspirators and anxiety by the party's top leaders that the more than one million demonstrators gathered on Tiananmen Square could demand their arrest.
- For example, a supposed inside account of the 1971 fall of Mao's designated successor, Lin Biao, was published to great interest in 1983 but is now of dubious credibility.

Q Describe causes of flooding in China's Hainan Province. (**Using Open Document Retrieval**)

A (Semantic+RSR) :

- Bad **weather** is expected to **continue** in China, with CCTV reporting that **parts** of *southern* Guangdong province and Hainan were battening down the hatches in preparation for **torrential** and strong **winds** caused by **typhoon** Koni.
- Ominously,**rainfall** in eastern China was causing water **levels** torise again in the Huai river, which has already seen its worst flooding in more than a decade.
- Guangdong's capital, Guangzhou, saw heavy **rainfall** on Sunday, relieving the city of the hottest **weather** in half a century as the mercury **hit** 39 degrees Celsius (102 degrees Fahrenheit).
- Local officials said the north part of the province was seriously **hit**, especially the capital city Haikou and *nearby* Wenshan.

A (Semantic) :

- China's *southern* provinces of Hainan and Guangdong issued warnings on Sunday about possible floods and mud-rock flows as typhoon Krovanh, the 12th to hit China this year, moves closer to the provinces.
- As of early Monday, Koni hovered over the South China Sea, about 200 kilometers (125 *miles*) southeast of Hainan, an island province near Guangdong, Xinhua reported.

- All the sea routes on the Qiongzhou Straight linking Hainan and Guangdong have been suspended while the two provinces have strengthened the security measures for reservoirs and major river dykes, local officials said.
- Hainan, the country's southernmost province, took the lead in January 1994, banning commercial logging of virgin forest.

A (Baseline) :

- China's southern provinces of Hainan and Guangdong issued warnings on Sunday about possible floods and mud-rock flows as typhoon Krovanh, the 12th to hit China this year, moves closer to the provinces.
- Hainan, the country's southernmost province, took the lead in January 1994, banning commercial logging of virgin forest.
- All the sea routes on the Qiongzhou Straight linking Hainan and Guangdong have been suspended while the two provinces have strengthened the security measures for reservoirs and major river dykes, local officials said.
- Officials in central China's Zigui county toiled feverishly to clean up debris and damage from earlier floods, clearing a week-old landslide with explosives, Xinhua news agency reported.

Q Describe effects of flooding in China's Hainan Province. (**Using Open Document Retrieval**)

A (Semantic+RSR) :

- China's *southern* provinces of Hainan and Guangdong issued warnings on Sunday about possible floods and mud-rock flows as typhoon Krovanh, the 12th to hit China this year, moves closer to the provinces.
- In Zhanjiang city, the director of the flood, wind and drought office, surnamed Yin, said the storm caused 150 million **yuan** in **economic** losses.
- According to the China News Service, 70 flights from Haikou airport were **delayed**, stranding 6,000 *passengers*, although operations have since resumed.

- Officials in *central* China's Zigui county toiled feverishly to clean up debris and damage from earlier floods, clearing a week-old landslide with explosives, Xinhua news agency reported.

A (Semantic and Baseline) : *Answers are the same as for the "cause" version of the question shown above.*

C.2.2 Comparison Questions

We show the answer as produced by CCQ, including any matching of comparable facts, as is done in our Web interface as shown in Figure 5.3. Note that in the actual evaluation interface, the sentences are presented without any matching, simply by grouping together the two sentences for each question Topic, Figure 5.5.

Q Compare and contrast [diplomatic strategy] in [summit meetings] , highlighting [UN Climate Conference] versus [Arab League Summit meeting in Cairo] .

A (Semantic+RSR) :

- Fact for [UN Climate Conference]: For his part, Hansen **seems** satisfied **simply** to have nudged the climate debate toward considering partial, common-**sense** solutions that he feels have been neglected.
- Fact for [Arab League Summit meeting in Cairo]: Meanwhile, an Arab summit in Cairo, Egypt, ended Sunday with a declaration that Arab states may consider cutting *ties* with Israel, but did not *insist* that they do so.
- Fact for [UN Climate Conference]: By next weekend, participants at the U.N. climate conference hope to have a detailed *agreement* on how to cut emissions of the greenhouse gases that are gradually warming the planet and changing its weather.
- Fact for [Arab League Summit meeting in Cairo]: The carefully worded *statement* addressed what emerged as the most contentious issue before the leaders meeting in Cairo: Whether *ties* several have with the Jewish state should be reconsidered. (doc 25)

A (Semantic) :

- Fact for [UN Climate Conference]: But for the last decade, carbon dioxide has been the main target of climate experts and diplomats trying to stabilize climate.
- Fact for [Arab League Summit meeting in Cairo]: The Cairo gathering, the first Arab League summit meeting in a decade to include Iraq, brought together the divergent trends in the Arab world.
- Comparable Facts:

  [UN Climate Conference]: The Shanghai conference was the start of a series of meetings under U.N. auspices to gather evidence for climate *negotiators*.

  [Arab League Summit meeting in Cairo]: It was not clear if Arab nations would not go to international gatherings attended by Israel, like economic development conferences sponsored by the European Union.

A (Baseline) :

- Fact for [UN Climate Conference]: The report's language is far more constrained than that, reflecting a delicate consensus that was reached only after months of debate and several rounds of comments by hundreds of scientists and government climate experts, Trenberth said.
- Fact for [Arab League Summit meeting in Cairo]: The Cairo gathering, the first Arab League summit meeting in a decade to include Iraq, brought together the divergent trends in the Arab world.
- Fact for [UN Climate Conference]: But for the last decade, carbon dioxide has been the main target of climate experts and diplomats trying to stabilize climate.
- Fact for [Arab League Summit meeting in Cairo]: But the presidents, princes and kings who lead the 22 member nations of the Arab League are also under exceptional public pressure to produce more at their meeting than their customary words of support for Palestinian self-determination and statehood.

Q Compare [India Gujarat State] and [El Salvador] with respect to [earthquakes] .

A (Semantic+RSR) :

- Comparable facts:

  [India Gujarat State]: Friday's magnitude-7.9 *quake* flattened two towns in India's western Gujarat state.

  [El Salvador]: The magnitude-7.6 *quake* Saturday left more than 200 people dead and some 1,200 missing.

- Fact for [India Gujarat State]: Indian Prime Minister Atal Bihari Vajpayee, who was to tour *quake* sites Monday, appealed for Indians to contribute, "no matter how small the amount," to help earthquake victims.

- Fact for [El Salvador]: Although the largest number of deaths appeared to be in Las Colinas, the *quake* caused 185 landslides across El Salvador, *burying* anyone who couldn't get out of the way fast enough and blocking hundreds of roads.

A (Semantic) :

- Fact for [India Gujarat State]: The first plane was to land in Ahmedabad, the commercial capital of India' Gujarat state, which was the region hardest hit by the earthquake.

- Fact for [El Salvador]: Above the tons of *dirt* covering her, firefighters worked furiously to free her, hammering away *concrete* to create a narrow hole and feeding an IV tube through in a desperate rescue effort following the earthquake in El Salvador.

- Comparable facts:

  [India Gujarat State]: Aid officials need up to 30,000 tents for those left homeless by the 7.9 magnitude *quake*, said Home Minister Haren Pandya of Gujarat state, which was devastated by Friday's *quake*.

  [El Salvador]: The *quake* destroyed 141,000 Salvadoran homes and left approximately 750,000 people homeless.

A (Baseline) :

- Fact for [India Gujarat State]: The first plane was to land in Ahmedabad, the commercial capital of India' Gujarat state, which was the region hardest hit by the earthquake.
- Comparable facts:

  [India Gujarat State]: Then Barney, a nine-year-old cocker spaniel who has worked in disaster zones from Turkey to Taiwan, Mozambique and El Salvador, is sent in to search for survivors.

  [El Salvador]: SAN SALVADOR, El Salvador – Hopes dwindled Monday for finding more survivors trapped under tons of earth in a suburb here after the earthquake on Saturday set off a landslide.
- Fact for [India Gujarat State]: All the deaths in India were recorded in Gujarat state.

Q Compare [Oil Tanker Jessica Galapagos Spill] versus [Russian Nuclear Submarine Kursk Sinks] with respect to [explanation].

A (Semantic+RSR) :

- Fact for [Oil Tanker Jessica Galapagos Spill]: Workers initially **thought** the ship had leaked machinery oil, “but by the afternoon it was confirmed that it was the fuel the barge was carrying and that it was spilling,” Galapagos National Park Director Eliecer Cruz said.
- Fact for [Russian Nuclear Submarine Kursk Sinks]: The controversy over what **caused** the sinking of the Russian nuclear submarine Kursk bubbled to the surface again, with Russia's top soldier saying NATO was still a suspect and the alliance's military chief again **denying** any NATO involvement.
- Fact for [Oil Tanker Jessica Galapagos Spill]: An estimated 15,000 to 20,000 gallons of additional fuel was spilled late Tuesday, *apparently* after pounding surf **caused** new breaks in the hull of the tanker Jessica, officials said.
- Fact for [Russian Nuclear Submarine Kursk Sinks]: Russian news outlets reported shortly after the disaster that debris from a foreign vessel had been spotted

near the Kursk, lending credence to that **theory**.

A (Semantic) :

- Fact for [Oil Tanker Jessica Galapagos Spill]: Ecuador is asking U.S. help in cleaning up a diesel fuel spill in the ecologically fragile Galapagos Islands, an official said Friday.
- Fact for [Russian Nuclear Submarine Kursk Sinks] The controversy over what caused the sinking of the Russian nuclear submarine Kursk bubbled to the surface again, with Russia's top soldier saying NATO was still a suspect and the alliance's military chief again denying any NATO involvement.
- Fact for [Oil Tanker Jessica Galapagos Spill]: But pounding surf prompted suspension of recovery operations after a giant wave washed over the Jessica's *bow*, injuring the tanker's captain and bruising several Ecuadorean recovery workers. (doc 8)
- Fact for [Russian Nuclear Submarine Kursk Sinks]: A popular Russian radio station runs a brief program that highlights frequently mispronounced Russian words and *tells* listeners how to say them correctly.

A (Baseline) :

- Fact for [Oil Tanker Jessica Galapagos Spill]: Officials and volunteers for the Galapagos Biological Marine Reserve raced against the clock and nature Sunday to try to keep an oil spill of an estimated 144,000 gallons from turning into an environmental disaster.
- Fact for [Russian Nuclear Submarine Kursk Sinks]: The controversy over what caused the sinking of the Russian nuclear submarine Kursk bubbled to the surface again, with Russia's top soldier saying NATO was still a suspect and the alliance's military chief again denying any NATO involvement.
- Fact for [Oil Tanker Jessica Galapagos Spill]: The Ecuadorean tanker Jessica, loaded with 240,000 gallons of oil, rammed into a reef on Tuesday night 800 yards off San Cristobal Island, part of the Galapagos.

- Fact for [Russian Nuclear Submarine Kursk Sinks]: Russian news outlets reported shortly after the disaster that debris from a foreign vessel had been spotted near the Kursk, lending credence to that theory.

Q Compare [Kobe charged with sexual assault] and [Edmond Pope Convicted for Espionage in Russia] with respect to [prosecution].

A (Semantic and Semantic+RSR) : (Both methods produce the same answer in this case.)

- Comparable facts for:

  [Kobe charged with sexual assault]: Now he faces a Class 3 felony sexual assault charge that carries a possible sentence of four years to life if he is *convicted*.

  [Edmond Pope Convicted for Espionage in Russia]: If *convicted* in a secret *trial*, he could be *sentenced* to up to 20 years in prison.

- Fact for [Kobe charged with sexual assault]: LOS ANGELES - Regardless of the outcome of his *criminal* case, Lakers star Kobe Bryant could foul out in the endorsement world after Friday's sexual-assault charge and **acknowledgment** he has committed adultery.

- Fact for [Edmond Pope Convicted for Espionage in Russia]: The *verdict* marked the first espionage *conviction* against a Westerner in Russia in the decade since the end of the Cold War, though both Moscow and Washington have expelled alleged spies from each other's capitals in recent years.

A (Baseline) :

- Fact for [Kobe charged with sexual assault]: Authorities looking to arrest NBA star Kobe Bryant on a sexual assault charge also wanted to charge him with false imprisonment, according to a copy of the arrest warrant.

- Fact for [Edmond Pope Convicted for Espionage in Russia]: U.S. businessman Edmond Pope, convicted of espionage in Russia, asked President Vladimir Putin to pardon him Thursday, and a presidential commission was set to hear his request, Putin's press service said.

- Fact for [Kobe charged with sexual assault]: Bryant was charged with a single count of felony sexual assault for allegedly raping an employee at a Colorado resort where he was a guest June 30.
- Fact for [Edmond Pope Convicted for Espionage in Russia]: The verdict marked the first espionage conviction against a Westerner in Russia in the decade since the end of the Cold War, though both Moscow and Washington have expelled alleged spies from each other's capitals in recent years.

Q Compare [Korea] and [France] with respect to [economic outlook]. (**Uses Open Document Retrieval**)

A (Semantic+RSR) :

- Fact for [Korea]: The sovereign outlook for South Korea remains *stable*, the credit appraiser said, reaffirming its A+ long-**term** local, A- long-**term** foreign, A-1 short-**term** local, and A-2 short-**term** foreign *currency* sovereign credit ratings.
- Fact for [France]: Investors were trying to gauge prospects for the *global* economy now that the war in Iraq has ended, with a recent stream of economic data and forecasts providing **somewhat** mixed signals.
- Fact for [Korea]: "If this **trend** persists and developing Asia continues to focus on policies to achieve faster growth in *domestic* demand, Asia's economic outlook would, over the time, become less vulnerable to economic swings in industrial countries," Ali said.
- Comparable facts:

  [Korea]: "An improving *global* outlook, prudent macro economic management and **strong** growth by China, spurring an increase in intra- regional trade, are among the main reasons for Asia's resilient performance," Ali said.

[France]: The IMF **team** said it believed there was a **strong** case for producing a multi-year spending outlook, if possible starting next year, based on prudent economic growth forecasts.

A (Semantic) :

- Fact for [korea]: Rising household *debt* and falling confidence tied to the *crisis* over North Korea's nuclear ambitions have *weakened domestic* demand in South Korea, it said.
- Comparable facts:

  [korea]: The International *Monetary* Fund said in its outlook Wednesday that South Korea's economic growth would reach five percent this year.

  [france]: The data rounded off a gloomy *weak* for the French economy with the International *Monetary* Fund cutting on Monday its growth forecast for this year to 0.8 percent from 1.2 percent.
- Fact for [korea]: "If this trend persists and developing Asia continues to focus on policies to achieve faster growth in *domestic* demand, Asia's economic outlook would, over the time, become less vulnerable to economic swings in industrial countries," Ali said.

A (Baseline) :

- Comparable facts:

  [korea]: The International Monetary Fund said in its outlook Wednesday that South Korea's economic growth would reach five percent this year.

  [france]: He said the growth outlook for the remainder of the year in France was "not great".
- Fact for [korea]: "If this trend persists and developing Asia continues to focus on policies to achieve faster growth in domestic demand, Asia's economic outlook would, over the time, become less vulnerable to economic swings in industrial countries," Ali said.

- Fact for [france]: The French export credit agency COFACE hailed Wednesday economic improvements in the United States and Brazil but was downbeat about the economic prospects for France and Germany.

C.3 Evaluation Instructions

An overview of the evaluation, as well as sample evaluation screens, are presented in Chapter 5. However, we include here for completeness the precise evaluation instructions exactly as they were given to users. As explained in the chapter, users are assigned to rate either Explanation or Comparison questions; thus there are two different sets of instructions depending on which group a given user was placed in. (The original instructions were formatted in HTML for display in a Web browser; bulleting is preserved here and headings are indicated with underlining.)

C.3.1 Explanation Rating Instructions

```
Survey Overview
---------------


In this survey you will be asked to provide feedback on
automatically-generated answers to questions.


These questions ask about topics in the news. More specifically, they
seek information about causes and effects with respect to these
topics.


For example, you will evaluate answers to questions like: "Describe
[causes of winning ] in [ the 2006 World Cup ]." In order to evaluate
answers to these questions, you should consider the key question
components:
```

* TOPIC ("the 2006 World Cup") : The third bracketed part of the question is the TOPIC. The TOPIC specifies broadly the event(s) or issue(s) with which the question is concerned, in this case the 2006 World Cup.

* FOCUS ("causes of winning"): The second bracketed part of the question is its FOCUS. This specifies narrows the area of interest within the TOPIC. In this case, we are concerned with causes of winning. Alternately, questions may ask about effects instead of causes.

Judging Answers

You will be asked to judge the quality of proposed answers to these cause/effect questions. Note that we are evaluating various question-answering methods to determine which is most successful, so you should be careful and consistent (but not charitable!) in your judgements.

Each answer will be composed of several sentences extracted from news articles. You will be asked to rate the relevance of each sentence into one of four categories. We specify here the relevance criteria for each category, and provide below several example answer judgements.

Rating Categories and Criteria

----

* Relevant: the sentence is about the TOPIC and provides "cause information" (or "effect information") as specified in the FOCUS. For cause questions, "cause information" is defined as information about probable causes or contributors as specified by the FOCUS. For effect questions, "effect information" is information about probable effects or results as specified by the FOCUS.

* Topic only: the sentence is about the TOPIC, but does not provide cause (or effect) information as specified in the FOCUS.

* Not relevant: the sentence is not about the TOPIC at all.

* Ambiguous: because of ambiguity in the sentence, question, or otherwise, you cannot assign one of the above categories. (You will be asked to provide a brief explanation.)

You may sometimes feel unsure about what judgement is appropriate for a given answer sentence. In such cases, you should consider the following options:

* Making "reasonable" assumptions/inferences: In almost every judgement, you will need to make some level of assumption or inference. While this is a subjective process, your general guideline for what counts as "reasonable" should be whether an assumption/inference is substantially more likely than not.

* Using context information: If a given sentence is hard to interpret out of the context of its original document, you may use the

the "show context" links to clarify the sentence meaning.

* Classifying as "ambiguous": If neither of the above possibilities clarifies the judgement, you should classify the sentence as "ambiguous" and provide a brief explanation of the ambiguity.

Example Judgements

----

* Question: Describe [causes of winning] in [2006 World Cup] .
* Proposed Answer Sentences:

o Portugal won the match with superior play throughout the second half.

+ Judgement: Relevant
+ Reasons:

# On-TOPIC? Yes, we can reasonably assume that the match being discussed is part of the 2006 cup.

# On-FOCUS? Yes, "superior play" is a probable cause or contributor to winning (and in this case is explicitly linked to a victory).

o France showed resilient spirit throughout the 2006 tournament.

```
      + Judgement: Relevant
      + Reasons:

            # On-TOPIC? Yes, we can reasonably infer that
              "the 2006 tournament" being discussed is the
              "2006 World Cup" of the TOPIC.

            # On-FOCUS Yes, "resilient spirit" meets the
              criterion of being a probable cause or
              contributor to winning. Note that it is not
              necessary for the sentence to explicitly
              mention winning, or excplicitly link
              "resilient spirit" and winning. It is
              sufficient that the sentence discusses
              probable causes and contributors to winning.

o Ireland took the field in green uniforms featuring a
leprechaun mascot.

      + Judgement: Topic Only
      + Reasons:

            # On-TOPIC? Yes, we can assume that it is
              substantially likely that this sentence is
              describing the 2006 cup.

            # On-FOCUS? No, despite the charm of
              leprechauns, the uniforms are not a "probable
              cause or contributor" to winning.
```

```
o Ireland took the field in green uniforms featuring a
leprechaun mascot, and rallied behind a supportive crowd to
defeat Portugal.

      + Judgement: Relevant
      + Reasons:

            # On-TOPIC? Yes, we can assume that it is
              substantially likely that this sentence is
              describing the 2006 cup.

            # On-FOCUS? Yes. Note that this is true even
              though the first part of the sentence (about
              the uniforms) is not cause-relevant. It is
              sufficient that some part of the sentence, in
              this case the clause describing crowd support,
              describes a cause of winning.

o Germany demonstrated strong offensive play in the 2003
European Tournament.

      + Judgement: Not relevant.
      + Reasons:

            # On-TOPIC? No, this sentence does not discuss
              the 2006 cup.

            # On-FOCUS? N/A, we are only concerned with the
              FOCUS in terms of the relevant TOPIC.
      o These issues came to the fore in the Cameroon - Argentina
      game on Saturday.

            + Judgement: ?? (Use "show context" to clarify.)
            + Reasons:

                  # On-TOPIC. Yes.

                  # On-FOCUS. Unclear. The "issues" mentioned may
                    or may not be "cause information." It would be
                    appropriate here to use the "show context"
                    option to determine this. If the context were
                    to show that the "issues" were, say "brilliant
                    play despite a lack of conditioning", then the
                    sentence would be "Relevant." If, instead, the
                    issues were "crowd control difficulty", then
                    "Topic only" would be the correct
                    judgement. If the context does not clarify
                    what the "issues" are, the rating should be
                    "Ambiguous" (accompanied by an explanation
                    like "unclear which 'issues' are referred
                    to").

* Question: Describe [effects of conference ] in [ Conference on
            Water-Ski Safety ] .
* Proposed Answer Sentences:

      o The delegation from Gambia drafted a resolution to
      strengthen ski curvature regulations.
            + Judgement: Relevant
            + Reasons:

                  # On-TOPIC? Yes, we can reasonably assume the
                    resolution is part of the conference.

                  # On-FOCUS? Yes. While the resolution by Gambia is
                    not explicitly stated to result from the
                    conference, but it meets the "probable effect
                    or result" standard.

      o At the conference last year, outboard motors were among
      the hot topics.

            + Judgement: Topic only.
            + Reasons:

                  # On-TOPIC? Yes. Note that the question does not
                    specify a particular conference year. So if
                    there appears to have been more than one
                    conference on water-ski safety, information
                    which relates to any one of them will be
                    on-TOPIC.

                  # On-FOCUS? No, since a subject of interest at
                    the conference is not a clear result or effect
                    of the conference.

      o Several luminaries of the water-skiing world boycotted,
      protesting a failure to address steroid testing at the
```

```
          conference.

                + Judgement: Relevant.
                + Reasons:

                      # On-TOPIC? Yes, we can reasonably assume this
                        is talking about the attendance of the
                        conference.

                      # On-FOCUS? Yes. The boycott/protest happens
                        because of the conference.
```

C.3.2 Comparison Rating Instructions

```
Survey Overview
---------------

In this survey you will be asked to provide feedback on
automatically-generated descriptions of topics in the news.

These descriptions are meant to help with the task of comparing two
distinct news topics according to some common theme. The desriptions
need not explicitly make comparisons themselves, but should provide
information which can easily be used in comparing two topics.

Comparitive Queries
-------------------

The comparitive descriptions are presented in response to queries
like: "Find information for comparing [ Bush gives State of Union
```

Address] and [European Union Summit] with respect to [economic outlook]."

These queries contain the following key components:

* TOPICS ("Bush gives State of Union Address", "European Union Summit"): The first two bracketed parts of the query specify two distinct topics. These are the topics which the response whould help you to compare.

* FOCUS ("economic outlook"): The last bracketed part of the question specifies the focus of the comparison you wish to make. This narrows the relevant areas for comparison of the topics. In this case, we are seeking information which will compare issues of economic outlook across the two topics. Therefore, information about, e.g., inflation rates is more relevant for the comparison than, e.g., information on primary school education standards.

Judgments

For each query, you will be presented with two responses and asked to indicate which one you prefer. Even if your preference is slight, you should indicate it; only rate the responses "equal" if you truly have no preference (or if they are actually the same; in some cases you may be presented with a pair of identical answers).

In determining which response is better, the core criterion should be: Given the task of creating a comparison of the two topics with respect to the focus, which response provides more useful information? While this is a subjective question, you may wish to consider the following points:

* TOPIC coverage: Does the response provide information on both of the the topics? For instance, in the example above, an acceptable response should contain information from both the Bush speech and the E.U. summit.

* FOCUS coverage: Does the response provide information which allows you to compare or contrast with regard to the question focus? For instance, for the "economic outlook" focus in the above example, such information might include mentions of key economic concerns or statistics.

* Comparability: If two responses both contain information that is relevant in terms of topic and focus, your preference should reflect whether the information provided by a given response lends itself more easily to making a comparison.

One consideration here is whether a response contains matching or similar information for both topics. In terms of our example, this could occur if, for instance, a response contains information that interest rates are falling in the E.U., as well as information that interest rates are stable in the U.S.

However, keep in mind that a comparison can still be made even if such neatly matching information is not present. For instance, if a response includes inflation rate information for the E.U. but focuses on energy issues in describing the Bush speech, this could support a comparison statement like: "In the E.U., inflation is a major concern, whereas the U.S. president focused on oil dependence." Thus, even when a precise point-for-point comparison is not supported, you should consider whether the information in the response could still form a comparison of this type. Of course, many kinds of information can be compared, and your preference in this case should reflect how good, or useful, the resulting comparison would be. That is, information on oil dependence has strong links to the focus of our example question ("economic outlook"); a more obscure bit of information, e.g. tax credits for haunted house owners, would be less useful in comparing the topics.

Bibliography

Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*.

Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., and Caputo, D. (1999). Topic-based novelty detection 1999 summer workshop at clsp. Technical Report Final Report, Johns Hopkins University.

Asher, N. and Lascarides, A. (2003). *The logic of conversation*. Cambridge University Press.

Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL Intelligent Scalable Text Summarization Workshop*.

Barzilay, R., Elhadad, N., and McKeown, K. R. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Barzilay, R. and Lapata, M. (2005). Modeling local coherence: An entity-based approach. In *ACL 2005*.

Barzilay, R. and Lee, L. (2004). Catching the drift: Probabilistic content models, with applications to generation and summarization. In *NAACL-HLT*.

Barzilay, R., McKeown, K. R., and Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In *Proc. of 37th ACL*, pages 550–557.

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*.

Bethard, S. (2005). Extracting eventive, temporal and causal structure for automated question answering. PhD thesis proposal, University of Colorado at Boulder, Department of Computer Science.

Blair-Goldensohn, S. (2005). From definitions to complex topics: Columbia university at DUC 2005. In *5th Document Understanding Conference (DUC 2004) at HLT/NAACL 2004*.

Blair-Goldensohn, S., Evans, D., Hatzivassiloglou, V., McKeown, K., Nenkova, A., Passonneau, R., Schiffman, B., Schlaikjer, A., Siddharthan, A., and Siegelman, S. (2004). Columbia University at DUC 2004. In *4th Document Understanding Conference (DUC 2004) at HLT/NAACL 2004*.

Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., et al. (2001). Issues, tasks and program structures to roadmap research in question & answering (q&a).

Burstein, J., Marcu, D., Andreyev, S., and Chodorow, M. (2001). Towards automatic classification of discourse elements in essays. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2001)*.

Carbonell, J., Harman, D., Hovy, E., Maiorano, S., Prange, J., and Sparck-Jones, K. (2000). Vision statement to guide research in question answering (QA) and text summarization.

Carbonell, J. G. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336.

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

Carlson, L., Marcu, D., and Okurowski, M. E. (2001). Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *2nd SIGDIAL Workshop on Discourse and Dialogue, Eurospeech 2001*.

Chen, F. R., Farahat, A. O., and Brants, T. (2004). Multiple similarity measures and source-pair information in story link detection. In *HLT-NAACL 2004*.

Chen, J. and Rambow, O. (2003). Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Cohen, W. (1995). Fast effective rule induction. In *Proc. of 12th Int'l Conf. on Machine Learning*, pages 115–123.

Collins, M. J. (1996). A new statistical parser based on bigram lexical dependencies. In Joshi, A. and Palmer, M., editors, *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184–191, San Francisco. Morgan Kaufmann Publishers.

Crammer, K. and Singer, Y. (2001). Pranking with ranking. In *Neural Information Processing Systems NIPS 2001*.

Cui, H., Kan, M.-Y., and Chua, T.-S. (2005). Unsupervised learning of soft patterns for generating definitions from online news. In *WWW 2005*.

Dahlgren, K. (1988). *Naive Semantics for Natural Language Understanding*. Kluwer Academic Press.

Dang, H. (2005). Overview of duc 2005. In *Fifth Document Understanding Conference (DUC-05)*.

Daumé III, H. and Marcu, D. (2005). Bayesian multi-document summarization at mse. In *Proceedings of MSE 2005*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society For Information Science*, 41:391–407.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38.

Di Eugenio, B. (2000). On the usage of kappa to evaluate agreement on coding tasks. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*.

Dom, B. (2001). An information-theoretic external cluster-validity measure. Technical Report Research Report 10219, IBM.

Dumais, S. T. (1996). Combining evidence for effective information filtering. In *AAAI Spring Symposium on Machine Learning and Information Retrieval*.

Dunning, T. (1994). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Elhadad, M. and McKeown, K. R. (1990). Generating connectives. In *Proceedings of the 13th International Conference on Computational Linguistics*.

Fellbaum, C., editor (1998). *Wordnet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

Filatova, E. and Prager, J. (2005). Tell me what you do and i'll tell you what you are. In *HLT-NAACL 2005*.

Fillmore, C. J., Baker, C. F., and Sato, H. (2002). The framenet database and software tools. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1157–1160.

Fisher, S. and Roark, B. (2006). Query-focused summarization by sueprvised sentence ranking and skewed word distribution. In *5th Document Understanding Conference (DUC 2006)*.

Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Pricesses*, 25(2-3):285–307.

Galley, M. and McKeown, K. (2007). Lexicalized markov grammars for sentence compression. Under Review.

Galley, M., McKeown, K. R., Fosler-Lussier, E., and Jing, H. (2003). Discourse segmentation of multi-party conversation. In Hinrichs, E. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL–03)*, pages 562–569, Sapporo, Japan.

Garcia, D. (1997). Coatis, an NLP system to locate expressions of actions connected by causality links. In *Proceedings of the 10th European Workshop on Knowledge Acquisition, Modeling and Management*.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Girju, R. (2003). Automatic detection of causal relations for question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003), Workshop on Multilingual Summarization and Question Answering*.

Girju, R., Badulescu, A., and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the Human Language Technology Conference (HLT-2003)*.

Glover, E. J., Pennock, D. M., Lawrence, S., and Krovetz, R. (2002). Inferring hierarchical descriptions. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM'02)*.

Grishman, R. (1997). *Information Extraction: Techniques and Challenges*. Springer-Verlag.

Grosz, B. and Sidner, C. (1986). Attention, intention and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Hachey, B., Murray, G., and Reitter, D. (2005). The embra system at DUC 2005: Query-oriented multi-document summarization with a very large latent semantic space. In *Document Understanding Conference*.

Hacioglu, K., Pradhan, S., Ward, W., Martin, J., and Jurafsky, D. (2004). Semantic role labeling by tagging syntactic chunks. In *Proceedings of CoNLL 2004 Shared Task*.

Harabagiu, S. and Maiorano, S. (1999). Finding answers in large collections of texts : Paragraph indexing + abductive inference.

Harman, D., editor (2002). *AQUAINT R&D Program 12 Month Workshop*, Arlington, VA. ARDA and NIST.

Hatzivassiloglou, V., Klavans, J. L., and Eskin, E. (1999). Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212, College Park, Maryland. Association for Computational Linguistics.

Hearst, M. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hearst, M. (1998). Automated discovery of wordnet relations. In Fellbaum, C., editor, *Wordnet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

Higgins, D., Burstein, J., Marcu, D., and Gentile, C. (2004). Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL*, pages pp. 185–192.

Hobbs, J. R. (1979). Coherence and coreference. *Cognitive Science*, 3(1):67–90.

Hobbs, J. R. (1990). Literature and cognition.

Hovy, E. (1993). Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(1-2):341–385.

Hovy, E. and Maier, E. (1993). Parsimonious or profligate: How many and which discourse structure relations? Unpublished Manuscript. Available online at http://www.isi.edu/natural-language/people/hovy/papers/93discproc.pdf.

Hovy, E. H., Hermjakob, U., and Lin, C.-Y. (2001). The use of external knowledge of factoid QA. In *Text REtrieval Conference*.

Jagadeesh, J., Pingali, P., and Varma, V. (2005). A relevance-based language modeling approach to duc 2005. In *Document Understanding Conference*.

Jones, M. P. and Martin, J. H. (1997). Contextual spelling correction using latent semantic analysis. In *5th Conference on Applied Natural Language Processing*.

Katz, B., Lin, J., Stauffer, C., and Grimson, E. (2003). Answering questions about moving objects in surveillance videos. In *AAAI-2003*.

Khoo, C., Chan, S., and Niu, Y. (2000). Extracting causal knowledge from a medical database using graphical patterns. In *ACL 2000*.

Klavans, J. and Muresan, S. (2000). Definder: Rule–based methods for the extraction of medical terminology and their associated definitions from on–line text. In *Proceedings of the AMIA 2000 Annual Symposium*.

Knott, A. and Sanders, T. (1998). The classification of coherence relations and their linguistic markers: An exploration of two languages. *Journal of Pragmatics*, 30(2):135–175.

Krippendorff, K. (1980). *Content Analysis: An Introduction to its Methodology*. Sage Publications.

Kupiec, J., Pedersen, J., and Chen, F. (1997). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR-1997*.

Lacatusu, F., Hickl, A., and Harabagiu, S. (2006). The impact of question decomposition on the quality of answer summaries. In *Seventh Language Resources and Evaluation Conference (LREC 2006)*.

Lakoff, G. (1987). *Women, Fire and Dangerous Things*. University of Chicago Press.

Landauer, T. K. and Dumais, S. (1997). A solution to plato's prolem; the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of ACL-2003*.

Lapata, M. and Lascarides, A. (2004). Inferring sentence-internal temporal relations. In *HLT 2004*.

Lascarides, A. and Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16(5):437–493.

Lee, L. (1999). Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32.

Lester, J. C. and Porter, B. W. (1997). Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101.

Li, X. and Roth, D. (2002). Learning question classifiers. In *COLING 2002*.

Lin, C. and Hovy, E. (2002). Automated multi-document summarization in NeATS. In *Human Language Technology Conference*, pages 50– 53.

Lin, C.-Y. (2004). Rouge: a package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization*.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774.

Lin, D. (1998b). Dependency-based evaluation of minipar. In *Workshop on the Evaluation of Parsing Systems*.

Lin, J. and Demner-Fushman, D. (2005). Automatically evaluating answers to definition questions. In *2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*.

Lowry, R. (2006). Concepts and applications of inferential statistics. Available online: http://faculty.vassar.edu/lowry/webtext.html.

Mani, I. and Bloedorn, E. (1997). Multi-document summarization by graph search and matching. In *Fifteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 622–628.

Mann, W. and Thompson, S. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Marcu, D. (1997). *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. PhD thesis, University of Toronto, Department of Computer Science.

Marcu, D. (2000). The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3).

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Martin, J. (1992). *English Text: System and Structure*. John Benjamins.

Marton, G. and Radul, A. (2006). Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *NAACL-HLT 2006*.

Maskey, S. and Hirschberg, J. (2005). Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Eurospeech 2005*.

Maybury, M. (1989). Enhancing explanation coherence with rhetorical strategies. In *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*.

McKeown, K., Barzilay, R., Chen, J., Elson, D., Evans, D., Klavans, J., Nenkova, A., Schiffman, B., and Sigelman, S. (2003). Columbia's newsblaster: New features and future directions (demo). In *Proceedings of NAACL-HLT'03*.

McKeown, K. R. (1985). *Text generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press.

Metzler, D. and Croft, W. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval*, 40(5):735–750.

Miller, T. (2003). Latent semantic analysis and the construction of coherent extracts. In Angelova, G., Bontcheva, K., Mitkov, R., Nicolov, N., and Nikolov, N., editors, *Proceedings of the International Conference RANLP-2003 (Recent Advances in Natural Language Processing)*, pages 270–277.

Mittal, V. and Paris, C. (1995). Generating explanations in context: The system's perspective. *Expert Systems with Applications*, 8(4).

Moore, J. and Paris, C. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–695.

Moore, J. D. and Pollack, M. E. (1992). A problem for rst: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4):537–544.

Moore, J. D. and Wiemer-Hastings, P. (2003). Discourse in computational linguistics and artificial intelligence. In A. G. Graesser, M. A. G. and Goldman, S. R., editors, *Handbook of Discourse Processes*, pages 439–487. Lawrence Erlbaum Associates.

Morris, J. and Hirst, G. (2004). Non-classical lexical semantic relations. In *NAACL 2004 Workshop on Computational Lexical Semantics*.

Moser, M. G. and Moore, J. D. (1996). Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–420.

Mussweiler, T. (2003). Everything is relative: Comparison processes in social judgment. *European Journal of Social Psychology*, 33:719–733.

Narayanan, S. and Harabagiu, S. (2004). Question answering based on semantic structures. In *20th International Conference on Computational Linguistics (COLING 2004)*.

Nenkova, A. (2006). *Understanding the process of multi-document summarization: content selection, rewriting and evaluation*. PhD thesis, Columbia University, Department of Computer Science.

Nenkova, A. and McKeown, K. (2003). References to named entities: A corpus study. In *NAACL-HLT 2003*. short paper.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: the pyramid method. In *NAACL-HLT 2004*.

Oakes, M. (1998). *Statistics for Corpus Linguistics*. Edinburgh University Press.

Over, P., editor (2004). *Fourth Document Understanding Conference (DUC-04)*. NIST.

Over, P., Ianeva, T., Kraaij, W., and Smeaton, A. (2005). TRECVID 2005: An overview. In *The Fourteenth Text REtrieval Conference (TREC-05)*.

Padó, S. and Lapata, M. (2003). Constructing semantic space models from parsed corpora. In *Proceedings of ACL-03*, Sapporo, Japan.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Popescu, A. (2005). Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP*.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Prager, J., Brown, E., Coden, A., and Radev, D. (2000). Question-answering by predictive annotation. In *SIGIR 2000*.

Prager, J., Chu-Carroll, J., and Czuba, K. (2004). Question answering using constraint satisfaction: Qa-by-dossier-with-contraints. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 574–581, Barcelona, Spain.

Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., and Joshi, A. (2006). The penn discourse treebank 1.0. annotation manual. Technical Report IRCS Technical Report IRCS-06-01, University of Pennsylvania.

Pustejovsky, J. (1991). The generative lexicon. *Computational Linguistics*, 17(4):409–441.

Pustejovsky, J. (1995). *The Generative Lexicon.* MIT Press.

Radev, D., Jing, H., and Budzikowska, M. (2000). Centroid-based summarization of multiple documents. In *Proceedings of ANLP-NAACL workshop on summarization.*

Radev, D. R. and McKeown, K. R. (1998). Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

Rambow, O. (1990). Domain communication knowledge. In *Fifth International Workshop on Natural Language Generation.*

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002).*

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*, chapter 4. Cambridge Univ. Press.

Richardson, S. D., Dolan, W. B., and Vanderwende, L. (1998). MindNet: acquiring and structuring semantic information from text. In *ACL 1998.*

Rosenberg, A. (2006). Proposal for evaluating clusterings. Personal communication.

Sager, J. C. and L'Homme, M. (1994). A model for definition of concepts. *Terminology*, pages 351–374.

Sanders, T., Spooren, W., and Noordman, L. (1992). Towards a taxonomy of coherence relations. *Discourse Processes*, 15(1):1–36.

Sarner, M. and Carberry, S. (1988). A new strategy for providing definitions in task oriented dialogues. In *Proc. Int'l Conf. on Computational Linguistics (COLING-88)*, volume 1.

Schank, R. and Abelson, R. (1977). *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates.

Schapire, R. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Siddharthan, A. (2003). *Syntactic simplification and Text Cohesion*. PhD thesis, University of Cambridge.

Siddharthan, A., Nenkova, A., and McKeown, K. (2004). Syntactic simplification for improving content selection in multi-document summarization. In *20th International Conference on Computational Linguistics (COLING 2004)*.

Silber, H. G. and McCoy, K. F. (2000). Efficient text summarization using lexical chains. In *Intelligent User Interfaces*.

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *NIPS 2005*.

Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia. Association for Computational Linguistics.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT-NAACL03)*.

Sundheim, B. (1992). Overview of the fourth message understanding evaluation and conference. In *Proceedings of the 4th conference on Message understanding*.

Swartz, N. (1997). Definitions, dictionaries and meanings. Posted online http://www.sfu.ca/philosophy/definitn.htm.

Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton, G. (2003). Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR conference on Research and development in information retrieval*.

Tiedemann, J. (2005). Integrating linguistic knowledge in passage retrieval for question answering. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 939–946, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Umbach, C. (2001). Contrast and contrastive topic. In *ESSLLI 2001 Workshop on Information Structure, Discourse Structure and Discourse Semantics*.

Vanderwende, L., Suzuki, H., and Brockett, C. (2006). Microsoft research at duc 2006: Task-focused summarization with sentence simplification and lexical expansion. In *5th Document Understanding Conference (DUC 2006)*.

Verberne, S. (2006). Developing an approach for why-question answering. In *Conference Companion of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*.

Verberne, S., Boves, L., Oostdijk, N., and Coppen, P.-A. (2006). Data for question answering: the case of why. In *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC 2006)*.

Verberne, S., Boves, L., Oostdijk, N., and Coppen, P.-A. (2007). A discourse-based system for answering why-questions. Under Review.

Voorhees, E. and Buckland, L. P., editors (2003). *Proceedings of The Fourteenth Text REtrieval Conference (TREC-03)*. NIST Special Publications.

Voorhees, E. and Buckland, L. P., editors (2005). *Proceedings of The Fourteenth Text REtrieval Conference (TREC-05)*. NIST Special Publications.

Wayne, C. (2000). Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *Second International Language Resources and Evaluation Conference*.

Webber, B., Joshi, A., Miltsakaki, E., Prasad, R., Dinesh, N., Lee, A., and Forbes, K. (2005). A short introduction to the penn discourse treebank. In *Copenhagen Working Papers in Language and Speech Processing*.

Weischedel, R., Xu, J., and Licuanan, A. (2004). A hybrid approach to answering biographical questions. In Maybury, M., editor, *New Directions In Question Answering*, chapter 5. AAAI Press.

Widdows, D. (2003). Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *HLT/NAACL 2003*.

Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*.

Xu, J., Weischedel, R., and Licuanan, A. (2004). Evaluation of an extraction-based approach to answering definitional questions. In *SIGIR 2004*.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP'03*.

Zhou, M. and Aggarwal, V. (2004). An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces.