Information Fusion for Multidocument Summarization: Paraphrasing and Generation

Regina Barzilay

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2003

Information Fusion for Multidocument Summarization: Paraphrasing and Generation

Regina Barzilay

The number and variety of online news sources makes it difficult for people to track the news concerning even a single event. Redundancy causes such tracking to be extremely time-consuming: multiple news feeds on the same event tend to contain similar information. A summary of such news feeds can present important information in one short text, dramatically reducing reading time. The focus of this thesis is *information fusion*, a technique which, given multiple documents, identifies redundant information and synthesizes a coherent summary. This technique is embodied in MultiGen, a system that I have designed, implemented and evaluated over the course of my Ph.D. Unlike previous work in the area, MultiGen is a domain-independent system: it generates news summaries on a variety of topics in any domain. Another contribution to the state of the art is that the system generates the summary by reusing and altering phrases from the input articles, creating a

more fluent and cohesive text. This is in contrast with other existing systems, which simply extract sentences from input articles and concatenate them together, leading to fluency problems. Currently MultiGen operates as part of Columbia's Newsblaster system. Everyday, Newsblaster downloads all news articles from a variety of sources, clusters articles by topic, and generates a cohesive, readable automatic summary of each document cluster.

One key challenge in multidocument summarization is eliminating redundant information in the produced summaries. Articles about the same event often contain descriptions of the same fact using different wording. To address this issue, we need a method to identify paraphrases — fragments of text that convey similar meaning even if they are not identical in wording. Automatic identification of paraphrases was not addressed in previous research, although it is necessary for many applications, including question-answering, information extraction and natural language generation. This thesis presents unsupervised learning techniques to identify paraphrases given a corpus of multiple parallel texts. This type of corpus provides many instances of paraphrasing, because these texts preserve the meaning of the original source, but may use different words to convey the meaning. Both the data and the method are departures from past approaches to corpus based techniques. Our evaluation experiments show that the algorithm extracts paraphrases with high accuracy and significantly outperforms a state of the art algorithm developed for related tasks in machine translation.

Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to thank my advisor, Kathy McKeown: she was a role model in many aspects and had prepared me for academic life. Kathy always encouraged me to look at the most unexpected directions, and always made me believe that I could succeed. Working with her, I learned the value of a vision, and persistence in materializing it.

Thanks also to my committee members: Michael Collins, Julia Hirschberg, Lillian Lee and Shree Nayar, for reading my thesis and providing valuable feedback.

Lillian Lee taught me the basics of conducting empirical research and greatly influenced this work. Her dedication to research has inspired me. Thanks to my master thesis advisor, Michael Elhadad, who introduced me to the area of natural language processing. I owe my initial interest in the area to him, and even from far away, Michael remains a dear friend.

Noemie Elhadad and Smaranda Muresan have been great colleages as well as close friends. They have read most of my thesis and provided many detailed and challenging comments. Their friendship, kindness and support made my graduate life much happier. My first officemates, Shimei Pan and James Shaw, introduced me to life in Columbia; many ideas that led to my thesis work originated in our discussions.

Luis Gravano (a.k.a Professor Gravano) was an immense source of wisdom and advice on every aspect of academic life and more. My friends Cohavit Tabouch and Samir Genaim were punished by hearing much more about NLP research that they ever wanted to. I spent many fun hours with Yael Netzer, Mirella Lapata and Simone Teufel talking about research and other interesting things. Hubie Chen is a very special friend whose wit enlightened my Ithaca days. I appreciate his help in proof-reading this text and others.

I would like to thank the past and present members of the Columbia NLP group. During my years in Columbia, the NLP group tripled in size without loosing its family atmosphere. Noisy group meetings, weekend lunches and hectic demo preparations conducted by Vasileios Hatzivassiloglou will be a source of nostalgic moments for years to come.

I am thankful to Robert Constable for hosting me in Cornell during two years of my PhD. I enjoyed the spirit of the local NLP and Nuprl groups, and had many inspiring conversations with Stuart Allen, Claire Cardie, Juanita Heyerman, Klara Kedem, Oren Kurland, Vincent Ng and Bo Pang.

I want to thank my family for their love and support, especially my parents Ala and Yaakov, my grandmother Genya, and my mother-in-law Mazal. Our cats, Gershon and Nehama had a special contribution to my well-being.

Finally, I would like to acknowledge my husband, Eli. Without his encouragement I would never have began, and much less completed this thesis.

To Tali (1970–1994) and Eli

Chapter 1

Introduction

"A twenty-something bus rider accused another passenger of trampling his feet. Later, in a park, a friend encourages the same man to reorganize the buttons on his coat." In Queneau's "Exercises in Style", this singularly pointless scenario unfolds 99 times, each time in a different style. The story is told as an official letter, as a series of exclamations and as a sonnet. Figure 1.1 shows three sample beginnings of this story. While these sentences barely share any words in common, all three convey basically the same information.

The multitude of versions of this one story seems unique, but every day we are surrounded with numerous information sources conveying the same messages. For instance, a typical query to a search engine returns documents stating similar facts. Also, news articles from different sources on the same event often cover the same information. A search engine user aiming to find relevant documents as well as a person who follows an event through several news-wires are each forced to read the same information over and over again as a result of this redundancy. Moreover, in today's information age, we are bombarded with huge amounts of information —

One day at about midday in the Parc Monceau district, on the back platform of a more or less full S bus (now #84), I observed a person with a very long neck who was wearing a felt hat which had a plaited cord around it instead of a ribbon. This individual suddenly addressed the man standing next to him, accusing him of purposely treading on his toes every time any passenger got on or off. However, he quickly abandoned the dispute and threw himself on to a seat which had become vacant. Two hours later I saw him in front of the gare Saint-Lazare engaged in earnest conversation with a friend who was advising him to reduce the space between the lapels of his overcoat by getting a competent tailor to raise the top button.
In a bus of the S-line, 10m long, 3 wide, 6 high, at 3.6km from its starting point, loaded with 48 people, at 12:17pm, a person of the masculine sex aged 27 years 3 months and 8 days, 1,72m tall and weighing 65kg and wearing a hat 35cm in height round the crown of which was a ribbon 60cm long, interpellated a man aged 48 years 4 months and 3 days, 1,68m tall and weighing 77kg, by means of 14 words whose enunciation lasted 5 seconds and which alluded to some involuntary displacements of from 15 to 20mm. Then he went and sat down about 110cm away. 57mn later he was 10m away from the suburban entrance to the gare Saint-Lazare and was walking up and down over a distance of 30m with a friend aged 28, 1,70m tall and weighing 71kg who advised him in 15 words to move by 5cm in the direction of the zenith a button which was 3cm in diameter.
I got into a bus full of taxpayers who were giving some money to a taxpayer who had on his taxpayer's stomach a little box which allowed the other taxpayers to continue their taxpayers' journey. I noticed in this bus a taxpayer with a long taxpayer's neck and whose taxpayer's head bore a taxpayer's felt hat encircled by a plait the like of which no taxpayer ever wore before. Suddenly the said taxpayer, peremptorily addressed a nearby taxpayer complaining bitterly that he was purposely treading on his taxpayer's toes every time other taxpayers got on or off the taxpayers' bus. Then the angry taxpayer went and sat down in a seat for taxpayers which another taxpayer had just vacated

Figure 1.1: Extracts from "Exercises in Style"

merely 10 years ago, a question like "Where else did I see that actor?" would likely have been answered using personal memories, whereas today, one can easily retrieve a non-trivial amount of information using a few quick clicks. This implies a genuine need for tools that help us to cope with large quantities of information. Clearly, it would be highly desirable to have a mechanism for producing a summary containing common information, given multiple related documents.

Today, the predominant method for summarization is sentence extraction, in which sentences are extracted verbatim from input articles and concatenated to form a summary. This technique is not effective in a multi-document scenario. For example, an extracted sentence may include details specific to the article from which it came, leading to source bias. Attempting to solve this problem by including more sentences might lead to a verbose and repetitive summary. Furthermore, fluency, a usual concern with extraction methods, is significantly aggravated by pasting together text fragments from multiple sources. In short, a generated summary would have numerous advantages over an extracted one.

These considerations suggest that methods developed in the area of Natural Language Generation could be relevant for the summarization task. These methods provide a mechanism for generating text from an underlying semantic representation. For example, a typical generation application might create financial reports given a database of stock market transactions. A generation system has to make decisions at various levels of text representation, from content planning to surface realization. Content planning determines which semantic concepts to include in the output text and how to organize them. Surface realization determines how to map content units into sentences by selecting appropriate syntactic structures and words. A significant

limitation of existing concept-to-text generation is its domain-dependence: criteria for content selection are determined by domain-specific pragmatic constraints, while surface realization requires a domain-specific dictionary for translating semantic concepts into an appropriate linguistic representation.

Can we use strategies developed for concept-to-text generation when generating summary sentences? Since summarization systems operate over textual input, direct applying these methods would require translating the input text into a semantic representation. Not only does such translation extend well beyond the ability of current analysis methods, but the type of resources required by current generation systems can be obtained only for limited domains. In addition, any method developed for concept-to-text generation does not use text readily available as input, but instead generates every phrase from scratch. These considerations suggest that we need a new method for the summary rewriting task. Ideally, such a method would not require full semantic representation, but would rather rely on input texts and knowledge that can be automatically derived from text to generate a summary by reusing and altering phrases from the input articles.

In this thesis, we focus on **information fusion**, a method for generating multi-document summaries. More specifically, this method creates a summary by synthesizing common information across input documents into a coherent text. Such a technique is suitable for multi-document summarization, since repetition of information among related sources is an indication of its importance. The key challenges of information fusion are identification of common information and combination of such information into a text, which are performed without having access to the full semantic representation of the input texts. Unlike traditional generation, content

planning for multidocument summarization operates over full sentences; it compares the predicate-argument structures of the input sentences and produces sets of sentence fragments which convey common information. The principal method we use for this task is alignment of predicate-argument structures. The task of surface realization in this scenario is to produce fluent sentences by combining these phrases and arranging them in novel contexts. We hypothesize that the lack of an elaborate semantic representation, which is required for traditional generation, can be compensated for by the knowledge automatically extracted from the input documents and a large text corpus, which provides clues as to how to constrain the number of ways the text fragments can be combined.

A principal source of knowledge required for information fusion is paraphrasing information, since related documents often contain descriptions of the same fact using different wording (e.g., "*Anti-terror tactics are defended by Ashcroft*" and "*Ashcroft supports anti-terrorism policies*"). Therefore, the accuracy of the content planner crucially depends on the ability to match fragments of text that convey similar meaning but are not identical in wording. As we show in chapter 2, existing thesauri do not provide sufficient coverage of paraphrases; moreover, they do not include phrasal and structural paraphrases. Although automatic identification of paraphrases can benefit many applications, including question-answering, information extraction and natural language generation, this was not a topic of previous research. In this thesis, we propose a method for paraphrase acquisition from a large body of text and use derived paraphrases in the information fusion algorithm.

Information fusion is embodied in the multi-document summarization system MultiGen, which was used as a testbed for the methods developed within this thesis.

We describe MultiGen in the next section.

1.1 MultiGen

MultiGen is a multi-document summarization system that automatically generates a concise summary of related documents by identifying similarities among them. The typical input to the system is a set of articles about the same event produced by different news agencies. Figure 1.2 shows an example of a summary produced by MultiGen (the input articles are available in Appendix A). MultiGen has two major components — an analysis component and a fusion component (see Figure 1.3). Our contribution is the fusion component. Below we briefly describe each of these two components.

> At least 30 people in Uganda have died of a hemorrhagic fever that authorities fear may be caused by the Ebola virus. Blood samples have been sent to South Africa and the United States. The World Health Organization have sent fact-finding missions to Gulu to investigate the outbreak. 10 people have died in hospital, including 3 nurses treating the sick. Symptoms include fever, muscle pains and bleeding from the mouth, nose and anus. One of the first victims, who died on 17 Sep 2000, was reported to be a soldier.

Figure 1.2: A summary of three articles produced by MultiGen

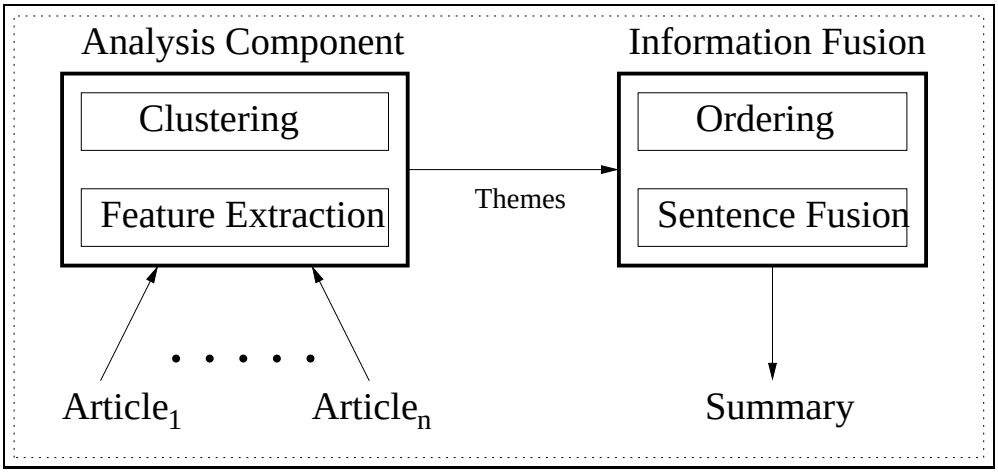

Figure 1.3: MultiGen architecture

1.1.1 Analysis Component: Simfinder

First, the system identifies *themes*, groups of sentence units from different documents that contain repeated information. Each theme corresponds to one sentence in the output summary. An example of a theme is shown in Figure 1.4. There may be many themes for a set of articles; for the articles from Appendix A, there are 7 themes shown in Appendix B. The analysis component, Simfinder (Hatzivassiloglou, Klavans, and Eskin, 1999), computes similarity among sentences from input texts and then clusters them into themes[1]. Unlike most systems that compute a measure of similarity over text, features used by Simfinder extend beyond a simple word count and include noun phrase, proper noun, and semantic sense overlap; it also utilizes positional and relational information between pairs of words. Combination of features is performed using log-linear regression, which has been trained on a large manually-marked set of sentence pairs.

[1] This component is not within the scope of this thesis.

On 13 Oct 2000, it was reported that at least 30 people in northern Uganda town have died in recent weeks of a hemorrhagic fever that authorities fear may be caused by the Ebola or Marburg virus.
About 30 people have died from an as-yet unidentified disease in northern Uganda.
Ugandan health authorities are battling to contain an outbreak suspected to be caused by the deadly Ebola virus which has killed at least 31 people in the north of the country.

Figure 1.4: A collection of similar sentences (part of a *theme*).

1.1.2 Fusion Component

The fusion component aims to create a coherent text from a set of themes computed by the analysis component described above. This process consists of two stages: sentence fusion and sentence ordering.

- **Sentence fusion** Given a theme extracted from the articles, shown in Figure 1.4, how can we determine that only the sentence "At least 30 people in northern Uganda have died of a hemorrhagic fever that authorities fear may be caused by the Ebola virus." should be represented in the summary? We have developed a novel algorithm for this task which analyzes the grammatical structure of each theme sentence with an off-the-shelf parsing tool. Our information fusion algorithm aligns predicate argument structures of the sentences within each theme to determine a *basis tree* — the tree which shares the most information with other theme sentences. The basis tree is augmented with alternative verbalizations from the other theme sentences, and subtrees of the basis tree

which are not representative of the theme are pruned. Finally, the transformed tree is linearized back into a sentence using a language model.

- **Sentence Ordering** Once the summary sentences have been generated, how can we order them into a coherent text? In the case of multiple-document summarization, some events may not be described in the same article. Furthermore, the ordering of sentences varies from one article to another. The automatic analysis of a newspaper corpus revealed that acceptable orders have to satisfy cohesion and chronological constraints. Our ordering algorithm first identifies groups of cohesively related themes based on word distribution in the input documents. Next, the chronological order among these groups is induced based on time stamps of the input articles.

1.1.3 MultiGen and Newsblaster

Currently MultiGen is a part of Columbia's Newsblaster system. Newsblaster operates over documents that are discovered and downloaded every day from many on-line news sources such as CNN, Reuters, and Washington Post. The system automatically collects, clusters, categorizes, and summarizes news from the web, and it provides a user-friendly interface to browse the results. Newsblaster automatically routes its input to one of two summarizers depending on the type of the input articles: sets of articles about the same event are summarized by MultiGen, and all others are directed to DEMS (Schiffman, Nenkova, and McKeown, 2002). Automatically generated summaries help users to identify stories of interest. If users want to learn more, Newsblaster provides links to the original articles, so a user can read all of the articles

pertaining to a given story. A recent analysis indicates that Newsblaster receives tens of thousands of hits a day and has a large set of followers.

1.2 Contributions

A main contribution of this dissertation is **information fusion**. The primary goal is a method which improves the quality of the produced texts beyond the level of summaries produced by extraction methods, and is also domain-independent and robust.

Because our generation method reuses and alters phrases from the input articles, we avoid the need for an elaborate semantic model of the domain required by traditional generation methods. In the upcoming chapters we show that the main steps of the generation process can be performed based on features extracted from the input documents and knowledge automatically derived from large text corpora. In particular, we proposed new algorithms for content selection, sentence ordering and sentence generation within a summary:

- **Interleaved content selection and sentence generation** The goal of this algorithm is, given a collection of related sentences, to generate a sentence which includes the information common across most input sentences. The research challenges in developing such an algorithm are the identification of the fragments conveying common information and the method for their combination into a sentence. Common information is identified by aligning syntactic trees of input sentences, based on paraphrasing information. A tree encompassing the resulting alignment after a series of transformations is linearized into a sentence

using a language model. According to our evaluation, this method generates a grammatical sentence which accurately synthesizes input phrases in most cases.

- **Sentence ordering** The problem of ordering information in multi-document summarization so that the generated summary is coherent has received relatively little attention. While sentence ordering for single document summarization can be determined from the ordering of sentences in the input article, this is not the case for multi-document summarization where summary sentences may be drawn from different input articles. To understand the properties of acceptable orderings, we collected a corpus of multiple acceptable orderings, and used sequence analysis methods to study these orderings. Based on our findings, we developed an ordering algorithm which utilizes cohesion and chronological information derived from input texts.

The accuracy of information fusion critically depends on its ability to identify paraphrases, since input documents may describe the same information using different words. Therefore, the investigation of paraphrasing mechanisms is another focus of this thesis. This important language phenomenon is largely unaccounted for in the linguistic literature, and was not directly addressed in previous NLP research. We have developed **unsupervised learning techniques to identify paraphrases from a text corpus**. Both the data and the method are departures from past approaches:

- **Corpus for paraphrase acquisition** We proposed using, as a corpus for paraphrase acquisition, a collection of texts which are either parallel or comparable, for example multiple English translations of the same source text written in a

foreign language, or multiple news articles about the same event. These types of texts provide many instances of paraphrasing, because they preserve the meaning of the original source, but may use different words to convey the meaning. Such a corpus not only yields a paraphrase thesaurus needed for information fusion, but it also allows the empirical study of paraphrasing to complement linguistic theories about this phenomena.

- **Algorithm for paraphrase acquisition** We base our method for paraphrasing extraction on the assumption that phrases in aligned sentences which appear in similar contexts are paraphrases. The co-training algorithm learns which contexts are good predictors of paraphrases by analyzing contexts surrounding identical words in aligned sentences. These contexts are used to extract new paraphrases, which in turn are used to learn more contexts. Our algorithm yields phrasal and single word lexical paraphrases as well as some syntactic paraphrases. This method adapts itself to parallel and comparable corpora dynamically. Our evaluation experiments show that the algorithm extracts paraphrases with high accuracy and significantly outperforms a state-of-the-art algorithm developed for related tasks in machine translation.

The multi-document summarization system MultiGen provides a context for the investigation described above and it also serves as a platform for verifying the adequacy of the proposed summary generation strategy in a real world setting.

1.3 Guide to Remaining Chapters

The thesis contains two interconnected parts. The first part (Chapter 2, Chapter 3) is dedicated to paraphrasing, while the second part (Chapter 4, Chapter 5, Chapter 6) focuses on information fusion, which uses paraphrasing as an essential source of knowledge.

Chapter 2 presents background information on the phenomenon of paraphrasing. It reviews the most relevant linguistic theories, and identifies gaps in these theories which hinder the building of a computational model of paraphrases. To bridge some of these gaps, we performed a manual analysis of a corpus rich in paraphrases. The results of this analysis are also discussed in Chapter 2.

An unsupervised method for paraphrase acquisition is described in Chapter 3. A description of the parallel and comparable corpora used for this study is provided in this chapter. We present an extensive evaluation of the algorithm and analysis of extracted paraphrases which sheds light on some questions raised in Chapter 2.

The next two chapters are dedicated to information fusion. Chapter 4 focuses on the sentence fusion method, and covers algorithms for syntactic tree alignment and a method for combining tree fragments into a sentence. Techniques for ordering generated sentences into a coherent text are discussed in Chapter 5. These chapters also include the discussion of related work and justifications for choosing our approach.

In addition to a component-based evaluation reported in Chapter 4 and Chapter 5, we present an overall system evaluation in Chapter 6. This chapter discusses two evaluation efforts: the results from a DARPA sponsored competition of summarization systems and an evaluation of MultiGen in the context of Columbia Newsblaster browsing system.

Finally, Chapter 7 summarizes the thesis work and points out the limitations as well as the future directions of this work.

Chapter 2

Paraphrases

It's not pinin,' it's passed on! This parrot is no more! It has ceased to be! It's expired and gone to meet its maker! This is a late parrot! It's a stiff! Bereft of life, it rests in peace! If you hadn't nailed him to the perch he would be pushing up the daisies! Its metabolical processes are of interest only to historians! It's hopped the twig! It's shuffled off this mortal coil! It's run down the curtain and joined the choir invisible! This.... is an EX-PARROT!

"Dead Parrot Sketch" by Monty Python

Paraphrases are alternative ways to convey the same information. As examples, consider the following different ways to state the fact "The parrot is dead" — "*The parrot has ceased to be*" and "*This is a late parrot*". In the "Dead Parrot Sketch" by Monty Python, this fact is conveyed in more than 15 different ways. A reader would recognize from each of the sentences that the author refers to the same fact about the parrot, despite surface differences in the actual expressions used. Furthermore, all Monty Python's verbalizations of the semantic predicate "to die" do not form an

exhaustive list. Indeed, an obituary writer would be most likely to select different phrases to verbalize this concept.

While speakers of a language deal with paraphrases every day in an effortless, scarcely conscious, fashion, the presence of paraphrases greatly complicates automatic language processing. In a hypothetical language where paraphrasing does not exist, semantic concepts have unique verbalizations — and thus, many complex NLP tasks (e.g., information extraction, question answering, identification of repeated information, etc.) could be reduced to string matching problems. For instance, in this scenario, all the objects of Bill Clinton's affections could be identified by scanning an appropriate news corpus for all Y, such that the phrase "*Clinton loves Y*" appears in the corpus, assuming that semantic relation "love(X,Y)" is verbalized as "X loves Y". In reality, this technique would produce poor results, since it does not account for paraphrases of such sentences.

A full semantic interpretation of an input text would allow one to easily cope with the phenomenon of paraphrasing. However, an intriguing question is whether we can identify paraphrases of a given phrase using linguistic devices without a direct mapping from words to semantics. Certain types of paraphrases appear to be systematic in natural language. For example, so-called active and passive voice sentences in English convey essentially the same meaning. We can identify them by a mechanical comparison of parse trees; this process does not require semantic interpretation of the sentences. Using these shallow linguistic devices is an effective way to grapple with paraphrasing in domains where deep semantic analysis is hard, as is, for example, the domain of topic-independent newspaper data.

The primary goal of this chapter is the study of mechanisms that can produce

paraphrases as a first step towards building a model for their computation described in Chapter 3. While the main source of information in our investigation is the linguistics literature, we supplement it with manual analysis of corpora rich in paraphrase occurrences.

Beyond pragmatic benefits, studying these devices can shed light on the phenomenon of paraphrasing, which is characteristic of all natural languages. Even linguistically naive speakers of a language understand what paraphrases are. Studies from the psycholinguistics literature, presented in Section 2.2, show that speakers of a language can easily recognize and produce paraphrases. However, it is extremely hard to formally define paraphrases in declarative terms. Many linguists (Halliday, 1985; de Beaugrande and Dressler, 1981) agree that paraphrases retain "approximate conceptual equivalence". But the extent of interchangeability between phrases which form paraphrases is an open question (Dras, 1999). This issue is discussed in Section 2.3. In this thesis, we do not attempt to develop a declarative definition of paraphrases. Given multiple pieces of evidence that humans can reliably identify paraphrases, we assume that paraphrasing is a coherent notion, and concentrate on language devices that can produce paraphrases. Research on these devices originating from the linguistic community is presented in Section 2.4. Even though paraphrasing plays a central role in two linguistic theories — Generative Transformational Grammar (Chomsky, 1957) and Meaning-Text Theory (Melcuk, 1988) — there are numerous gaps in the understanding of this phenomena; we underline these questions in Section 2.5. Finally, in Section 2.6, we present our analysis of corpora which contain several verbalizations of the same semantic information in attempt to bridge these gaps and to empirically validate linguistic theories about paraphrasing.

We start this chapter with an informal definition of paraphrases and an introduction of the terminology used throughout this chapter.

2.1 Paraphrases: The Basics

In informal terms, paraphrases are pairs of units with approximate conceptual equivalence that can be substituted for one another in many contexts. Pairs of sentences and phrases in Figure 2.1 fall into this description. Paraphrases do not have to be identical in meaning. The first pair of sentences in Figure 2.1 frequently would be treated as identical in meaning, but one can imagine a context where these sentences would not be substitutable. For example, if Emma took waltz lessons all her life without much progress, the first sentence of this pair still conveys a true proposition, while the second sentence is no longer true.

Paraphrases are frequently classified in terms of the relationship between the members of a paraphrasing pair. The most commonly used classification is lexical versus syntactic paraphrases. The paraphrasing pair D in Figure 2.1 is an example of lexical paraphrase, and the active-passive sentence pair E in Figure 2.1 is an instance of syntactic paraphrase. However, many paraphrases fall in between two categories, as in paraphrasing pair F. As we show in Section 2.4, the distinction between lexical and syntactic paraphrases is even more opaque than it may seem at a first glance; consequently, we do not use this classification scheme within this thesis.

Instead, we distinguish paraphrases in terms of their granularity, a feature which is also tightly related to mechanisms of paraphrase production. Figure 2.1 illustrates paraphrases of different granularity – words(D), phrases(C, G) and sen-

(A) Emma did not know how to waltz. Emma had no clue about waltzing.
(B) The article was warmly discussed , which procured it a high reputation. The paper was hotly debated , causing a fine old uproar.
(C) wooden frame frame made of wood
(D) debate discuss
(E) Eli planted a tomato bush. A tomato bush was planted by Eli.
(F) Louis sold the book to Noemie. Noemie bought the book from Louis.
(G) to aim the guns to get the best firing angles.

Figure 2.1: Examples of paraphrases

tences(A, B, E, F). Some of these pairs (for example D) form what we call *atomic paraphrases* or paraphrases which can not be decomposed any further. Obviously, all word level paraphrases are atomic, but not all atomic paraphrases are word level paraphrases. An example of a phrase level atomic paraphrase is given in (G); these phrases do not contain any subunits which are paraphrases of each other.

Atomic paraphrases can be combined to produce *compositional paraphrases*. For example, sentence (B) contains several atomic paraphrases such as ("*article*", "*paper*"), ("*discussed*", "*debated*") and ("*warmly*", "*hotly*"). Compositional paraphrases can be also derived by applying *compositional rules* which result in different combinations of atomic paraphrases. An example of a compositional rule is the active-passive transformation; an instance of this transformation is shown in Figure 2.1(E). We represent compositional rules as partly lexicalized dependency trees such as the ones in Figure 2.2. In addition to lexicalized nodes, these trees contain slots marked with their part-of-speech tags and reference indexes. Slots marked with the same in-

dex and the same part-of-speech refer to the identical items. For conciseness, we also use the flat representation of compositional rules; for instance the rule from Figure 2.2 is flattened into (NP_1 VB_1 NP_2 ; NP_2 was $VBed_1$ by NP_1).

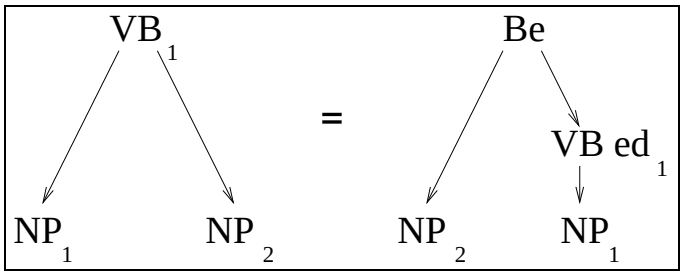

Figure 2.2: Representation for compositional rules.

Similar representations have been used to represent syntactic paraphrases in the past (Chomsky, 1957; Dras, 1999). However, compositional rules encompass a wider range of transformations, some of which are clearly lexical. The pair of sentences in Figure 2.1(F) is an instance of such a paraphrase pair. While the relation between "*buy*" and "*sell*" is lexical, this transformation should be represented by syntactic trees (as the one in Figure 2.3) since it changes the order of the verb arguments, thus modifying the tree structure of the sentences.

In previous research, paraphrases were also classified in terms of their meaning distortion effects. (Dras, 1999) classified syntactic paraphrases into five groups: change of perspective, change of emphasis, change of relation, deletion and clause movement. While this taxonomy makes sense within the specific application considered by Dras[1], the construction of a general classification scheme of this type is hard. The effect of meaning distortion is defined not only by a transformational rule itself,

[1]Dras uses these paraphrases for text simplification.

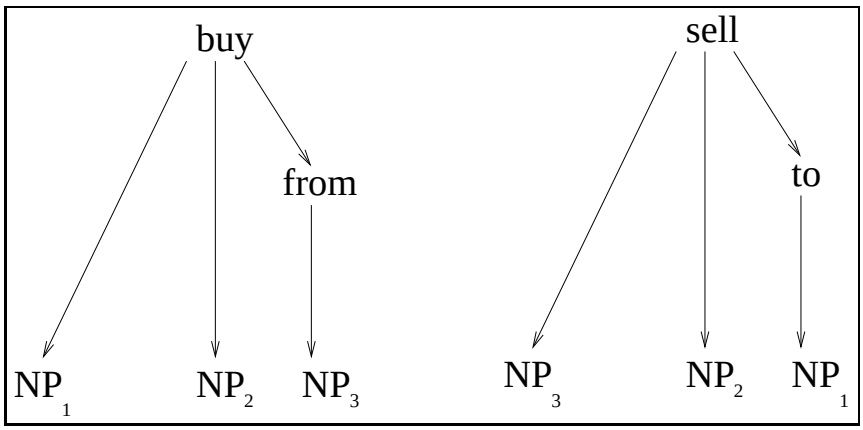

Figure 2.3: Representation for the buy/sell transformation.

but also by the context in which this rule is applied. More importantly, five classes, or any discrete number of categories, are a coarse approximation of the continuum representing meaning changes. Thus, we will not try to classify paraphrases according to their distortion effect. The concern for us is, roughly speaking, a binary distinction — whether or not humans consider two phrases as referring to the same thing, despite some distortion effect in meaning.

In the next section, we present evidence from the literature that humans are capable of identifying and generating paraphrases in consistent fashion.

2.2 The Nature of Paraphrasing Skills

The ability "to say it in one's own words" is intuitively taken as an index of understanding by many in the linguistic community. As a result, in linguistic studies, the task of paraphrasing is used as a standard tool in measuring comprehension. In these studies, the subject is given a phrase, and is asked to perform either the *pro-*

duction task of paraphrasing the given phrase in his/her own words, or to perform the *recognition task* of selecting an appropriate paraphrase from a list of candidates. When the subject fails to provide an appropriate paraphrase for the test sentence, it is a standard assumption in such experiments that s/he does not understand the sentence. One sees that the ability to paraphrase is considered so basic that one would generally rule out the possibility that the subject understands a sentence but can not paraphrase it.

Several psycholinguists (Livant, 1961; Gleitman and Gleitman, 1970) argued that this ruling out may not be justified; that is, there may be cases when a subject does understand a sentence but can not paraphrase it. They looked into the connection of the paraphrasing ability to grammatical knowledge. (Livant, 1961) performed a study of paraphrasing compound nouns: three subjects were given a list of compound nouns and were asked to paraphrase the compounds. Livant reports that his subjects could in all cases paraphrase the compounds. This is all the more surprising because some of the compounds were quite bizarre in interpretation (e.g. *"bird house black"* is paraphrased to *"a blackener of houses who is a bird"*[2]). Based on this evidence, he concludes that all speakers have the same mechanism for paraphrasing; in effect, the words of the compound are treated like variables in a known formula, allowing a speaker to correctly paraphrase semantically peculiar phrases.

(Gleitman and Gleitman, 1970) come to less optimistic conclusions about the mechanical nature of paraphrasing skills after replicating Livant's experiment in a larger scope — 12 participants and 144 compound nouns of various level of complexity and semantic plausibility. The experiment included both generation and recognition

[2]If you managed to paraphrase this one, try "Volume Feeding Management Success Formula Award."

tasks. While the overall performance was satisfactory, none of the twelve subjects performed perfectly. Analysis of errors revealed that some syntactic structures are easier to paraphrase than others, and that certain compounds are associated with certain types of errors. Another interesting question that the study addressed is the correlation between paraphrasing ability and education level. The study included subjects from various educational backgrounds, ranging from high school graduates to holders of Ph.D's. The final results indicate massive differences in paraphrasing ability among different groups of the population. The less-educated groups make more errors, and to a significant extent make different errors than the most educated group. Gleitman and Gleitman's results raise questions about the nature of the paraphrasing process, and do not support the view of paraphrasing as a mechanical process which doesn't require semantic interpretation. However, because many paraphrases in the study were not semantically well-formed, its results do not refute the fact that humans can adequately paraphrase a semantically well-formed phrase.

In addition to studies of paraphrasing phenomena in a research setting, there are multiple pieces of evidence that humans are frequently using their paraphrasing skills in everyday life. Numerous studies link paraphrasing ability with communication skills. A speaker participating in a conversation often paraphrases information stated by other participants to construct the common ground in a dialog (Walker, 1992).[3] In addition to paraphrases, such redundant affirmations can be realized by using other linguistic devices — literal repetitions, entailments and logico-pragmatic inferences. (Walker, 1993b) noted that paraphrases are the most frequently used device.

[3]See (Walker, 1993a) for in depth discussion on the role of informationally redundant statements in the dialog.

As an example of another use of paraphrasing in conversation, consider a grown-up who simplifies his language when conversing with a child. This is an instance of the more general phenomenon of when a speaker wishes to accommodate a dialect/sublanguage of a listener by paraphrasing his message into the listener's sublanguage. Numerous studies in education and linguistics showed that this translation is essential for better communication. In one such study, (Labov, 1972) asked African-American teenagers to repeat verbatim various English sentences. The participants were native to an English dialect different from mainstream English. The children were rarely able to repeat verbatim a sentence in mainstream English, and sometimes repeated a translation of the sentence in their dialect. For example, "*I wonder whether he'll come to my house tonight*" was repeated as "*I wonder will he come to my house tonight*". However, they did not have any problem repeating sentences stated by Labov in their dialect. Thus, "translation" of a message into a listener's language makes it more understandable; speakers employ such paraphrasing techniques to accommodate different types of audiences.

2.3 Lack of a Formal Model

While paraphrases are frequently used by speakers of a language, at the same time, the concept of paraphrasing is remarkably resistant to definition in formal terms. A definition of paraphrasing representative of those in the literature is that of (de Beaugrande and Dressler, 1981) — "approximate conceptual equivalence among outwardly different material". The scope of this "approximate equivalence" is not clearly specified, and is clarified by examples. (Quirk et al., 1985) in their extensive descriptive

grammar of English introduce paraphrases when referring to *correspondences* between phrases such as:

"*He spoke these words*" → "*The speaker of these words*"

"*The girl standing in the corner*" → "*The girl in the corner*"

(Quirk et al., 1985) do not define the scope of these correspondences; they appear to rely on the reader's intuition as to when two constituents mean or refer to the same thing or at least an approximation of the same thing. (Dras, 1999) analyzed definitions of paraphrases in the existing literature and found that typically they are too broad to circumscribe the notion of paraphrase to any great extent.

To adequately model the paraphrasing phenomena one should not consider paraphrases in isolation, but within a particular context. But as (Edmonds and Hirst, 2002) note, "the context-dependent nature of lexical knowledge is not very well understood as yet". In classical models of lexical knowledge (Lyons, 1977; Cruse, 1986) each element of the lexicon is represented as a conceptual schema. Words which have the same meaning are connected to the same schema; if a word is ambiguous, sub-entries for its different senses are connected to their respective schemata. Understanding of the word in this framework is equivalent to finding the schema to which the word corresponds, disambiguating it if necessary. In this model, the connection between a word and its meaning is static. Obviously, it does not provide a mechanism to account for contextual dependency. Thus, two paraphrases which share the same meaning in many contexts but are not fully equivalent in meaning are treated in this model as "widely different words" (Edmonds and Hirst, 2002).

To complicate matters even further, when reasoning about meaning equivalence one should consider the difference between *the sense meaning* and *the reference mean-*

ing in Frege's terms (Frege, 1892). *The reference meaning* of a linguistic unit is the object it represents, while its *sense* is just the concept it expresses as comprehended by the listener. Frege's classical example, "The morning star is the evening star", illustrates the difference. Since "the morning star" and "the evening star" both refer, in fact, to the planet Venus, the statement is equivalent to the tautology that "Venus is Venus." But since this star is observed in the sky at distinct locations on different days, the listener may not be aware of their identity. Thus, his understanding of the sentence would be totally different. Therefore, an adequate model of paraphrasing has to take into account the listener, his expertise and his prior knowledge.

Even limiting ourselves to a more restricted class of paraphrases does not produce a good model. Unable to formally define synonymy, (Quine, 1985) argued that synonymy is as complex as the notion of analyticity and can not be reduced to simpler concepts. In response to Quine's arguments, many prominent philosophers contended that synonymy is part of the linguistic competence of a speaker, and that it does not require further reduction to more primitive notions.

The debate on the formal definition of paraphrases has continued for several decades and is still not closed. We will not go further in the investigation of this issue in the thesis.

2.4 Paraphrasing in Linguistic Theories

Natural languages provide devices to support the paraphrasing ability of speakers. These devices have been investigated by various linguistic schools. Surprisingly, for a long time, lexical paraphrases and syntactic paraphrases were considered in linguis-

tics as two independent phenomena: lexical paraphrases were treated as semantic phenomenon and syntactic paraphrases were considered within a grammar. However, some theories do not fall in any of the two categories, therefore we do not follow the traditional classification of paraphrases; instead we first introduce theories dealing with atomic paraphrases and then present material on compositional paraphrases.

2.4.1 Atomic Paraphrases

When one thinks of lexical paraphrasing, the first thing that comes into mind is the synonymy relation. Webster defines synonymy as "sameness of meaning". This definition suggests that the synonymy relation is a subclass of atomic paraphrases, whose members are completely identical in meaning, in contrast to the "approximate conceptual equivalence" of other atomic paraphrases. This binary distinction makes us think of synonyms as an easily understandable phenomenon. As (Edmonds and Hirst, 2002) point out, "synonymy has often been thought of as a 'non-problem'": either two words are synonyms, which are completely identical in meaning and hence easy to deal with; or, they are not synonyms, in which case each word can be handled like any other.

However, this apparent simplicity of synonymy is deceptive, since absolute synonymy is rare. Absolute synonyms could be substituted one for the other in any context in which their sense is denoted with no change to truth value and communicative effect. Both pragmatic arguments and empirical evidence suggest that absolute synonymy is unlikely to be found in natural language. (Clark, 1992) says that "every two forms contrast in meaning", and argues that for any pair of words with identical meaning either one falls into disuse or they bifurcate in meaning. An excellent illus-

tration of this point is McCawley's example of the usage of the paraphrases "*pink*" and "*pale red*"(McCawley, 1978). While "*pink*" is defined literally as "*pale red*"[4], it seems that "*pale red*" will not be used, since "*pink*" encompasses the same meaning as its syntactically complex "equivalent". However, in some cases "*pale red*"[5] is selected over pink, for example describing a color which is paler than red but not so pale to be pink. McCawley attributes this fact to conversational implicature — "one would go to the effort of saying pale red instead of pink only if there were some reason why pink would be inappropriate." Selecting "*pale red*" allows to a speaker more precisely characterize color only because "*pink*" exists in the language; for example, such a distinction would be impossible for green. In addition to denotational variants, illustrated above, synonyms can differ with respect to their stylistic features ("*inebriated*","*pissed*"), their connotation ("*skinny*", "*slim*") and their structural variations ("*die*", "*pass away*") (Clark, 1992; Edmonds and Hirst, 2002).

The conclusion of these arguments is that no text is interchangeable with any other without a distortion effect. At first glance, this seems to refute the very idea of synonymy and paraphrasing, since now we are claiming that there is "only one way to say the same thing", and any alternative way of saying it will introduce some difference. However, in many contexts this difference is negligible. The paraphrasing phenomena exist in language, because speakers are capable of abstracting over these differences and realizing when two phrases actually refer to the same thing. At the same time, speakers can produce paraphrases which are unambiguously interpreted by the listener.

Acknowledging the existence of differences among synonyms, linguists intro-

[4]WordNet defines "*pink*" as"*a light shade of red*".
[5]Google search on the query "*pale red*" returned 11,300 hits.

duced a notion of *near-synonymy* (Cruse, 1986; Edmonds and Hirst, 2002), which denotes words that are "intuitively very similar in meaning but can not be interchangeable in many contexts without changing some semantic or pragmatic aspect of the language." Given this definition, the distinction between near-synonymy and paraphrasing is blurred. For both, context dependency plays an important role. The context dependency means that beyond two or three synonyms of a word listed in a dictionary, a significantly higher number of other words can act as its paraphrases in a specific context. (Edmonds and Hirst, 2002) use dictionaries of synonyms as a source of near-synonyms. Clearly, these resources do not provide sufficient coverage of near-synonyms, and atomic paraphrases, in general. For example, the words "*debate*" and "*discuss*" are perfectly substitutable in the context of sentences B from Figure 2.1. However, they are not synonyms — "*debate*" is more specific in its meaning then "*discuss*". In fact, in the WordNet hierarchy "*debate*" is a hyponym of a word "*discuss*".

A natural question here is how to exploit existing lexical resources to obtain atomic paraphrases. In other words, what types of lexical relations, beyond synonyms, can yield paraphrases? Evidently, some lexical relations are more likely to yield paraphrases than others. The correspondence between paraphrases and types of lexical relations has not been investigated systematically. In numerous applications WordNet is utilized to acquire atomic paraphrases. In some applications, only synonyms are considered as paraphrases (Langkilde and Knight, 1998); in others, looser definitions are used (Barzilay and Elhadad, 1997). The existing linguistic theories do not give an answer to this question. In Chapter 3, we show how corpus-based techniques can be used to address this question. For now, we continue our survey

with linguistic theories dealing with compositional paraphrases.

2.4.2 Compositional Paraphrases

While atomic paraphrases were viewed as lexico-semantic phenomena, compositional paraphrases were treated as part of a language grammar. This view on compositional paraphrases as syntactic phenomenon originates from the Transformational Grammar days, but later the disconnection of compositional paraphrases from the lexicon in this framework was found to be problematic. The bridge between lexico-semantic and syntactic views on compositional paraphrasing was achieved by Text Meaning Theory which is a lexico-semantic model. (The Text Meaning Theory and Transformational Grammar are discussed later in this section.) Despite surface differences, the two theories exhibit a significant similarity in the primordial role they give to paraphrasing. Both frameworks start with deep-structure (that is, a semantic representation) and use a variety of transformations to produce surface structures with the meaning represented by the original deep-structure. But in sharp contrast to the Transformational Grammar approach, which captures compositional paraphrases in grammar, Text Meaning Theory emphasizes the lexical nature of transformations by representing them in the lexicon. Below, we first briefly outline the basics of each theory and then discuss the treatment of compositional paraphrases within these theories.

Generative-Transformational Grammar

The golden age of syntactic paraphrases was the period of Generative-Transformational Grammar, started by (Harris, 1981a) and (Chomsky, 1957). The concept of transformation, central to this grammar, covers meaning-preserving transformations —

syntactic paraphrases. While pure transformational grammars are out of fashion nowadays, their attempt to represent paraphrases within a grammar gives interesting insights on the paraphrasing phenomenon.

A Transformational Grammar consists of a phrase structure component and a transformational component. The phrase structure produces very basic syntactic trees from deep-structure representations; these syntactic trees have to be further refined by the transformation component. In addition to paraphrasing rules, the transformation component contains rules with a variety of roles, for example, rules that encode morphological operations (number agreement among subject and verb) and rules for generating complex sentences. A transformation rule specifies a structural analysis of the strings to which it applies and the structural change that it induces on these strings. Figure 2.4 shows an example of the passive transformation:

Structural analysis: $NP - Aux - V - NP$ Structural change: $X_1 - X_2 - X_3 - X_4 \rightarrow X_4 - X_2$ + be + en - X_3 - $by + X_1$

Figure 2.4: Examples of sentential paraphrases

By applying different transformation rules to a deep structure, we can produce various surface structures corresponding to the original deep structure. Thus, in terms of the Transformational Grammar, paraphrases are pairs of sentences with identical deep structure and different surface structures.

Transformations appear to be a natural way to represent syntactic paraphrases. However, their addition to the grammar comes with a cost. Beyond significant computational overhead[6], many issues arose from the linguistic side: order of rule applica-

[6]The complexity of the grammar with transformations is equivalent to a Turing machine.

tion, interaction among transformations, ways to constrain the rules, and so on. One such issue relevant to our discussions is lexical dependency of transformation rules. Evidently, many transformation rules are heavily conditioned on lexical items. Even for the most general paraphrasing patterns such as the active-passive paraphrase of transitive verbs, there are some exceptions: such as, some verbs, like "*resemble*", do not typically occur in passive. Thus, applying the transformation from Figure 2.4 to the sentence "*Pablo resembled Apollo*" will result in the odd sentence "*Apollo was resembled by Pablo*". This lexical dependency is even more obvious for other types of paraphrases, such as dative movement and nominalization. This challenged the notion of syntactic paraphrase in general, shifting syntactic paraphrases into a class of lexical phenomena.

Beyond the representation-related issues, work on Transformation Grammar gives us some insight into the typology of syntactic transformations. (Harris, 1981a) listed around twenty different classes of syntactic paraphrases, grouping them according to their granularity (e.g. sentential paraphrases versus paraphrases between noun-phrases and sentences). His class descriptions are often too general to be useful; for example, one of his classes (4.14) contains all of the paraphrases characterized by the change of words in the sentence, such as paraphrases in Figure 2.5, which are very distinct in their nature. Despite its shortcomings, his list includes a variety of paraphrasing types, and raises the questions about how many syntactic paraphrases are there. Harris notes that the list is not exhaustive, and doubts that an exhaustive list will be identified. However, he conjectures that for many applications[7] his list is

[7]Harris proposed to use paraphrases for discourse analysis – using transformations to reduce every sentence to the simplest syntactic form (kernel form) and, then, apply rules for semantic interpretation for kernel sentences.

sufficient.

The public he always despised.→ The public always he despised.
He, an inveterate libertarian, opposed the measure. → An inveterate libertarian, he opposed the measure.

Figure 2.5: Paraphrases generated by the same rules according to Harris

Since the publication of *Co-occurrence and Transformation in Linguistic Structure* there were several attempts by Harris and others (Harris, 1981b; Lees, 1960) to produce a more detailed and systematic account of paraphrases. But even today several questions about syntactic paraphrases remain open. How many syntactic variations are there? Do the types proposed by Harris and others cover the most frequently used ones?

Meaning-Text Theory

The Transformational Grammar focuses on syntactic paraphrases. However, compositional rules can be purely lexical in their nature. An example of an application of such a rule is given in sentence pair F from Figure 2.1. These rules were first studied within Meaning-Text Theory (MTT) developed by Melcuk(Melcuk, 1988) and his colleagues.

As in the Transformational Grammar, the MTT describes the relation of a text to its linguistic meaning by representing an utterance on seven distinct levels ranging from the semantic level, through syntactic levels, to the phonetic level. Meaning is taken to be an invariant of synonymic transformations between utterances originating from the same semantic representation.

An utterance can be mapped from its linguistic meaning to text through each

level of representation by applying paraphrasing rules that replace part of the utterance with a new utterance at the same level, or transformation rules that convert the utterance from one level to another. In sharp contrast with Transformational Grammar, lexical knowledge governs the mapping process at each level, so the mapping rules that are applied depend on the lexical entries of the words involved.

Consequently, the lexicon, called the *Explanatory Combinatorial Dictionary*, plays a central role in this formalism. Among other information, each lexicon entry contains, what are called, the *semantic*, the *syntactic* and the *lexical combinatorial structures*. The *semantic structure* defines a word sense as a semantic network, the nodes of which represent "primitive" word senses in the dictionary. The *syntactic structure* of a word contains the so-called government pattern that describes how the semantic arguments, variables in the semantic network, are realized as syntactic arguments. Finally, the *lexical combinatorial structure* is a dependency structure between word senses that systematically encodes the restricted lexical co-occurrences of every word sense. These co-occurrences are encoded in *lexical functions*.

Each lexical function expresses a certain kind of language-independent relation. For example, the lexical function *Magn(X)* (Melcuk, 1996) maps a word into words "that intensify X":

$Magn(\text{“}shave\text{”}) = \text{“}close\text{”}, \text{“}clean\text{”}$
$Magn(\text{“}condemn\text{”}) = \text{“}strongly\text{”}$

Magn is an example of a lexical function which is semantic in its nature. MTT also includes lexical functions which are based on syntactic relations, such as $Oper_1$. This function maps a noun N to the verb that takes N as its direct object.

$Oper_1(\text{“}cry\text{”}) = \text{“}to\ let\ out\ \text{”}$

$Oper_1(\text{“}strike\text{”}) = \text{“}to\ be\ on\text{”}$

Lexical functions can also be combined to make a *compound* lexical function whose meaning is a “combination” of the meaning of the individual functions. Melcuk claims that 60 lexical functions and their combinations can describe systematically and exhaustively almost the whole range of restricted lexical co-occurrences in any language.

In addition to lexical function, the lexicon includes 60 paraphrasing rules[8] which according to Melcuk are all that are necessary to cover all the systematic paraphrases in any language. They are language independent, just as the 60 lexical functions are. Paraphrases are expressed through lexical functions. For example, the paraphrasing rule shown in Figure 2.6 maps a verb node V to a syntax tree containing a support verb linked to a nominal. By applying this rule to the syntactic representation of the sentence “*Sales decreased sharply in October*”, we get: “*Sales showed a sharp decrease*”.

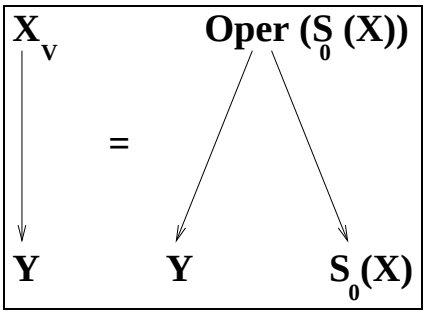

Figure 2.6: Representative transformation from MTT.

The fusion of semantic and syntactic elements within lexical functions poten-

[8] One supposes that 60 is a magic number in MTT.

tially makes the MTT framework conducive to the representation of paraphrasing phenomena. Lexical functions provide a fine taxonomy of semantic relations among items in the lexicon. Beyond commonly used relations such as synonymy and hyperonymy, MTT encodes many other lexico-semantic relations. These relations and their combinations form a more expressive language to represent paraphrasing than in frameworks where only the synonymy relation is marked. Moreover, compositional rules can be readily formulated within MTT since lexical functions are "aware of" how a lexical item manipulates its arguments. For example, consider the *Conversium* relation. This relation holds between words with two arguments which have the same meaning when their arguments are permuted, such as "*precede*" and "*follow*". The paraphrasing rule based on *Conversium* relation, shown in Figure 2.7, allows one to identify automatically that the pairs of sentences in Figure 2.8 are paraphrases.

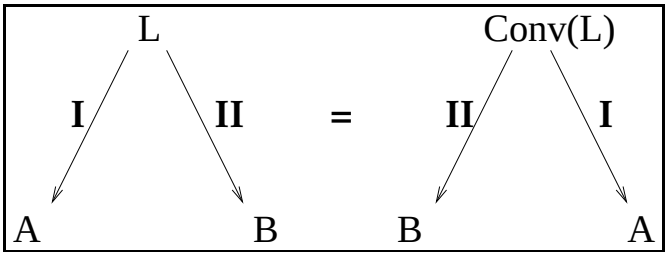

Figure 2.7: Conversion rule.

The keys are **behind** the mug. The mug is **in front of** the keys.
Mary is Peter's **wife**. Peter is Mary's **husband**.
This set set **includes** the element epsilon. The element epsilon **belongs** to this set.

Figure 2.8: Paraphrasing pairs based on *Conversium* relation

Unfortunately, MTT does not tell us the complete story about paraphrasing due to numerous gaps in the theory. Many lexical function definitions are vague and specified only intuitively. Consequently, the values of the lexical functions for every word have not been compiled in any systematic way. In fact, the *Explanatory Combinatorial Dictionary*, the core of the theory, is still not available for English. An *Explanatory Combinatorial Dictionary* of French is incrementally being published (Arbatchewsky-Jumarie et al., 1984) but, at the present time, only several hundreds words have been covered. In addition, an empirical study would be needed to verify whether 60 compositional paraphrasing rules are indeed sufficient to cover paraphrases in any language.

In summary, each of the linguistic theories we surveyed in this section captures certain aspects of the paraphrasing, but does not give a complete account of the phenomena. Furthermore, it is not obvious how to induce a computational model for identification of paraphrases based solely on these theories. In the next section, we underline key issues to be addressed for automatic identification of paraphrases, specifying what information can be taken from the existing linguistic theories and emphasizing what issues required further investigation.

2.5 Towards Automatic Identification of Paraphrases

To automatically identify the paraphrasing relation among sentences, we need to know to what extent they are decomposable into subunits which are paraphrases. This will determine the granularity and the abstraction level of paraphrases that our system

aims to compute. Previous research in linguistics and natural language processing did not directly address this question.

Even assuming the validity of the decomposition assumption, we need some way to compile the set of paraphrasing rules. As the work of both Harris and Melcuk suggest, the number of compositional paraphrasing rules may be limited — Melcuk claims that 60 rules are sufficient, while Harris limits the list to only 20 classes. If they are right, even to the extent that a few compositional paraphrasing rules cover the majority of occurrences, then good results can be achieved with a short manually handcrafted list of compositional paraphrases.

Atomic paraphrases are typically acquired from large-scale lexical resources, such as WordNet. Paraphrases are not identical in meaning; thus, using only synonyms in WordNet as atomic paraphrases may not be sufficient, and we need a way to map WordNet relations to paraphrases. It may also be the case that WordNet is an insufficient resource for acquisition of atomic paraphrases, either because there is no regular mapping between atomic paraphrases and WordNet relations, or because of the insufficient coverage of WordNet.

One way to find answers to these questions is to analyze a corpus of "naturally-occurring" paraphrases: samples of text which convey the same meaning but are produced by different writers. By analyzing the paraphrasing patterns used in such a corpus, we can estimate to what extent the decomposition assumption holds and also identify the language devices used to construct paraphrases and estimate their frequency. The next section presents manual analysis of a corpus. The results of this analysis informed the development of an algorithm for paraphrasing acquisition presented in Chapter 3.

2.6 Empirical Studies of Paraphrasing Phenomena

We first describe our corpus, then our procedure for paraphrase extraction and finally we present our analysis of extracted paraphrases

2.6.1 The Corpus

In our investigation, we used two types of corpora: parallel translations of the same foreign source text[9] and newspaper articles about the same event produced by different writers. Our translation corpus consists of multiple English translations of literary texts written by foreign authors. This corpus provides many instances of paraphrasing, because translations preserve the meaning of the original source, but may use different words to convey the meaning. An example of parallel translations is shown in Figure 2.9. These sentences contain three pairs of paraphrases: ("*burst into tears*", "*cried*"), ("*comfort*", "*console*") and ("*saying things to make her smile*", "*adorning his words with puns*").

Emma burst into tears and he tried to comfort her, saying things to make her smile.
Emma cried, and he tried to console her, adorning his words with puns.

Figure 2.9: Two English translations of the French sentence from Flaubert's "Madame Bovary"

The specific nature of the multiple translations corpus influences the types of paraphrase patterns that we observe. Since both translators start with the same sentence in a foreign language, we know that both translators try to convey exactly the same factual information. However, the form of the original sentence may affect

[9]Foreign sources are not used in our experiment.

the form of the translated sentence. For example, if the original sentence uses active voice, it may bias a translator to keep it, without changing it to a passive voice. This phenomenon constrains to some extent divergences in translations, but still this data is an abundant source of paraphrases – only 2% of sentences from two translations of "Madame Bovary" are translated into identical sentences. This can be explained by the fact that the rendition of a literary text into another language not only includes the translation, but also restructuring of the translation to fit the appropriate literary style. This process introduces differences in the translations which are an intrinsic part of the creative process. Clauses such as *"adorning his words with puns"* and *"saying things to make her smile"* from the sentences in Figure 2.9 are examples of distinct interpretations. The interaction of the two contrasting forces – the attempt to keep the meaning and form of the original sentence, and creativity in translation literary work – determines the types of paraphrases which occur in this corpus.

Another type of corpus we are using is newspaper articles about the same event produced by different writers. These articles frequently overlap in the information they contain. While there is no guarantee that any pair of sentences convey the same semantic information, frequently pairs of sentences containing repeated information can be viewed as paraphrases of each other. An example of such a pair is shown in Figure 2.10. Unlike the translation corpus, in this case writers start with the same facts, and they independently select linguistic forms to verbalize this information. Potentially,[10] this process may produce a wider variety of paraphrases since writers are not constrained by some linguistic realization of this information.

[10]In reality, journalists often familiarize themselves with Reuters and AP news-feeds prior to their writing (Clough et al., 2002).

Two men accused of kidnapping a 7-year-old Philadelphia girl were arrested on Thursday, police said.
Police arrested two men Thursday who are accused of this week's abduction of a 7-year-old Philadelphia girl.

Figure 2.10: Two sentences conveying the same information extracted from the AP and Reuters articles

2.6.2 The Procedure for Paraphrase Extraction

Now, we describe our findings based on the results of manual decomposition of paraphrased sentences. Even though manual analysis significantly limits the size of the corpus we can process, it enables us to accurately investigate the paraphrasing phenomenon and empirically validate findings of linguistic theories. In addition, the results of this analysis justify the automatic methods for paraphrase extraction described in the next chapter.

Prior to paraphrasing extraction, we automatically aligned sentences of the corpus using techniques described in the next chapter. We selected 51 sentences from the corpus of parallel translations and 50 sentences from the corpus of news articles. The only criterion that guided our selection was the semantic similarity between sentence pairs — we manually checked that sentence pairs were indeed paraphrases of each other to eliminate the chance of erroneous alignment.

Then, given a pair of sentences which are paraphrases of each other, we extracted all atomic paraphrases and compositional rules which occur in these sentences. For example, we extracted from sentence pair in Figure 2.10 five paraphrases, among them the atomic paraphrase ("*kidnapping*", "*abduction*") and the compositional rule ("NOUN1 who are VERB1 PP1", "NOUN1 VERB1 PP1") derived from "*men who*

are accused of ..." and "*men accused of ...*".

Paraphrase extraction from parallel sentences is not a straightforward task. The granularity of a decomposition unit can vary greatly, since sometimes paraphrasing relations stretch beyond the word level to larger textual units. This happens when multi-word phrases form atomic paraphrases, for example "*burst into tears*" and "*cried*". But more frequently, multi-word paraphrases are created by compositional rules. The active-passive rule shown in Figure 2.10 "*Two men ... were arrested ...*" and "*... Police arrested two men ...*" manipulates full sentences, operating over several components. Identification of the aforementioned rule requires the knowledge that "*two men accused of kidnapping a 7-year-old Philadelphia girl*" and "*two men who are accused of this week's abduction of a 7-year-old Philadelphia girl*" are paraphrases. These paraphrases, in turn, embrace an atomic paraphrase ("*kidnapping*", "*abduction*") and the compositional rule ("$NOUN_1$ who are $VERB_1$ PP_1", "$NOUN_1$ $VERB_2$ PP_1") derived from "*men who are accused*" and "*men accused*".

To systematically extract paraphrases, we traverse in bottom-up fashion the syntactic trees of parallel sentences, identifying paraphrasing constructions of increasing length and marking them as identical. Given the sentences in Figure 2.10, we first identify that "*kidnapping*" and "*abduction*" are paraphrases of each other, and mark them as identical by substituting the word "*kidnapping*" with the word "*abduction*" in the first tree. Then, we align together the subtree corresponding to "*this week's abduction*" with the node "*abduction*". From this alignment, we induce the following compositional rule "this week's $NOUN_1$" $\rightarrow$ "$NOUN_1$". The second rule that we induce in this process is "$VERB_1$ on $NOUN_1$" $\rightarrow$ "$VERB_1$ $NOUN_1$" which follows from "*arrest on Thursday*" and "*arrest Thursday*". Next, we extract the rule

"$NOUN_1$ who are $VERB_1$ PP_1" $\rightarrow$ "$NOUN_1$ $VERB_2$ PP_1". The last rule extracted in this process is the active-passive transformation. Following this procedure, we extracted from these sentences one atomic paraphrase and four compositional paraphrases.

2.6.3 The Analysis of Extracted Paraphrases

The central question in our investigation is the typology of the extracted paraphrases and their granularity. The length of paraphrases directly reflects to which degree the decomposition hypothesis holds - shorter extracted paraphrases provide more support to this hypothesis, since they correspond to decomposition of sentences into smaller matching units. We present our analysis for two types of corpora below.[11]

Parallel Translations From 51 pairs of sentences of parallel translations, we have extracted 266 paraphrase pairs – 178 atomic paraphrases and 88 compositional rules.

As shown in Figure 2.11, the majority of extracted atomic paraphrases (60.1%) are one-to-one paraphrases (notated as [1;1] in the table), furthermore, paraphrases in which one of the pair members is a word (represented by combination of [1;1] (60.1%) and [1;2] (23.6%) categories in the table) account for 83.7% of atomic pairs. The remaining multi-word paraphrases account for only 17%, however they appear in 12 (23.5%) sentences in our corpus.

These statistics motivated us to take a closer look at multi-word paraphrases. A significant ratio of the paraphrases in [1;1] and [2;2] categories can be attributed to collocations. Verb-particle constructions such as "*stand up*", are the dominant type

[11] Our corpus along with extracted paraphrases is available at `http://www.cs.columbia.edu/~regina/manual`.

Granularity	Frequency	Example
[1;1]	107(60.1%)	(“*mute*”, “*damb*”), (“*teacher*”, “*master*”)
[1;2]	42(23.6%)	(“*realize*”, “*find out*”), (“*enormous*”, “*inordinately large*”)
[2;2]	16(9%)	(“*lancers's cap*”, “*billycock hat*”), (“*had to*”, “*was obliged*”)
Others	13(7.3%)	(“*we'll move him up*”, “*he will go into*”)

Figure 2.11: Length distribution of 178 atomic paraphrases extracted from the corpus of parallel translations

of collocations in our data: 12 out of 41 one-to-two ([1;2]]) paraphrases, and five out of 16 [2;2] paraphrases contain such verbs. Multi-word paraphrases containing idiomatic expressions do not contradict the decomposition hypothesis, since a collocation can be viewed as a single lexical unit, because its meaning “cannot be inferred from the meaning of its part” (Cruse, 1986). Another class of multi-word paraphrases consists of what we call “definition-paraphrases”, e.g. (“*mumble*”, “*articulate in a stammering voice*”) and (“*bangs*”, “*square on his forehead*”). These pairs consist of a word and a multi-word phrase which defines the word (similar to a word gloss in WordNet). Paraphrases which consist of phrases longer than two words contain neither collocations nor “definition paraphrases”. An obvious characteristic of these paraphrase, is that they exhibit divergence in meaning. A representative example from this category is the pair “*The door half hid him from the view*” and “*He could hardly be seen*”. Strictly speaking, these sentences do not convey the same information, and generally can not be substitutable. However, in a context where we care only about the fact “*he is invisible*”, given that one can infer it from either sentence, the two sentences provide the same information. Obviously, in this case the decomposition hypothesis

does not hold.

We analyzed compositional extracted paraphrases in terms of transformations they represent, and divided them into four groups – deletions, permutations, noun phrase transformations and lexical paraphrases. As shown in Figure 2.12, the most common transformation among compositional rules is deletion, which includes, mainly, elimination of verb arguments (propositions and adverbials), elimination of noun arguments (adjectives, determiners) and deletion caused by ellipsis constructions. The permutation category encompasses all the cases where the only change to a syntactic tree was a change of order among children of some node in the tree. All such permutations were caused by change in locations of a prepositional phrase within other verb arguments. Noun phrase transformations include changes in realization of dependencies of head nouns, such as rewriting of adjective arguments as prepositional phrases attached to the same head noun.

Type	Frequency	Examples
Deletions	41(46.6%)	[$VERB_1$ PP_1; $VERB_1$]
Permutations	4(10.1%)	[$VERB_1$ PP_1 PP_2; $VERB_1$ PP_2 PP_1]
Noun Phrase Trans.	7(19.3%)	[$NOUN_1$, in $NOUN_2$; $ADJ(NOUN_2)$ $NOUN_1$]
Lexical	35(39.8%)	[$NOUN_1$ had a ADJ_1 look; $NOUN_1$ looked ADJ_1]

Figure 2.12: Type distribution of 88 compositional paraphrases extracted from the corpus of parallel translations

These three categories of compositional paraphrases — deletions, permutations and noun phrase transformations — are usually considered to be syntactic paraphrases (Dras, 1999; Harris, 1981b), since they modify the structure of the parse tree without adding new open class words (at least, in one direction). One may still argue that these paraphrases are not purely syntactic, since they are conditioned on lexical

types of their slots. However, the last category of compositional paraphrases can not be classified as syntactic even using the looser definition of this term. Each of these rules contains a syntactic tree with several open class words which map into another syntactic tree with different open class words. These transformations are heavily dependent on lexical items within them. Consider a representative transformation of this group — "soon $VERB_1$; was not long in $VERB_1$-ing" as in ("*He soon left.*" "*He was not long in leaving*"). This transformation can be viewed as a generalization of atomic paraphrases which cause changes in tree structure when substituted one for another.

Not surprisingly, lexical compositional rules are similar to atomic rules in other respects. Short lexical compositional rules (containing no more than two lexical items in each tree) are usually meaning preserving, for example "like $NOUN_1$ and "in style of $NOUN_1$" (as in "*like Auster*" and "*in Auster style*"). Longer lexical compositional rules produce sentences that do not convey exactly the same information but rather intersect in some limited aspect of their meaning. A pair ("give $NOUN_1$ education"; "send $NOUN_1$ to a public school") is typical of this category. Most readers will agree that the second clause implies the first, but clearly the inference does not hold in the other direction.

News Corpus From 50 pairs of news sentences, 155 paraphrases were extracted — 84 atomic paraphrases and 71 compositional rules.[12]

The set of atomic paraphrases extracted from the News Corpus exhibits many

[12]In our previous work (Barzilay, McKeown, and Elhadad, 1999), we studied paraphrases extracted from 200 sentences derived from the Topic Detection and Tracking corpus. The articles in our sample were produced by the same source (Reuters), which may limit the variety of observed paraphrases. For this reason, we didn't include this data in the thesis.

Granularity	Frequency	Examples
[1;1]	46(54.8%)	("*congregation*", "*church*"), ("*primary*", "*battle*")
[1;2]	13(15%)	("*police*", "*security official*"), ("*before now*", "*previous*")
[2;2]	9(10.7%)	("*be confusing*", "*cause problems*")
Others	16(19%)	("*will not become noticeably ill*", "*suffer nothing more than headaches*")

Figure 2.13: Length distribution of 84 atomic paraphrases extracted from the news corpus

similarities to atomic paraphrases extracted from parallel translations. In particular, the length distribution of the atomic paraphrases shown in Figure 2.13 resembles the distribution of atomic paraphrases in the corpus of parallel translations (See Figure 2.11): the majority of atomic paraphrases are word-level paraphrases (notated as [1;1] in the table) — 54.8%, and paraphrases in which one of the pair members is a word (represented by combination of [1;1] (54.8%) and [1;2] (15%) categories in the table), comprises 69.8% of extracted paraphrases. This corpus contains a larger fraction of long paraphrases (the remaining two rows of the table) — 25.7%. These long paraphrases appear in 13 (26.5%) sentences. As in the translation corpus, paraphrases in [1;1] and [1;2] categories are mostly collocations and "definition" paraphrases; long non-decomposable pairs do not convey exactly the same information, and their similarity in meaning holds only in very particular context or requires world knowledge.

While the two corpora are essentially similar in terms of atomic paraphrases, there are some differences in the types of compositional paraphrasing rules (see Figure 2.14). This corpus contains a new category of transformation rules – active-

Type	Frequency	Examples
Deletions	18(25.4%)	[$NOUN_1$ $NOUN_2$; $NOUN_1$]
Permutations	8(11.3%)	[$VERB_1$ PP_1 PP_2; $VERB_1$ PP_2 PP_1]
Noun Phrase Transformations	15(21.1%)	[N1 N2; N1]
Active-passive Transformation	5(7%)	
Lexical	25(35.2%)	[sicken NP_1; NP_1 were infected]

Figure 2.14: Type distribution of 71 compositional paraphrases extracted from the news corpus

passive transformations, which do not appear in parallel translations. This confirms our hypothesis that the type of corpus influences types of differences between sentences paraphrasing the same information. In other aspects, the compositional rules extracted from two corpora do not differ significantly: they have similar length distribution, and longer lexical paraphrases exhibit larger divergence in meaning than shorter ones.

In the next section, we discuss how the findings of our analysis guide the development of an algorithm for paraphrase acquisition.

2.6.4 Discussion

We found that the majority of sentence level paraphrases break down into one word paraphrases or short phrases. This is good news from a computational perspective, whether in a statistical or a symbolic framework. The effective use of statistical techniques is contingent on reliable counts collected from a corpus, and, obviously, words

and short phrases have a higher chance of appearing in the corpus than long phrases. In a symbolic framework, we can get a better coverage of shorter paraphrases than the longer ones using existing lexical resources, since such resources rarely include long phrases.

Our corpus gives an interesting insight on compositional rules. While extracted compositional rules exhibited high variability, only very few rule types account for the majority of cases. In other words, compositional rules follow a Zipfian distribution: there is a small number of very frequently used rules, and a large number of rules that occur once in the corpus. This result correlates with the hypothesis of Harris and Melcuk about a limited number of compositional rules. From a practical point of view, it implies that by encoding these few frequently used compositional rules, we can effectively deal with compositional rules, and the acquisition of all the rules is not as crucial as that of atomic paraphrases. So far, we have not discussed the issue of the identification of long non-decomposable paraphrases. They occur in up to 25% of the sentences in our corpus; therefore, one cannot simply neglect them. As our analysis shows, their similarity in meaning is context dependent and is based on inference and world knowledge.In limited domains it is worth it to explore techniques for extracting such paraphrases, because the same context may frequently re-occur in that domain (see (Barzilay and Lee, 2003)). Since we aim to apply collected paraphrases in a domain-independent summarization system, long non-decomposable paraphrases are beyond the scope of our investigation.

The results of our analysis have several implications for the information fusion algorithm. The decomposition assumption justifies sentence pair comparison by matching their fragments, rather than considering each sentence as a non-decompos-

able unit. Furthermore, matching of sentence fragments can be typically expressed in terms of syntactic rules, suggesting that comparison on the level of syntactic structures may benefit alignment process. Our analysis also revealed that a limited number of such rules covers the majority of cases, thus a comparison algorithm relying only on these rules would achieve reasonable coverage. Therefore, the accuracy of sentence comparison depends on how well we can identify phrase-level paraphrases. In the next chapter, we focus on a method for learning such paraphrases from a corpus.

Chapter 3

Computational Models of Paraphrasing

This chapter[1] introduces an automatic method for paraphrase extraction from a corpus of texts conveying the same information. The extracted paraphrases allow empirical studies of paraphrasing phenomena to complement the research of these phenomena in linguistics described in the previous chapter.

3.1 Introduction

A method for the automatic acquisition of paraphrases would have both practical and linguistic consequences. From a practical point of view, diversity in expression presents a major challenge for many NLP applications. In multi-document summarization, identification of paraphrasing is required to find repetitive information in the

[1] The algorithm described in this chapter was originally presented in an ACL paper (Barzilay and McKeown, 2001).

input documents. In generation, paraphrasing is employed to create more varied and fluent text. Most current applications use manually collected paraphrases tailored to a specific application, or utilize existing lexical resources such as WordNet (Miller et al., 1990) to identify paraphrases. However, the process of manually collecting paraphrases is time consuming. Moreover, the resulting collection is not reusable in other applications. Existing resources include mainly word-level paraphrases; they do not include phrasal or compositional paraphrases.

From a linguistic point of view, there are questions concerning which language devices are used to construct paraphrases. Even though linguists who studied paraphrasing phenomena manually compiled lists of paraphrasing rules (see Chapter 2), no empirical evidence has been provided to support the claim that such lists are exhaustive. In fact, there is no evidence that such lists cover the most frequently used paraphrases. One way to find answers to these questions is to analyze instances of paraphrase usage.

To get a sizable sample of paraphrases, we need a corpus rich in paraphrases and an automatic method for acquiring paraphrases from this corpus. Hopefully, such a corpus would consist of texts conveying the same meaning, but using different words to express this meaning. In the previous chapter (see page 39), we proposed two types of corpora which fit this profile: alternative translations of the same foreign source text, and newspaper articles about the same event produced by different writers. A manual analysis of a small portion of these corpora confirmed our hypothesis that this corpus contains many instances of paraphrases. We also observed that pairs of sentences conveying the same information are usually decomposable into word and phrase level units which are paraphrases of each other.

In this chapter, we present an unsupervised method for paraphrase extraction from a parallel corpus. Given a pair of sentences with similar meaning, our technique identifies atomic paraphrases and compositional rules contained in this pair. For example, the pair of sentences shown in Figure 3.1 yields three pairs of atomic paraphrases: ("*reproach*", "*accuse*"), ("*wounding*", "*venomous*"), and one compositional rule: [VB1 VB2ing; VB1 to VB2] (corresponding to "*began accusing*" and "*began to accuse*").

Suddenly her face changed entirely and instead of sadness it expressed irritation, and with the most venomous words she began accusing me of selfishness and cruelty.
She dried her tears, and began to reproach me, in wounding terms, for my selfishness and cruelty.

Figure 3.1: Two English translations of the Russian sentence from Tolstoy's "The Kreutzer Sonata"

Our method for paraphrase extraction builds upon the methodology developed in Machine Translation (MT). In MT, pairs of sentences from a bilingual corpus are aligned, and occurrence patterns of words in two languages in the text are extracted and matched using correlation measures. These techniques usually rely on parallel texts, produced by consistent literal translations, such as translations of parliamentary proceedings (Canadian Hansards).

Resources of this type are not readily available for paraphrase acquisition. In fact, our corpus is far from being as clean as the parallel corpora used in MT. It is unrealistic to expect that two articles about the same event from several newspapers would present exactly the same information. They may differ in attitude, angle and facts. Even the more homogeneous part of our corpus — multiple translations

of literary texts — exhibit divergence in meaning. Not surprisingly, this type of resource is not generally used as a parallel corpus in MT. This suggests that adopting the techniques used in traditional MT without modification is not the most effective way to tackle the task of paraphrasing extraction.

We developed an unsupervised learning algorithm for paraphrase extraction, designed to operate over texts conveying the same information produced by different writers. We now briefly describe the algorithm. During a preprocessing stage, corresponding sentences are aligned. We based our method for paraphrase extraction on the assumption that phrases in aligned sentences which appear in similar contexts are paraphrases. To automatically infer which contexts are good predictors of paraphrases, contexts surrounding identical words in aligned sentences are extracted and filtered according to their predictive power. Then, these contexts are used to extract new paraphrases. In addition to learning lexical paraphrases, the method also learns syntactic paraphrases, by generalizing syntactic patterns of the extracted paraphrases. Extracted paraphrases are then applied to the corpus, and used to learn new context rules. This iterative algorithm continues until no new paraphrases are discovered.

Our algorithm has several novel features:

Identification of phrasal paraphrases.

In contrast to earlier work, our approach allows for identification of multi-word paraphrases, in addition to single word paraphrases.

Extraction of compositional paraphrasing rules.

Our approach yields a set of paraphrasing patterns by extrapolating the syntac-

tic and morphological structure of extracted paraphrases. This process relies on morphological information and part-of-speech tagging. Many of the rules identified by the algorithm match those that have been described as productive paraphrases in the linguistic literature.

Adaptation to the similarity level of input translations.

Our method can handle translations along a continuum of similarity, ranging from parallel translations to comparable corpora.

We use our technique to extract paraphrases from our translations and news corpora, with good results on both parts of the corpora in terms of accuracy of extracted paraphrases and coverage. We also found that our method significantly outperforms state of the art MT techniques applied to paraphrasing extraction task. The high quality of produced output allows us to reliably analyze the paraphrasing mechanisms exhibited in text, complementing our manual analysis of the small portion of the corpus.

The remainder of the chapter is organized as follows: we first provide an overview of existing work on paraphrasing extraction methods. Section 3.3 describes data used in our work, emphasizing its differences from typical parallel corpus used in machine translation. In section 3.4, we detail our paraphrase extraction technique. The description of the corpus, a methodology for parameter estimation and results of our evaluation are presented in section 3.5. We conclude with the analysis of extracted paraphrases in section 3.5.4.

3.2 Related Work on Paraphrasing

Many NLP applications are required to deal with the unlimited variety of ways to express the same information. So far, three major approaches of collecting paraphrases have emerged: manual collection, utilization of existing lexical resources and corpus-based extraction of paraphrases.

3.2.1 Manual Collection of Paraphrases

Manual collection of paraphrases is usually used in generation (Iordanskaja, Kittredge, and Polguere, 1991; Robin, 1994; McKeown, Kukich, and Shaw, 1994). Paraphrasing is an inevitable part of any generation task, because a semantic concept can be realized in many different ways. Knowing multiple ways to verbalize a concept can help to generate a text which best fits existing syntactic and pragmatic constraints. Traditionally, alternative verbalizations are derived from a manual corpus analysis, and are application-specific. The generation literature has concentrated on ways to deal with paraphrases within a lexical chooser, while quantitative descriptions of the paraphrasing capacities and a methodology for collecting paraphrases usually have not been addressed.

3.2.2 Deriving Paraphrases through Existing Lexical Resources

The utilization of existing lexical resources, such as WordNet, to derive paraphrases overcomes the scalability problem associated with the previous approach. Lexical resources are used in statistical generation, summarization and question-answering.

A pertinent question here is what type of WordNet relations can be considered as paraphrases. In some applications, only synonyms are considered to be paraphrases (Langkilde and Knight, 1998); in others, looser definitions are used (Barzilay and Elhadad, 1997). These definitions are valid in the context of particular applications; however, it is not clear which WordNet relations yield paraphrases in general.

Automatically constructed thesauri seem to be an appealing alternative to manually-derived lexical resources. Methods for thesauri construction are based on the Distributional Hypothesis, which states that words that occurred in the same contexts tend to have similar meaning (Hindle, 1990; Pereira, Tishby, and Lee, 1993; Hatzivassiloglou and McKeown, 1993; Lin, 1998). For instance, two nouns are likely to have similar meaning if they tend to occur as the direct objects of the same verbs. Originally, similarity-based approaches were developed for dealing with the data sparseness problem by estimating the likelihood of unseen events from that of similar events that have been seen (Hindle, 1990; Pereira, Tishby, and Lee, 1993). Recently, these methods were also applied to thesauri construction, and even directly compared to the human-crafted lexical resources WordNet and Roget (Lin, 1998). While the similarity based approaches have been shown to improve language modeling, they do not seem be as promising for paraphrase extraction. Often the extracted words are similar, but they are not paraphrases. For example, while "dog" and "cat" are recognized as the most similar concepts by the method described in (Lin, 1998), it is hard to imagine a context in which these words would be interchangeable.

Another obvious limitation of using existing lexical resources for paraphrase acquisition is that they do not contain multi-word paraphrases; this has been the case for manually-crafted thesauri as well as for automatically constructed resources.

Such thesauri also do not contain compositional paraphrases.

3.2.3 Corpus-based Extraction of Paraphrases

As in many other tasks, corpus-based techniques for paraphrase acquisition have the potential to yield the type of paraphrases required by front-end applications, without the scalability bottleneck associated with hand-crafted methods. Not surprisingly, all existing corpus-based approaches aim to extract compositional paraphrases which previously were exclusively collected manually.

The first method described in this section semi-automatically extracts paraphrasing rules for technical terms and uses a corpus to tune them. The corpus in this approach helps to create an initial set of rules, which are then further refined by comparing the terms they generate with terms found in the corpus. Considering the manual effort involved, this technique seems applicable to situations where it is known *a priori* that the number of paraphrases is limited.

The other methods described in this section use a corpus to completely automatically learn paraphrases. Usually, such a corpus repeats the same information many times, using different words. These methods collect paraphrases by grouping together these repeated occurrences, and abstracting from them compositional rules. The common feature of these approaches is that they aim to extract paraphrases of a relation or relations, in contrast to our method.

Semi-automatic Methods

The first attempt to derive a large-scale collection of paraphrases was undertaken by (Jacquemin, Klavans, and Tzoukermann, 1997; Jacquemin, 1999), who investigated

variants of technical terms. The main motivation behind these approaches was to improve information retrieval by conflating all the variants of the term into a normalized form for index construction.

In their corpus, the majority of terms are multi-word phrases; thus their paraphrases are derived through compositional rules. Compositional rules are represented by pairs of part of speech tags sequences, with identical words co-indexed. The rules also indicate morphological and semantic dependencies between words, which are computed using lexical resources, such as CELEX (Baayen, Piepenbrock, and van Rijn, 1993), WordNet (Miller et al., 1990) and the thesaurus of the word processor Microsoft Word97. For example, one of the extracted rules transforms an Adjective-Noun term $A_1\ N_2$ into

$$N_1((\text{CC Det}^?)^?\text{Prep Det}^?(A|N|part)^{0-3})N_2'$$

N_1 is a noun in the morphological family of A_1 and N_2 is semantically related with N_2'. This variation recognizes "*malignant tumor*" and "*malignancy in orbital tumors*" (N_2 $\rightarrow$ "*tumor*", $A_1 \rightarrow$ "*malignant*", $N_1 \rightarrow$ "*malignancy*", $N_2' \rightarrow$ "*tumors*", $Prep \rightarrow$ "*in*", $A \rightarrow$ "*orbital*"). The rules were semi-automatically compiled and manually tuned using a large collection of terms extracted from the corpus. This process yielded 62 compositional rules; 91% of them were judged to be accurate, by a manual inspection of the variants they produced. An interesting finding of this work is that indexing which incorporates paraphrasing information significantly increases the effectiveness of information retrieval (Jacquemin, Klavans, and Tzoukermann, 1997). While the methodology used for rule construction yielded good results for the identification of domain-specific term paraphrases, the techniques used have not yet been extended to other types of paraphrasing. The obvious barrier to such an extension would be the

manual effort involved in building these rules.

Bootstrapping Approaches

The majority of fully automatic approaches for paraphrase extraction attempt to learn an extensive list of paraphrases for a specific semantic relation (Duclaye, Yvon, and Collin, 2002; Ravichandran and Hovy, 2002). These methods were developed for question-answering, where the goal is to reformulate a question to find an answer in the document if it is stated in the paraphrased form. For example, an answer to the question "*Who is the author of 'The New York Trilogy'?*" may be stated in the text as "*Paul Auster wrote 'The New York Trilogy'*". Thus, successful extraction of this answer requires the knowledge that "X is the author of Y" is equivalent to "X wrote Y". This task has a lot in common with traditional information extraction, where the goal is to fill slots in a template representing a particular relation. Not surprisingly, the unsupervised (or weakly supervised) techniques developed for both tasks are very similar, originating from (Brin, 1998; Riloff and Jones, 1999) and further developed in (Agichtein and Gravano, 2000). These methods work best in an environment like the World-Wide Web, which contains numerous verbalizations of the same fact. For instance, the Google query consisting of the two terms "*Paul Auster*" and "*The New York Trilogy*" returns 1510 pages; one would expect that the majority of them verbalize the *authorship* relation between the two. This redundancy is the main knowledge source for these methods.

While these methods vary in the details, they can be viewed as instances of the same high-level algorithm:

1. Start with a seed set of tuples known to satisfy the relation of interest (e.g.

("*Paul Auster*", "*The New York Trilogy*") for the the *authorship* relation).

2. Retrieve sentences from the corpus containing tuple members and also satisfying some other constraints (for example, a predefined distance between tuple members); extract from each sentence a plausible verbalization of the relation in the form of a template by abstracting over tuple values (e.g. "X wrote Y", "X is the author of Y").

3. Filter the extracted templates (for example, by their frequency) and apply them to the collection; extract more tuples (e.g. ("*Gustave Flaubert*", "*Madame Bovary*")).

4. Filter the extracted tuples and return to the second step or finish.

Extensive evaluations of these methods in information extraction[2] (Brin, 1998; Riloff and Jones, 1999; Agichtein and Gravano, 2000) showed that these approaches produce accurate results for several relations, such as *organization-location* and *authorship*. However, the majority of these systems were tested on what we call *dominant* relations — a tuple satisfying a dominant relation typically does not satisfy any other relation. The *authorship* relation is an example of a dominant relation. When the name of the writer "Paul Auster" appears as the subject of a sentence and the novel "The New York Trilogy" as its object, most likely a verb of this sentence verbalizes the relation of authorship. This is not the case for many relations — for example, it is hard to find two concepts which are primarily related by the relation *visit*. Therefore, it is not clear how these methods will perform for non-dominant verbs. The

[2]Reported evaluations of these methods in Question-Answering are more limited in scope

techniques for paraphrasing extraction were tested on few relations, mainly on dominant ones. For example, (Ravichandran and Hovy, 2002) learned paraphrases for five relations, such as *birthdate*, *location*, *inventor*, *discoverer*, *definition* and *why-famous*. Not surprisingly, their evaluation showed that performance on *why-famous* was significantly lower than on the rest of relations. More extensive analysis of scalability of these techniques has yet to come.

Paraphrase Acquisition from Related Data

The approach which is the closest to our work is a method for paraphrase acquisition from related news articles (Shinyama et al., 2002). The goal of this work is to identify paraphrases of patterns used for the information extraction task. For example, given the pattern "*X killed Y*" the system aims to induce the paraphrases "*X murdered Y*" and "*Y was killed by X*". The paraphrasing ability would allow an information extraction system to achieve high accuracy when it is provided with one extraction pattern and the rest of them are derived through paraphrasing. Since information extraction is a domain dependent application, the authors focus on domain specific patterns.

This method operates over a corpus similar to our news corpus — collection of articles about the same event published in the same day, with the only difference being that their articles are selected in a domain of choice. However, the pairwise comparison is performed only over sentences containing information extraction patterns; such sentences are recognized automatically using a method described in (Sudo and Sekine, 2001). Then, sentence similarity is computed through named entity over-

lap,[3] weighted by their $tf * idf$ scores. Sentences with similarity exceeding a certain threshold are considered paraphrases. This method yields paraphrases such as "$PERSON_1$ *admits [something]*", "$PERSON_1$ *testifies [something]*". Even though limited in scope, the evaluation based on the manual inspection of produced pairs revealed quite surprising results — the system achieved 94% accuracy on a personnel affairs domain, and 49% on arrest events. This impressive performance is achieved with a simple similarity function. Moreover, it only looks at named entities, which represent a small fraction of sentence words. We argue that the performance highly depends on the relation to be extracted. Even relations which are not dominant in general, would be dominant in this case. For example, a person may have numerous relations with an organization over the time, e.g. the person my be hired to the organization, be its spoke-person, he may sue it, etc.; but most likely an article describing an event in a personnel affairs domain describes only one of this relations, and this relation will be dominant between the person and the organization in the input articles. In a context of a specific event, two named entities usually stand in only one relation. So, if two clauses contain the same named entities, it is safe to infer that they verbalize the same relation. This can explain the gap in performance in the personnel affairs domain and the arrest events domain. As (Shinyama et al., 2002) noted, the average number of named entities in the latter domain is low, significantly decreasing the performance.

This approach, as well as bootstrapping approaches, exploits redundancy in the data to extract verbalizations. While the bootstrapping approaches achieve their accuracy through more complicated machinery, (Shinyama et al., 2002) draws its

[3](Shinyama et al., 2002) defined a named entity to be a name of an organization, person, location, date, or a numerical expression.

strength from carefully collected data, known to describe the same event.

All the approaches described in this section aim to extract paraphrases of a given relation. In contrast, our method is not limited to a particular paraphrase type, but rather we are aiming to extract a variety of paraphrases from a corpus containing repeated information. In the next section, we describe in detail the corpus used by our algorithm.

3.3 The Data

Adam was the only man who, when he said a good thing, knew that nobody had said it before him. Mark Twain

The success of our approach is contingent on the availability of a corpus abundant in instances of "naturally-occurring" paraphrases. Fortunately, such resources are plentiful – given the redundancy of information on the web, we can easily collect texts which contain basically the same information described in different sources. Beyond parallel English translations and newspaper articles about the same event used in this thesis, other instances of such corpora include definitions of the same concept from different encyclopedias, biographies of the same person composed by different writers and different descriptions of a disease from the medical literature. In this section, we underline the properties of this type of corpus which informed the development of the algorithm for paraphrase acquisition. (In section 3.5.1, we detail the size of the corpus we used in our experiments and methods for its collection.)

At first glance, our corpus seems quite similar to the parallel corpora used by researchers in MT; in both cases, basically the same content is available in several

languages. The major distinction lies in the degree of similarity between parallel parts of the corpus. It is true that even the most careful translation from one language to another may introduce some divergences in meaning. However, researchers in MT usually select parallel texts where the chance of such divergences is minimized, as in translations of parliamentary proceedings (Canadian Hansards) or other official documents of countries with multiple official languages. (Manning and Schutze, 2000) suggest that the nature of these texts has been helpful to MT researchers, since "the demands of accuracy lead the translators of this sort of material to use very consistent, literal translations". The dependence of existing MT systems on the accuracy of the translations seems to be so strong, that it rules out use of more "free" style translations as parallel text: "...these sources are easily available (religious and literary works are often freely available in many languages), ...but they tend to involve much less literal translation, and hence results are harder to come by". (Manning and Schutze, 2000)

We cannot control proximity in our corpus to the same extent, since often paraphrases of the same data are produced by writers independently of each other. Given the same semantic input, different authors compose text with the same core meaning, but they may (and do) delete or insert information. Obviously, we can not expect that two stories about the same event from different newspapers would present exactly the same information.[4] Even in the case of multiple translations of the same literary text, where two translators start with the same textual input (in a foreign language), the discrepancy between parallel parts is unavoidable. Analyzing multiple translations of the literary texts, critics (e.g. (Wechsler, 1998)) have observed that

[4] Unless the reporters commit themselves to keeping the language of the source Reuters Newsfeed, which sometimes happens.

translations "are never identical", and each translator creates his own interpretation of the text. Therefore, one cannot neglect the presence of such noise in our input corpora, and it is crucial to keep this in mind while designing an algorithm for paraphrase extraction.

Another distinction between our corpus and parallel MT corpora is that many words are identical in both parts of the corpus. In MT, no words (except cognates) in the source language are retained in the target language translation; for example, an English translation of a French source does not contain untranslated French fragments. In contrast, in our corpus the same word is *often* used in both translations, and only sometimes are its paraphrases used; this means that word–paraphrase pairs will have lower co-occurrence rates than word–translation pairs in MT. For example, consider occurrences of the word "boy" in two translations of "Madame Bovary" — E. Marx-Aveling's translation and EText's translation. The first text contains 55 occurrences of "boy", which correspond to 38 occurrences of "boy" and 17 occurrences of its paraphrases ("son", "young fellow" and "youngster"). This implies that even high frequency for a word in one part of a parallel corpus does not guarantee a high count for appearances of its paraphrases in another part of the corpus. This feature of the corpus hampers the process of collecting reliable statistics of paraphrase pair appearances.

On the positive side, these non-translated words, which we call *anchors*, can greatly assist in the matching process, since they reduce the number of alternative mappings between sentence units. For example, given two sentences of length three, there are six possible on-to-one mappings between their words. However, if these two sentences share one word in common, only two mappings are plausible. The efficient

use of this corpus feature can benefit a paraphrase extraction algorithm.

We describe below a method of paraphrase extraction, which exploits these features of our corpus.

3.4 Method for Paraphrase Extraction

Given the aforementioned differences between translations, our method builds on similarity in the local context, rather than on global alignment. Consider the two sentences in Figure 3.2.

And finally, dazzlingly white, it shone high above them in the empty [?].
It appeared white and dazzling in the empty [?].

Figure 3.2: Fragments of aligned sentences

Analyzing the contexts surrounding "[?]"-marked blanks in both sentences, one expects that they should have the same meaning, because they have the same premodifier "empty" and relate to the same preposition "in". In fact, the first "[?]" stands for "sky", and the second for "heavens". Generalizing from this example, we hypothesize that if the contexts surrounding two phrases look similar enough, then these two phrases are likely to be paraphrases. The definition of the context depends on how similar the translations are. Once we know which contexts are good paraphrase predictors, we can extract paraphrase patterns from our corpus.

Examples of such "good" contexts are verb-object relations and noun-modifier relations, which were traditionally used in word similarity tasks from non-parallel corpora (Pereira, Tishby, and Lee, 1993; Hatzivassiloglou and McKeown, 1993).

However, in our case, more indirect relations can also be clues for identifying paraphrasing, because we know *a priori* that input sentences convey the same information. For example, in the sentences from Figure 3.3, the verbs "ringing" and "sounding" do not share identical subject nouns, but the modifier of both subjects "Evening" is identical. Can we conclude that identical modifiers of the subject imply verb similarity? To address this question, we need a way to identify contexts that are good predictors of paraphrases in a corpus.

To find "good" contexts, we can analyze all contexts surrounding identical words in the pairs of aligned sentences, and use these contexts to learn new paraphrases. This provides the basis for a bootstrapping mechanism. Starting with identical words in aligned sentences which form our initial seed of *anchors*, we can learn the "good" contexts, and in turn use them to learn new paraphrases. These new paraphrases extend our original set of anchors, and are used to learn more contexts. Anchors play two roles in this process: first, they are used to learn context rules; second, anchors are used in application of these rules, because the rules contain information about the equality of words in context.

People said "The Evening Noise is sounding, the sun is setting."
"The evening bell is ringing," people used to say.

Figure 3.3: Fragments of aligned sentences

This method of co-training (Blum and Mitchell, 1998) has been previously applied to a variety of natural language tasks, such as word sense disambiguation (Yarowsky, 1995), lexicon construction for information extraction (Riloff and Jones, 1999), and named entity classification (Collins and Singer, 1999). In our case, the

co-training process creates a binary classifier, which predicts whether or not a given pair of phrases is a pair of paraphrases.

Our model is based on the DLCoTrain algorithm proposed by (Collins and Singer, 1999), which applies a co-training procedure to decision list classifiers for two independent sets of features. In our case, one set of features describes the paraphrase pair itself, and another set of features corresponds to contexts in which paraphrases occur. These features and their computation are described below.

3.4.1 Feature Extraction

Our paraphrase features include lexical and syntactic descriptions of the paraphrase pair. The lexical feature set consists of the sequence of tokens for each phrase in the paraphrase pair; the syntactic feature set consists of a sequence of part-of-speech tags where equal words and words with the same root are marked. For example, the value of the syntactic feature for the pair ("the vast chimney", "the chimney") is ("DT_1 JJ NN_2", "DT_1 NN_2"), where indices indicate word equalities. We believe that this feature can be useful for two reasons: first, we expect that some syntactic categories can not be paraphrased by another syntactic category. For example, a determiner is unlikely to be a paraphrase of a verb. Second, this description is able to capture regularities in phrase level paraphrasing. In fact, a similar representation was used by (Jacquemin, Klavans, and Tzoukermann, 1997) to describe term variations as mentioned above.

The contextual feature is a combination of the left and right lexico-syntactic contexts surrounding actual known paraphrases. There are a number of context representations that can be considered as possible candidates: lexical n-grams, POS-

ngrams and parse tree fragments. The natural choice is a parse tree; however, existing parsers perform poorly in our domain[5]. Part-of-speech tags provide the required level of abstraction, and can be accurately computed for our data. The left (right) context is a sequence of part-of-speech tags of n words, occurring on the left (right) of the paraphrase. As in the case of syntactic paraphrase features, tags of identical words are marked. For example, when $n = 2$, the contextual feature for the paraphrase pair ("*comfort*", "*console*") from Figure 2.9 sentences is $left_1$="VB_1 TO_2", ("*tried to*"), $left_2$="VB_1 TO_2", ("*tried to*"), $right_1$="PRP\$$_3$ $,_4$", ("*her,*") $right\_context\$_2$="PRP\$$_3$ $,_4$", ("*her,*"). In the next section, we describe how the classifiers for contextual and paraphrasing features are co-trained.

3.4.2 The co-training algorithm

Our co-training algorithm has three subroutines: initialization, training of the contextual classifier and training of the paraphrasing classifiers.

Initialization Words which appear in both sentences of an aligned pair are used to create the initial "seed" rules. Using the initial set of anchors, we create a set of positive paraphrasing examples, such as $word_1$=*tried,* $word_2$=*tried.* However, training of the classifier demands negative examples as well; in our case it requires pairs of words in aligned sentences which are not paraphrases of each other. To find negative examples, we match the anchors in the alignment against all different words in the aligned sentence, making the assumption that *anchors* can match only each other, and not any other word in the aligned sentences. For example, "*tried*" from

[5]To the best of our knowledge all existing statistical parsers are trained on WSJ or similar type of corpora. In the experiments we conducted, their performance significantly degraded on a portion of our corpus — literary texts.

the first sentence in Figure 2.9 does not correspond to any other word in the second sentence but "*tried*". Based on this observation, we can derive negative examples such as $word_1$=*tried,* $word_2$=*Emma* and $word_1$=*tried,* $word_2$=*console*. Given a pair of identical words from two sentences of length n and m, the algorithm produces one positive example and $(n-1)+(m-1)$ negative examples.

Training of the contextual classifier Using these initial seed rules, we record contexts around positive and negative paraphrasing examples. From all the extracted contexts, we must identify the ones which are strong predictors of their category. Following (Collins and Singer, 1999), filtering is based on the strength of the context and its frequency. The positive strength of a context x is defined as count$(x+)$/count(x), where count$(x+)$ is the number of times context x surrounds positive examples (paraphrase pairs) and count(x) is the frequency of the context x. The negative strength of a context is defined in a symmetric manner. For each of the positive and the negative categories we select the k rules (for parameter estimation, see 3.5.3) with the highest frequency and strength higher than the predefined threshold. Examples of selected context rules are shown in Figure 3.4.

A parameter of the contextual classifier is maximal context length. We observed that for some rules a shorter context works better. Therefore, when recording contexts around positive and negative examples, we record all the contexts with length less than or equal to the maximal length.

Because our translation corpus consists of several books, created by different translators, we expect that the similarity between translations varies from one book to another. This implies that contextual rules should be specific to a particular pair of translations. Therefore, we train the contextual classifier for each pair of translations

separately. We do not perform such a separation when we extract paraphrases for news articles, because our comparable corpus does not include information about writer identities.

$left_1 = (\text{VB}_0\ \text{TO}_1)$ $right_1 = (\text{PRP\$}_2\ ,)$ $left_2 = (\text{VB}_0\ \text{TO}_1)$ $right_2 = (\text{PRP\$}_2\ ,)$
$left_1 = (\text{WRB}_0\ \text{NN}_1)$ $right_1 = (\text{NN}_2\ \text{IN})$ $left_2 = (\text{WRB}_0\ \text{NN}_1)$ $right_2 = (\text{NN}_2\ \text{IN})$
$left_1 = (\text{VB}_0)$ $right_1 = (\text{JJ}_1)$ $left_2 = (\text{VB}_0)$ $right_2 = (\text{JJ}_1)$
$left_1 = (\text{IN}\ \text{NN}_0)$ $right_1 = (\text{NN}_2\ \text{IN}_3)$ $left_2 = (\text{NN}_0\ ,)$ $right_2 = (\text{NN}_2\ \text{IN}_3)$

Figure 3.4: Example of context rules extracted by the algorithm.

Training of the paraphrasing classifier Context rules extracted in the previous stage are then applied to the corpus to derive a new set of pairs of positive and negative paraphrasing examples. Applications of the rule are performed by searching sentence pairs for subsequences which match the left and right parts of the contextual rule, and are less than N tokens apart. For example, applying the first rule from the sentences in Figure 3.4 to sentences from Figure 2.9 yields the paraphrasing pair ("*comfort*", "*console*"). Note that in the original seed set, the left and right contexts were separated by one token. This parameter N (when set to be greater than one) allows us to extract multi-word paraphrases.

For each extracted example, paraphrasing rules are recorded and filtered in a similar manner as contextual rules. Examples of lexical and syntactic paraphrasing rules are shown in Figure 3.5 and in Figure 3.6. After extracted lexical and syntactic paraphrases are applied to the corpus, the contextual classifier is retrained. New

paraphrases not only add more positive and negative instances for training the contextual classifier, but also revise contextual rules for known instances based on new paraphrase information.

The iterative process is terminated when no new positive paraphrases are discovered or the number of iterations exceeds a predefined threshold.

$(NN_0\ POS\ NN_1) \leftrightarrow (NN_1\ IN\ DT\ NN_0)$
King's son son of the king
$(IN\ NN^0) \leftrightarrow (VB^0)$
in bottles bottled
$(VB_0\ to\ VB^1) \leftrightarrow (VB_0\ VB^1)$
start to talk start talking
$(VB_0\ RB_1) \leftrightarrow (RB_1\ VB_0)$
suddenly came came suddenly
$(VB\ NN^0) \leftrightarrow (VB^0)$
make appearance appear

Figure 3.5: Morpho-Syntactic patterns extracted by the algorithm. Lower indices denote token equivalence, upper indices denote root equivalence.

Parameters There are eight parameters that can be set to adjust the performance of the algorithm:

- *The maximal length of the context* — a number of words surrounding a target word considered by the contextual classifier as its context
- *The minimal frequency of a contextual rule* — a number of times a contextual rule has to appear to be considered for paraphrase extraction
- *The minimal predictive power of a contextual rule* — a threshold on predictive power of a contextual rule; the predictive power is defined to be a number

(dash, look), (sending, working), (bored, tendered), (alienated, only estranged), (going, chase), (sought, tried), (mold, form), (tales, stories), (enough nerve, cheek), (bit, pursed), (swing, arc), (beadle, sexton), (grew, throve), (pleased, satisfied), (walks, outings), (withdrew, took), (every lecture, all the courses), (doing, grinding), (went back, climbed back up), (would open, opened), (close, warm), (expanded, opened), (returned, came home), (shut, spend), (every time, when), (couplets, verses), (beginning, outskirts), (excused, forgave), (knew, learned), (surgery, office), (plenty, no lack), (attain, gain), (free, able), (liked, wanted), (letter, mail), (complained incessantly, constantly complained), (garret-window, attic window), (Natasie, Nastasie), (afraid, fearful), (cradle, horse), (suddenly remembered, quickly remember), (stoop, bend low), (mass, quantity), (separated, marked), (drapery, draperies), (shades, domes), (ill-groomed, tangled), (described, weaving), (close, shut), (respond, answered), (was off, set off), (when, where), (time, years), (would, might), (loved, adored), (stopped, stayed), (legends, captions), (pomps, splendors), (tendernesses, sensibilities), (entertainment, recreation), (week-nights, Sunday), (each, every), (ancient noble family, noblemen), (sewed, stitched away), (sentimental realities, genuine feeling), (weakness, indiscretions), (companions, schoolmates), (goodman, father), (peaked shoes, Eastern slippers), (tips, ends), (asked, begged), (glide, meander), (salvation, saving), (sick, disgusted), (wondrous passion, marvelous thing), (cornfield, wheat field), (bells, tinkling), (dressed, clad), (enshrine, nursing), (look, glance), (commonplace, flat), (energies, intensities), (wondered, marveled), (finger-glasses, finger bowls), (possessing, having), (people, citizens), (brought, delivered), (believing, recognizing), (come, been there), (shrew-mice, field mice), (prods, pokes), (rose, get up), (crackled, crunched)

Figure 3.6: Lexical paraphrases extracted by the algorithm from the translation corpus.

of contextual rule occurrences surrounding positive paraphrases divided by its total frequency

- *The minimal frequency of a lexical rule* — a number of times a pair of phrases has to be extracted by the positive contextual classifier

- *The minimal predictive power of a lexical rule* — a threshold on predictive power of a lexical rule; the predictive power is defined to be a number of times a pair of phrases was extracted by the positive contextual classifier divided by the number of times it is extracted by positive and negative classifiers

- *The minimal frequency of a part-of-speech rule* — a number of times a pair of

(auto, automobile), (closing, settling), (rejected, does not accept), (military, army), (IWC, International Whaling Commission), (Japan, country), (researching, examining), (harvesting, killing), (mission-control office, control centers), (father, pastor), (past 50 years, four decades), (Wangler, Wanger), (teacher, pastor), (fondling, groping), (Kalkilya, Qalqilya), (accused, suspected), (language, terms), (got, has), (setback, rebuff), (highlight, head), (all territories, Arab lands), (advanced, broke), (territory, West Bank), (armored vehicles, armor), (uprising, revolt), (talks, negotiations), (believe, say), (surrender, bow), (European officials, ammunition), (Palestinian leader Yasser Arafat s compound, movement), (EU, Arab League), (violence, Palestinian revolt), (erupted, began), (must, have to), (smash, blast), (clashes, gun battle), (move, Israel's attack), (cessation, halt), (militia, group), (resident, member), (terror, terrorism), (Arafat, Palestinian leader), (let, allow), (handed, issued), (Ramallah, Palestinian areas), (smash, punch), (exchange, return), (had been killed, have died), (hospital officials, hospital sources), (wanted, chief suspect), (23, 24), (50 percent, half), (spokesman, spokesperson), (has pretty seriously gotten off, is headed), (October, September), (plan, proposal), (fresh, significant), (place, room), (behavior, behaviour), (attempting, trying), (could, would), (authorities, police), (assassination, killing), (director, head), (Canadian soldiers, Canadians), (head, president), (U.N., United Nations), (Islamabad, Kabul), (goes, travels), (said, testified), (article, report), (chaos, upheaval), (Gore, Lieberman), (revolt, uprising), (more restrictive local measures, stronger local regulations) (countries, nations), (barred, suspended), (alert, warning), (declined, refused), (anthrax, infection), (expelled, removed), (White House, White House spokesman Ari Fleischer), (gunmen, militants)

Figure 3.7: Lexical paraphrases extracted by the algorithm from the news corpus.

part-of-speech sequences has to be extracted by the positive contextual classifier to be considered a paraphrase

- *The minimal predictive power of the part-of-speech rule* — a threshold on predictive power of a part-of-speech rule; the predictive power is defined to be the number of times a pair of part-of-speech sequences was extracted by the positive contextual classifier divided by the number of times it is extracted by positive and negative classifiers

The procedure used for parameter estimation is described in section 3.5.

3.5 Evaluation

So far, there is no consensus as to which evaluation methods and baselines should be used for the paraphrasing extraction task. One of the most commonly used measures is *accuracy* — judgment whether an extracted pair forms a paraphrase. This evaluation is usually complemented by other measures. (Lin and Pantel, 2001) compared paraphrases extracted by their system with paraphrases produced by humans. However, they noticed that the last evaluation is problematic, since "it is difficult for humans to generate a diverse list of paraphrases, given a starting formulation and no context." (Ravichandran and Hovy, 2002; Jacquemin, Klavans, and Tzoukermann, 1997) used task-based evaluation by measuring the impact of paraphrasing on Question-Answering and Information Retrieval systems. The coverage of the system is usually not reported, since the number of paraphrases present in the corpus is not known. The closest approximation to system coverage which was included was the number of extracted paraphrases, but this number has little comparative value, since each system uses a different type of corpus.

In this section, we describe the results of our evaluation using the traditional accuracy measure as well as a number of new measures. To ensure the soundness of the evaluation, we used several human judges and introduced alternative synthetic tests targeting the same dimension of the output.

We now describe in detail our corpora and techniques for parameter estimation. Then, we introduce our baselines, evaluation methodology and the results of the evaluation.

3.5.1 The Corpora

As stated above, our data consists of two distinct types of corpora: multiple English translations of foreign source texts and news articles about the same event. Since we used different methods for the collection and alignment of these corpora, we will describe them in turn.

Multiple Translations Corpus

The translation corpus is comprised of literary texts written by foreign authors. Many classical texts have been translated more than once, and these translations are readily available on-line. In our experiments we used nine translations,[6] among them, translations of Flaubert's *Madame Bovary*, Andersen's *Fairy Tales* and Verne's *Twenty Thousand Leagues Under the Sea*. Some of the translations were created during different time periods and in different countries.

Next, we performed sentence alignment. Sentences which are translations of the same source sentence tend to contain many identical words, which can greatly help in the matching process. Alignment is performed using dynamic programming (Gale and Church, 1991) with a weight function based on the number of anchors in a sentence pair. This simple method achieves good results for our corpus, because nearly all the sentences are translated in the same order and many words in corresponding sentences are identical. To measure the word overlap in aligned sentences, we used the *anchor density* of an aligned pair, defined to be the ratio of identical tokens to the union of tokens. The average anchor density in our corpus is quite high — 42%.

Alignment produced 25,962 pairs of sentences. To evaluate the accuracy of

[6]The corpus is available at `http://www.cs.columbia.edu/~regina/par`.

the alignment process, we analyzed 127 sentence pairs from the algorithm's output. 120(94.5%) alignments were identified as correct alignments. The aligned corpus consists of 1,087,530 words and 25,788 word tokens. Figure 3.8 gives some statistics on word distribution in the translations. Not surprisingly, word distributions follows a Zipfian distribution: half of the tokens appear only once or twice in one of the translations, and only one third of tokens appear more than five times in each translation. This statistic illustrates that in this corpus we are dealing with many low frequency words, and consequently with many low frequency pairs.

Frequency	First Translation	Second Translation	Overall
Overall (tokens)	21,458	19,978	25,788
Overall (words)	574,567	512,963	1,087,530
Once (tokens)	6,135	6,026	4,607
Twice (tokens)	4,046	3,785	6,297
More than five (tokens)	6,437	5,912	10,040

Figure 3.8: Word distribution in the translation corpus

News Corpus

To accumulate large amounts of news articles about the same event, we use news data collected by the Columbia Newsblaster system during a seven month period (from September 2001 until March 2002). Every day, Newsblaster downloads all articles from a variety of news websites, such as CNN, Washington Post and AP. Then, the system categorizes them into five categories (US news, international news, sports, technology and entertainment) and clusters articles by topic within each category. The clustering threshold is set sufficiently high so, that in practice, each cluster contains articles describing the same event. To ensure sufficient redundancy in an event description, we selected event clusters from two categories – US and international

news, since events in these categories usually have wider coverage than, for example, the category of technology news. This way we collected 5.5GB of data — 69,743 clusters encompassing 915,068 articles in total.

Now, we have to automatically find pairs of sentences which convey similar information. First, we break each cluster into pairs of articles; a cluster of n articles yields $n * (n-1)/2$ article pairs. Then, within each article pair we extract all pairs of sentences which have a predefined number of words in common. More specifically, we consider sentences as similar if the length of their token intersection is greater than or equal to half of the length of the shortest sentence in the sentence pair. Pairs of sentences in which one of the sentences fully contains another sentence are eliminated from the corpus. Our analysis of 120 sentence pairs collected using this technique revealed that 118 (98.3%) of the sentence pairs contain repeated information. Clearly, we can not attribute this accuracy to the sophistication of the method, but to the characteristics of the input data — lots of redundant information.

An obvious objection against this method is that it skews our corpus selection towards sentence pairs with many anchors, which may result in losing sentence pairs with interesting paraphrasing patterns. Unfortunately, it was the only option in our case. Existing tools for sentence alignment in related texts perform poorly on our corpus. For example, the output of one such tool, Simfinder (Hatzivassiloglou, Klavans, and Eskin, 1999), is too noisy to be useful in our task — a sample of 120 pairs produced by Simfinder contained only 34 (28.3%) valid pairs. Since our method yields highly accurate pairs, it satisfies our needs; we can always compensate the lack of coverage by running it over large amounts of related articles.

Frequency	Overall
Overall (tokens)	56,257
Overall (words)	18,336,123
Once (tokens)	4,860
Twice (tokens)	9,050
More than five (tokens)	34,652

Figure 3.9: Word distribution in the news corpus

This method yields 1,496,284 pairs of aligned sentences[7], containing 75,966,189 words and 64,217 tokens. From this corpus we selected a sample of 347,345 sentence pairs[8]; this sample is of the magnitude of the corpora used in machine translation. Figure 3.9 provides more statistics on the corpus distribution. Two thirds of the tokens appear more than five times in the corpus. This is in sharp contrast to the multiple translation data, where the majority of tokens appear once or twice.

Both types of the corpora are similar in terms of the length deviation between sentences in the same pair — in the news corpus the average ratio between length of the longer sentence in a pair and a shorter one is 1.3, compared to 1.4 in the translation corpus.

3.5.2 Preprocessing

We use a part-of-speech tagger and chunker (Mikheev, 1997) to identify noun and verb phrases in the sentences. These phrases become the atomic units used by the algorithm. We also record for each token its derivational root, using the CELEX (Baayen, Piepenbrock, and van Rijn, 1993) database. An example input is shown in Figure 3.10.

[7]The corpus is available at `http://www.cs.columbia.edu/~regina/comp`.

[8]We were unable to run one of the baseline systems on the full set, therefore we performed the evaluation on a subset. However, we used the full corpus to extract paraphrases for information fusion.

“_“_“ [[The_NN_THE Evening_NN_EVEN Bell_NN_BELL]] ((is_is_BE sounding_VB_SOUND)) ,_,_, [[the_DT_THE sun_NN_SUN]] ((is_is_BE setting_VB_SET)) ._._. ”_”_”
“_“_“ [[The_DT_THE evening_NN_EVEN bell_NN_BELL]] ((is_is_BE ringing_VB_RING)) ,_,_, ”_”_” [[people_NN_PEOPLE]] ((used_VB_USE)) ((to_TO_TO say_VB_SAY)) ._._. ”_”_”

Figure 3.10: Pair of aligned sentences after preprocessing

3.5.3 Parameter Estimation

To find the optimal parameter settings for our algorithm, we use a small development set with manually annotated paraphrases, which we previously used for the manual analysis of paraphrases (described in section 2.6). Our goal is to select parameter values which will increase the similarity between the system output and the paraphrases from the development set. We measure the “goodness” of the extracted paraphrases using the following objective function: the number of paraphrases extracted from the development set minus the number of pairs extracted by the algorithm which are identified as paraphrases by a human. In these terms, parameter estimation falls into a class of optimization problems where we aim to minimize the objective function.

Given the large number of elements in the space of all parameter settings, we cannot explore them exhaustively. While considering numerous methods developed for optimization problems (Press et al., 1997), we should keep in mind the dimensionality of the search space (eight in our case). Powell’s algorithm (Press et al., 1997), a commonly used heuristic grid search (Chen and Goodman, 1996), tends to overfit (Paciorek and Rosenfeld, 2000) the development set in the high-dimensional space, especially for small development sets. Furthermore, we are not dealing with a search space of continuously variable parameters, since the majority of our parameters have discrete values. In such conditions, *simulated annealing* (Metropolis et al.,

1953) seems to be a desirable search strategy.

Simulated annealing is an iterative process, which starts with some initial point in the search space. During each iteration, a point in the neighborhood of the current point is selected and the objective function is calculated. The neighborhood is defined by a generation mechanism, which selects new points by a small perturbation. If the change in cost function is negative, the transition is unconditionally accepted; if the cost function increases, the transition is accepted with a probability based upon the Boltzmann distribution: $exp(-\Delta E/kT)$, where k is a constant and ΔE is a change in energy. The parameter T is gradually lowered throughout the algorithm from a high starting value to an equilibrium, where no further changes occur. We need to specify the generation mechanism, initial temperature, stop criterion and temperature decrement between successive stages and number of transitions for each temperature value.

Following (Press et al., 1997), the original value of T has to be considerably larger than the expected differences in objective function from move to move. Since the maximum and minimum values of our objective function are known *a priori*,[9] T is assigned to be a difference between them. Following (Press et al., 1997), we proceed downward in steps amounting to a 10 percent decrease in T. The value of T is held for 44 points, or for 10 selected points, whichever comes first. The search stops when there is no improvement in objective function among 44 points. The generation mechanism we use is based on the variation of the downhill simplex method described in (Press et al., 1997) (p. 451).

[9]Maximal value zero is reached when paraphrases extracted by human are identical to the output. The minimal value is bounded by the negative of the number of all possible word pairs from aligned sentences.

The derived parameter values are shown in Figure 3.11.

Parameter	Value
The maximal length of the context	2
The minimal frequency of a contextual rule	20
The minimal predictive power of a contextual rule	88
The minimal frequency of a lexical rule	1
The minimal predictive power of a lexical rule	97
The minimal frequency of a part-of-speech rule	33
The minimal predictive power of the part-of-speech rule	96

Figure 3.11: Estimated Parameter Values

3.5.4 Results

Our algorithm produced 9,322 pairs of lexical paraphrases and 29 morpho-syntactic rules from the translation corpus, and 836 lexical paraphrases[10] and 27 rules from the news corpus. We aim to evaluate the quality of extracted paraphrases and the algorithm coverage. Ideally, we would want to compare directly the paraphrases extracted by the algorithm to all the paraphrases presented in the corpus. Since it is infeasible to manually identify all the paraphrases in the corpus, we can not perform such a direct comparison. To cope with this difficulty, we use alternative evaluation measures such as random sampling of the output and synthetic tests.

Baselines

To assess the merit of our method, we compare its performance against the state of the art methods from statistical machine translation systems (Melamed, 2001; Brown et al., 1993; Al-Onaizan et al., 1999). These methods can be easily used to acquire

[10] Our system extracted 56,942 pairs from the full news corpus.

paraphrases, by treating our data as a bilingual corpus, where each verbalization is considered to be in a different language. While these methods were developed for a different task, they are the closest in their aim to the paraphrase extraction task — finding words with the same meaning given a parallel corpus. More specifically, we compared our system against two techniques: the first of these is a technique for deriving bilingual lexicons (Melamed, 2001), and the second one is a full statistical machine translation system, Giza[11] (Brown et al., 1993; Al-Onaizan et al., 1999). While Melamed's system outputs a translation lexicon, Giza produces translation tables (t-tables) as part of its translation model, which can be viewed as translation lexicons.

These methods can be further adjusted to our task by directly exploiting features of our monolingual parallel corpus, mainly the availability of anchors. Our algorithm greatly benefits from them, since they reduce the space of possible matches. Machine translation methods do not usually rely on such identical words, because the same word rarely appears in two different languages. Even given two sentences with identical words, Giza does not "see" their identity, and tries to learn their translations. To overcome this limitation, we added to Giza a translation dictionary, consisting of pairs of identical tokens for each token that appears in the corpus. Giza was designed to use entries in the dictionary "as is", without trying to induce their translation from the corpus. This way Giza can benefit from the presence of anchors as our methods does. We did not have the code of Melamed's tool, which prevented us from doing a similar adjustment for his method.

[11] Giza parameters were set to be 5-5-1-5-5 (5 iterations of the first and second models, transfer, and 5 iterations of the third and fourth models) as recommended by Kevin Knight (personal communication).

Comparison issues An important question that arose in designing our experiments was how to deal with the difference between the output format of our method and that of the MT lexicon induction techniques. The MT systems usually produce a lexicon where each pair of words is associated with a score representing the confidence that they are translations of each other. Pairs with a low score are unlikely to be translations of each other. However, the MT lexicon induction methods do not give a threshold score of acceptable pairs, leaving this decision to other components of the MT system. This raises a difficulty in comparison, since our algorithm implicitly sets up a threshold, including in the output only pairs which it considers as paraphrases.

We first tried to establish a threshold using our development corpus, making an assumption that a score of any paraphrase from the development corpus should be higher than the threshold. In other words, the threshold would be a minimal score among paraphrases in the development set. This method reduced the size of the Giza lexicon only to 80% (from 223,668 to 175,441), including pairs with a score as low as 4.1e-7. Even a quick glance at these pairs reveals that this threshold yields a very noisy lexicon. Therefore, using it for comparison with our output is problematic, since in this case the threshold we selected contributes significantly to the low accuracy of the baseline system.

Instead of using threshold on the confidence score, we used k paraphrasing pairs with the highest confidence score, where k is equal to the length of our output. Even though this decision prevents us from comparing the coverage of the systems, we can fairly compare different systems on the same level of coverage.

Paraphrase Quality Evaluation

Our goal is to evaluate whether pairs extracted by a given algorithm are actual paraphrases. The context dependent nature of paraphrases complicates this seemingly simple task. Two words can be fully substitutable in one context, but may not be substitutable in other contexts. For example, the words "*lady*" and "*wife*" are plausible paraphrases in the sentences shown in Figure 3.12, but a judge may not consider them as paraphrases when they are presented in isolation. This makes context a critical component in judgment. Thus, the main dilemma in designing the evaluation is whether to factor in the context: should the human judge see only a paraphrase pair or should a pair of sentences containing the paraphrases also be given? If the latter option is followed, we evaluate whether the algorithm correctly performs phrase alignment in sentence pairs from the corpus. If the former option is taken, we also evaluate the context dependency for a given paraphrase pair. Since we did not attempt to assess the context dependency of extracted paraphrases, we decided to include context in our evaluation. In fact, an evaluation of a similar MT task — word-to-word translation — usually includes context (Melamed, 2001).

He traveled all over the world in hopes of finding a **lady**, but there was always something wrong.
He traveled all over the world to find a **wife**, but nowhere could he get what he wanted.

Figure 3.12: Example of sentences in which words "*wife*" and "*lady*" are substitutable

To evaluate the quality of produced paraphrases, we picked at random 250 paraphrasing pairs from the paraphrases produced by our algorithm from the translation corpus and 250 pairs from the news corpus. We also extracted a random sample

of the same size from the output of Melamed's system and the output of Giza. For each of these pairs, we extracted a pair of sentences from the corpus containing both words. To provide an appropriate context for judgment, we substituted a member of a paraphrasing pair in one of the sentences with the other paraphrase. The human judge was presented with two paraphrases as well as the two sentences containing them, and was asked to judge whether or not paraphrase substitution distorted the meaning of the sentence. An example of such a paraphrasing pair in the form presented to a judge is given in Figure 3.13.

He traveled all over the world in hopes of finding a **lady**, but there was always something wrong.
He traveled all over the world in hopes of finding a **wife**, but there was always something wrong.

Figure 3.13: Example of sentential contexts presented to a judge for evaluating the pair "*wife*" and "*lady*"

We mixed the outputs of the three systems and gave the resulting, randomly ordered 1500 pairs to six evaluators, all of whom were native speakers of English naive to linguistic theory. The authors were not among the judges. Each evaluator provided judgments on 500 pairs with context; thus, each pair was evaluated by two judges. The agreement was measured using the Kappa coefficient (Siegel and Castellan, 1988). Complete agreement between judges would correspond to K equals 1; if there is no agreement among judges, then K equals 0.

Our system significantly outperformed the baseline systems on two data sets. Averaging the decisions of two judges, the accuracy of our system on the translation corpus was 88%, for Melamed's accuracy was 63% and for Giza only 41%. On the

news corpus[12], the accuracy of our system was 72%[13], and Giza's accuracy was 36%. The agreement on the paraphrasing judgment was $K = 0.64$ which is reasonable agreement (Landis and Koch, 1977).

This increased precision is a clear advantage of our approach and shows that machine translation techniques cannot be used without modification for this task, particularly for producing multi-word paraphrases. There are several caveats that should be noted; Melamed's system was run without changes for this new task of paraphrase extraction and his system does not use chunk segmentation. He ran the system for three days of computation and his result may be improved with more running time since it makes incremental improvements on subsequent rounds. Giza is developed to operate over a large parallel corpus. Our translation corpus is not large enough for Giza, while our news corpus is not as clean as a typical bilingual parallel corpus.

Coverage Evaluation

The evaluation of coverage[14] is a problematic issue due to the lack of a complete list of the paraphrases present in the corpus. A plausible way to estimate the coverage in these conditions, is to perform an evaluation on a portion of the corpus which can be feasibly annotated by a human judge. We asked a human judge to extract para-

[12]Melamed's system did not produce reasonable results on the news corpus. The produced results are under investigation by Melamed (personal communication).

[13]The accuracy drop for comparable corpus is related to our method for collection of this corpus. We found that many sentence pairs are quite similar even though they are collected from different sources. As a result, wrong paraphrases extracted from these pairs have high counts and, thus, are selected by the algorithm as correct paraphrases.

[14]We didn't report on the coverage of our baselines system, since they give a non-negative score for any pair of words in the parallel corpus. Therefore, recall is determined by a threshold selection. Note that we compared the accuracy on the same recall level.

phrases from 50 randomly selected sentence pairs[15], and then counted how many of these paraphrases where predicted by our algorithm. The human judge extracted 114 paraphrases; 56(49.1%) of these 114 were identified as paraphrases by our algorithm.

Manual paraphrase annotation is very time consuming; therefore, we can not significantly increase the size of our sample. We used an alternative evaluation method which annotates a corpus using the electronic thesaurus WordNet. Using this thesaurus, we marked all pairs of synonyms occurring in pairs of aligned sentences. Since synonyms in related sentences are likely to be paraphrases, the algorithm should be able to extract them. Other lexical relations in WordNet may yield paraphrases, but it is not clear *a priori* which relations do so. Consequently, we limited ourselves to the noun synonymy relation to ensure the accuracy of annotations. Even though synonyms produce only a fraction of all possible paraphrases, this method of annotation produced a more sizable sample than the previous annotation technique, allowing us to get a better estimation of the coverage of our algorithm. From the 972 unique pairs of synonyms which appeared in the translation corpus, our method identified 354 (36.4%) paraphrases. For synonyms which occur more than once, the coverage is 54% (231 from 432). Synonyms with frequency higher than four, were discovered in 71% of the cases (44 from 66).

To further investigate this dependency, we performed the following synthetic test: we selected an *anchor*, and substituted it with a pseudo-word (Schutze, 1992; Gale, Church, and Yarowsky, 1992) in the first sentence of the pair. This substitution introduces a new "artificial" pair of paraphrases consisting of the original anchor and the corresponding pseudo-word. For several frequency levels (1,2,3,4,5,10,20), we

[15]Note that this set is different from the development set described in 3.5.3

selected 20 anchors which appear with that frequency in the corpus and substituted all of its occurrences in one side of the corpus with the corresponding pseudo-word. After every substitution, we ran our algorithm and tested whether the masked pair was identified as identical. For each run, we made only one substitution at a time, since removing many anchors at once can change performance for other reasons (see section 3.5.4). The graph in Figure 3.14 shows the average number of retrieved pairs of masked anchors as a function of their frequency. Our algorithm does a good job (> 70% retrieval rate) in extracting paraphrases which appear four times or more. The performance on infrequent paraphrases is lower, but is not terrible — paraphrases which occur only twice in the corpus are retrieved in 65% of the cases. In fact, many statistical machine translation system (Brown et al., 1993) do not even attempt to predict translations of tokens that appear only a few times.

Note that both synonym and anchor sampling produce consistent results in terms of retrieval coverage.

Extent of Anchor Dependence

Our algorithm performs uniformly well on the translations and the news corpus. Despite many differences between these two types of corpora, they are similar in one aspect — they both contain a significant amount of anchors. To be able to predict on which type of corpora our algorithm will produce good results, we need to investigate the dependency between anchor density and algorithm performance.

Since we do not have real data with different levels of anchor density, we created synthetic data by masking some portion of the existing anchors in our corpus to achieve the desired level of anchor density. We performed an evaluation on three

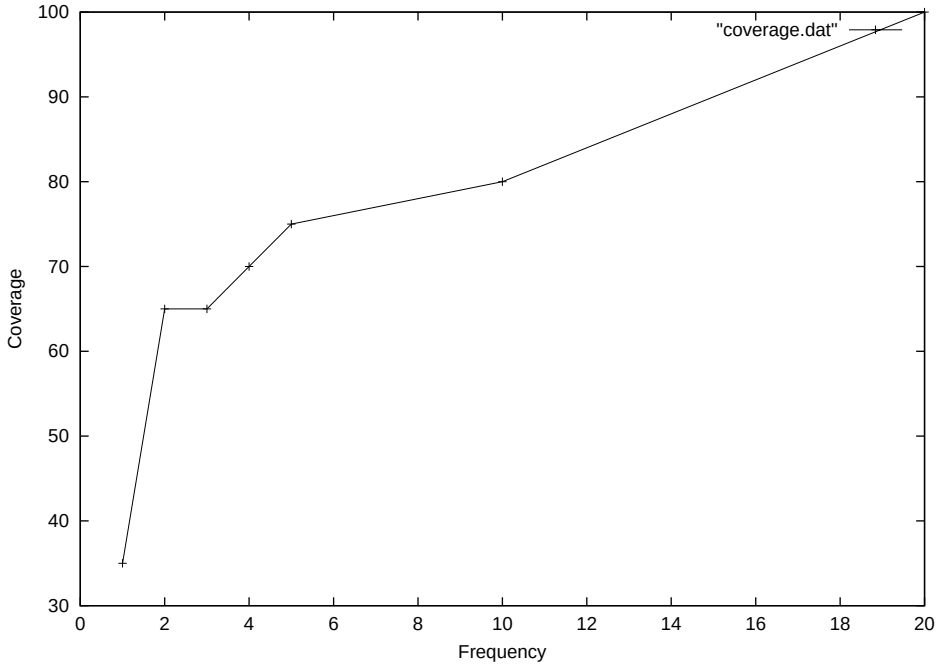

Figure 3.14: Retrieval rate for anchors on different frequency level (pseudo-word experiments).

levels of density — 0.4, 0.3, 0.2. The output was tested in terms of accuracy with random sampling on 250 pairs, and in terms of coverage of WordNet synonyms. The results of the evaluation shown in Figure 3.15 suggest that our method sustains reasonable accuracy while dropping in coverage.

Anchor Density	Coverage	Accuracy
40%	35%	83%
30%	15.4%	78%
20%	10.8%	76%

Figure 3.15: Performance on the different levels of anchor density.

Analysis of Extracted Paraphrases

The accuracy of our system output allows us to address questions about the nature of paraphrasing phenomena we posed in the previous chapter, namely issues regarding the decomposition hypothesis, lexical mechanisms yielding paraphrases and types of compositional rules.

Decomposition hypothesis In the previous chapter, we proposed the decomposition hypothesis. This hypothesis states that a pair of sentences which are paraphrases of each other can typically be decomposed into pairs of words or phrases which are paraphrases of each other. In other words, it is unlikely (according to this hypothesis) that two sentences with the same meaning do not contain words or phrases which are paraphrases.

Previously, we tested this hypothesis by taking a sample of aligned sentences from our parallel corpus, and examining the granularity of the paraphrases contained in each pair of sentences. Analysis confirmed our hypothesis to the following extent: sentences with identical meaning are decomposable to short phrases, whereas inferentially similar sentences usually cannot be fully decomposed. The time-consuming nature of manual analysis prohibited us from a larger-scale confirmation of this hypothesis. Using automatically extracted paraphrases allows us to extend the scope of our analysis. However, we cannot use the same analysis techniques as we did in the manual experiments, for two reasons. First, our algorithm only extracts paraphrases of limited length; second, as we have discussed, it misses many of the paraphrases occurring only once in the corpus.

We thus used a different approach to testing the decomposition hypothesis,

which utilized the automatically extracted paraphrases. This approach is based on a measure we call the *overlap density* of a pair of sentences. The overlap density is defined to be the ratio of anchor pairs (contained in the sentences pair) to the total number of units; an anchor pair is either a pair of identical words, or a paraphrase identified by the algorithm, and the total number of units is the number of anchor pairs plus the number of words not contained in an anchor pair. Intuitively, the overlap density is intended to capture the extent to which the pair of sentences can be aligned, on a fine level. An overlap density of zero indicates that the sentences are not decomposable; an overlap density of one indicates that all subunits are aligned. Of course, this definition is only meaningful if the algorithm identifies a substantial fraction of existing paraphrases. Since our algorithm has difficulty retrieving paraphrases with low frequency in the corpus, we restricted our analysis to sentences which consist completely of words appearing at least twice in the corpus.

On this subset of sentences, the average overlap density was 0.65. Interestingly, the distribution of the overlap densities exhibited high variance. In particular, densities tended to fall in one of two groups, one with densities between 0.1 and 0.25, and the other with densities between 0.7 and 0.82. The high density group consisted mainly of literally translated sentences, while the low density group contained pairs of sentences having the same meaning in a looser sense. Typically, one would need to perform some amount of inference or application of world knowledge to identify the similarity in meaning between paired sentences from the low density group. These results are consistent with those obtained through the manual analysis.

Lexical mechanisms yielding paraphrases Numerous manually-created lexical resources provide classification of lexico-semantic relations between English words. These resources are wide in scope and are constantly extended and refined (Harabagiu, Miller, and Moldovan, 1999; Harabagiu and Moldovan, 2000)). Thus, it seems natural to use these thesauri as sources for paraphrases; we only need to know what relations in these resources can produce paraphrases. To address this question, we analyzed types of relations that hold between paraphrases in our output in terms of the WordNet thesaurus. We focused our analysis on WordNet, since it is one of the most frequently used resources in the NLP community.

To get more insights on this question, we selected 112 paraphrasing pairs which occurred at least 20 times in our corpus such that the words comprising each pair appear in WordNet. All of these pairs were identified as correct paraphrases by a native speaker of English. The cutoff of 20 was chosen to ensure that the identified pairs are general enough and not idiosyncratic. We use the frequency threshold to select paraphrases which are not tailored to one context. Examples of paraphrases and their WordNet relations are shown in Figure 3.16. Only 40(35%) paraphrases are synonyms, 36(32%) are hyperonyms, 20(18%) are siblings in the hyperonym tree, 11(10%) are unrelated, and the remaining 5% are covered by other relations. These figures quantitatively validate our intuition that synonymy is not the only source of paraphrasing, suggesting that words from distinct nodes can also be paraphrases. In fact, we found paraphrases consisting of words from nodes very far apart in the WordNet tree — 12 links. Obviously, the majority of words at such a distance are not paraphrases, and this means that the commonly used notion of distance as the shortest path between two nodes can not be used to extract paraphrases from the

WordNet tree. We note that this does not refute the possibility of another notion of distance which corresponds more closely to the paraphrasing relation. The extracted paraphrases can inform the development of such a measure.

In addition to lexical paraphrases, our algorithm extracts compositional paraphrases, analyzed in the next section.

Synonyms: (rise, stand up), (hot, warm)
Hyperonyms: (landlady, hostess), (reply, say)
Siblings: (city, town), (pine, fir)
Unrelated: (sick, tired), (next, then)

Figure 3.16: Lexical paraphrases extracted by the algorithm.

Compositional Rules The compositional rules extracted by the algorithm fall in two categories — noun phrase and verb transformations. Noun phrase transformations include various cases of modifier deletions, permutations and rewriting of noun-noun pairs and adjective-noun pairs into a noun with attached prepositional phrase. Beyond deletions and permutation, compositional rules dealing with verbs cover alternations in tense (e.g. past continuous into past tense), changes in modality and rewriting of verb-infinitive pairs into a verb with a gerund. The full list of extracted templates is shown in Figure 3.17.

These rules match descriptions of syntactic rules found in the linguistic literature (Harris, 1981b). However, the rules we extract are only a partial list of existing rules, since we only rely on shallow syntactic representation. We cannot compute the degree of overlap with Harris' paraphrase ontology since the mapping between his paraphrase class descriptions and paraphrasing rules is not obvious (see page 2.4.2). Therefore, we can not accurately assess how many rules our algorithm is missing.

Since our algorithm extracts only rules encoding local transformations, we miss all sentence-level rules such as the active-passive transformation. We also didn't learn the lexical constraints on the use of these rules, which is essential in many applications. We leave these issues to future work.

3.6 Conclusions and Future Work

In this chapter, we presented a method for corpus-based identification of paraphrases from a comparable corpus. We showed that a co-training algorithm based on contextual and lexico-syntactic features of paraphrases achieves high performance on our data. In contrast to earlier work, our approach allows for identification of multi-word paraphrases, in addition to single-word paraphrases, as well as extraction of compositional rules. We found that our method can handle translations along a continuum of similarity, ranging from parallel translations to comparable corpora. We also showed that our method significantly outperforms state-of-the-art MT techniques applied to paraphrasing extraction task.

Besides the performance evaluation, we also studied the properties of the algorithm. We found that a decrease in the anchor density causes a slow decrement in accuracy, mostly affecting the coverage of the algorithm. This is desirable behavior for a paraphrase acquisition algorithm. We also found that our method can accurately identify paraphrasing pairs if their frequency in the corpus is as low as five times.

Our approach identifies mainly phrasal lexical paraphrases and local compositional rules. An obvious future direction of research is a method for extracting compositional lexico-syntactic paraphrases from a parallel or comparable corpus. Phrasal

paraphrases extracted by our method could facilitate learning of such rules since knowledge about phrasal equivalence helps to reveal structural similarity of the sentences which contain them.

Our method uses a parallel corpus as a source of paraphrases. A more ambitious goal is to use the parallel corpus as a seed for paraphrase extraction from other resources such as large-scale knowledge sources and non-parallel corpora. For example, paraphrases extracted from our corpus can inform the development of a WordNet distance which corresponds closely to the paraphrasing relation. This would benefit multiple applications which rely on WordNet as a source of paraphrases.

In the next chapter, we present information fusion, an application which uses as its principal source of knowledge the automatically derived paraphrasing thesaurus produced by our algorithm. This application demonstrates the practical value of paraphrasing information.

(DT JJ NN_2 NN_3)↔(NN_2 NN_3 NN)
(DT_0 JJ NN_2)↔(DT_0 JJ NN_2)
(DT JJ_1 NN_2)↔(DT JJ_1 NN_2)
(DT JJ NN_2)↔(DT JJ NN_2)
(DT_0 NN^1)↔(DT_0 NN^1)
(DT NN_1)↔(DT NN_1)
(DT NN_1)↔(NN POS NN_1)
(DT NN_1)↔(PRP$ NN_1)
(IN NN^1)↔(VB^1)
(JJ NN_1)↔(JJ NN_1)
(MD_0 RB VB_2)↔(MD_0 RB VB_2)
(MD RB VB_2)↔(MD RB VB_2)
(MD^0 VB_1)↔(MD^0 VB_1)
(MD VB_1)↔(MD VB_1)
(MD VB_1)↔(MD VB_1)
(MD VB_1)↔(VB_1)
(NN NN_1)↔(NN NN_1)
(NN_0 POS NN_3)↔(DT NN_3 IN NN_0)
(NN^0)↔(NN^0)
($PRP\$_0$ JJ_1 NN)↔($PRP\$_0$ JJ_1 NN)
($PRP\$_0$ JJ NN_2)↔($PRP\$_0$ JJ NN_2)
(PRP$ NN_2)↔(NN POS NN_2)
(VB_0 RB VB_2)↔(VB_0 RB VB_2)
(VB_0 RB_1)↔(RB_1 VB_0)
(VB_0 RB)↔(VB_1)
(VB_0 TO VB^2)↔(VB_0 VB^2)
(VB^0 VB_1)↔(VB^0 VB_1)
(VB)↔(VB NN) NIL ((1 0)))
(VB^0)↔(VB TO VB^0)
(VB^0)↔(VB VB^0)

Figure 3.17: Morpho-Syntactic patterns extracted by the algorithm. Lower indices denote token equivalence, upper indices denote root equivalence.

Chapter 4

Sentence Fusion for Multidocument Summarization

This chapter focuses on a method for sentence fusion. Sentence fusion is part of a multi-document summarization system, MultiGen (McKeown et al., 1999; Barzilay, McKeown, and Elhadad, 1999; Hatzivassiloglou, Klavans, and Eskin, 1999; Barzilay, Elhadad, and McKeown, 2002). We start our presentation with a description of MultiGen, since it provides context for the development and testing of information fusion.

4.1 MultiGen

One of the many benefits of the web is that it provides many on-line news sources that are updated whenever a new story is available. Such sites include CNN, Reuters, New York Times, and many others. For some people, these sites have replaced printed newspapers as their main source of news, and for others, they provide an additional

source of news. There are so many such news sites and so many articles posted every day that it is impossible to read them all.

Furthermore, a large portion of the articles on these sites describe the same events, since major news events are covered by most news agencies. Hence, summaries that synthesize common information across related documents in one short text significantly cut down reading time for people. MultiGen was designed to produce such summaries: to automatically generate a fusion of similar information across multiple related documents into a concise text. While MultiGen is a standalone component that could be integrated in various applications, the primary large-scale system in which it currently operates is the Columbia Newsblaster (McKeown et al., 2002).

Newsblaster is a publicly accessible system that has been developed at Columbia University to help users find and browse news that is of the most interest to them. Rather than traversing many different sites to find news of interest, a user can turn to Newsblaster, which agglomerates information from various sites. The system automatically collects, clusters, categorizes, and summarizes news from several sites on the web, and it provides users with a friendly interface to browse the results. Newsblaster uses two separate summarizers for different clusters depending on the type of documents in the cluster as determined by a router. MultiGen is used when there is a high enough degree of similarity between articles in a cluster. The other system, DEMS (Schiffman, Nenkova, and McKeown, 2002), is used for sets of articles that are more loosely related, and also for biographical documents.

In this scenario, fusion of common information across related articles is a valuable summarization strategy. Multiple news articles about the same event usually agree on the key aspects of the event, but they vary in peripheral details. Thus,

repeated information is a good indicator of its importance to the event, and can be used for summary generation. Although the generated summaries do not indicate differences between articles, they can help users to determine which stories are important to them. If users want to learn more, Newsblaster provides links to the original articles, so they can find all source articles pertaining to a given story.

At the time MultiGen was developed, most approaches to summarization focused on single document summarization and were based on sentence extraction (e.g., (Paice, 1990; Kupiec, Pedersen, and Chen, 1995; Marcu, 1997)). While sentence extraction may be adequate for single document summarization, it will not work effectively for multiple document summarization. Any individual document does not contain explicit comparisons with all other documents which can be extracted; alternatively, if all sentences containing similar information are extracted (Mani and Bloedorn:, 1997; Yang, Pierce, and Carbonell, 1998), this would make for lengthy and repetitive reading.

Unlike previous work in the area, MultiGen generates the summary by reusing and altering phrases from input articles, creating a more cohesive text. The design of MultiGen is unique in its integration of machine learning and symbolic techniques to identify similar sentences, intersection of similar phrases within sentences, and language generation to reformulate the wording of the summary. An overall system evaluation presented in Chapter 6 shows MultiGen performs well according to various measures.

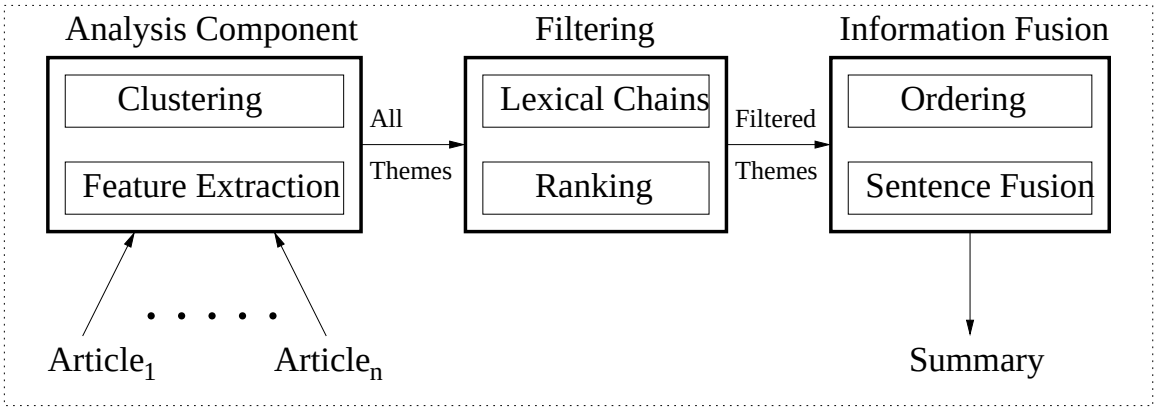

Figure 4.1: MultiGen Architecture

4.1.1 MultiGen Architecture

MultiGen follows a pipeline architecture, shown in Figure 4.1. The analysis component of the system, Simfinder (Hatzivassiloglou, Klavans, and Eskin, 1999), breaks documents into smaller text units and then computes a similarity metric across text units, regardless of the source document. Once groups of similar paragraphs are identified, the system selects a subset of the groups to be included in the summary depending on the desired compression length. The selected groups are passed to the generation component which further identifies and chooses information to be reformulated as coherent text. Below we describe each component in more detail.

4.1.2 Analysis Component: Simfinder

First, the analysis component of the system, Simfinder, identifies *themes*, groups of sentences from different documents that contain repeated information. Each theme corresponds to one sentence in the output summary, generated by a fusion component. An example of a theme is shown in Figure 4.2. There may be many themes for a

1. IDF Spokeswoman did not confirm this, but said the Palestinians fired an anti-tank missile at a bulldozer.
2. The clash erupted when Palestinian militants fired machine-guns and anti-tank missiles at a bulldozer that was building an embankment in the area to better protect Israeli forces.
3. The army expressed "regret at the loss of innocent lives" but a senior commander said troops had shot in self-defense after being fired at while using bulldozers to build a new embankment at an army base in the area.
Fused sentence: Palestinians fired an anti-tank missile at a bulldozer.

Figure 4.2: An input set with the corresponding fused sentence.

set of articles. To identify themes, Simfinder breaks the article into sentences for comparison, and then computes a set of linguistic and positional features, which serve as input into the similarity algorithm. These features include primitive features such as word, stem and WordNet overlap, as well as composite features, which aim to capture matches on the syntactic level, such as subject-verb and verb-object relations. Simfinder constructs a vector for each pair of sentences, representing matches on each of the different features. A log-linear regression model is used to convert the evidence from the various features to a single similarity value. The model was trained on a large set of sentences which were manually marked for similarity. The output of the model is a listing of real-valued similarity values on sentence pairs. These similarity values are fed into a clustering algorithm that partitions the text units into clusters of closely related ones. The clustering is performed using a non-hierarchical clustering technique, the *exchange method* (Späth, 1985).

Note that usually theme sentences do not convey exactly the same information. As in the case of the theme shown in Figure 4.2, sentences usually include embedded phrase(s) containing information that is *not* common to all sentences in the theme.

Moreover, automatically computed themes frequently contain unrelated sentences. In fact, the evaluation of Simfinder on the task of computing similar sentence pairs reveals that the system reaches 49.3% precision at 52.9% recall (Hatzivassiloglou, Klavans, and Eskin, 1999). (We will discuss later how these factors influence sentence fusion.)

4.1.3 Filtering

Typically, Simfinder produces at least 20 themes given an average Newsblaster cluster of articles. To generate a summary of predetermined length, we induce a ranking on the themes. This ranking is based on theme size, similarity score, and significance. The first two of these scores are produced by Simfinder, and the significance score of the theme is computed using *lexical chains* (Barzilay and Elhadad, 1997), as the sum of lexical chain scores of theme sentences computed from the text to which the sentence originally belongs. Lexical chains — sequences of semantically related words — are tightly connected to the lexical cohesive structure of the text and have been shown to be useful for determining which sentences are important for single document summarization. Here, a theme that has many sentences that have been ranked by lexical chains to be important for a single document summary, is, in turn, given a higher significance score for the multi-document summary.

4.1.4 Information Fusion

Finally, MultiGen generates a text from the computed set of themes. During this process, the themes are first ordered into a coherent text. The ordering algorithm (described in Chapter 5) groups together cohesively related themes and then induces

chronological order among them. Next, the sentence fusion component generates a sentence conveying the information common among theme sentences.

In the next section, we describe the sentence fusion task. In Section 4.3, we provide an overview of related work. In Section 4.4, the fusion algorithm is introduced. The evaluation of our algorithm is presented in section 4.5.

4.2 Sentence Fusion

Given a group of similar paragraphs—a theme—the problem is to create a concise and fluent fusion of information with this theme, reflecting facts common to all paragraphs. An example of a fused sentence is shown in Figure 4.2. A straightforward method for fusion would be to pick a representative sentence that meets some criteria (e.g., a threshold number of common content words). In practice, however, any representative sentence will usually include embedded phrase(s) containing information that is *not* common to all sentences in the theme. These phrases may not be salient enough for a summary, or may bias the summary towards a particular detail. For example, picking any one sentence from the cluster in Figure 4.2 results in the inclusion of some unnecessary details. Therefore, to achieve our goal we need to identify phrases common to most theme sentences, and then combine them into a new sentence.

Obviously, sentence intersection in a set-theoretic way produces poor results. For example, the intersection of the first two sentences from the theme shown in Figure 4.2 is "*the fired anti-tank at the*". Besides being ungrammatical, it is impossible to understand what event this intersection describes. The inadequacy of the "bag of

words" approach to the fusion task motivates the use of a more elaborate representation for input sentences. (Radev and McKeown, 1998) demonstrated that this task is feasible when a detailed semantic representation of the input sentences is given. In their framework, sentences are compared on the level of their semantic representation, and selected semantic concepts are translated into an English sentence using concept-to-text generation methods. The system developed by (Radev and McKeown, 1998) operates in a limited domain (terrorist events), where information extraction systems can be used to interpret the source text. However, the task of mapping input text into a semantic representation in a domain-independent setting extends well beyond the ability of current analysis methods. These considerations suggest that we need a new method for the sentence fusion task. Ideally, such a method would not require full semantic representation. Rather, it would rely on input texts and knowledge that can be automatically derived from a corpus to generate a fusion sentence.

Our approach analyzes theme sentences and regenerates a new sentence containing just the information common to most sentences in a theme. It operates in three phases: parsing the sentences in each cluster with an existing statistical parser, aligning the resulting dependency trees (allowing for paraphrases), and finally, generating a new sentence from matched elements. Regeneration is achieved by selecting a sentence from the cluster as a skeleton, and modifying it to include only phrases matched across the entire theme while preserving the grammatical validity of the sentence. This approach generates a fusion sentence by reusing and altering phrases from the input articles, performing text-to-text generation.

4.3 Related Work

Text-to-text generation is an emerging area of NLP. Unlike traditional concept-to-text generation approaches, text-to-text generation methods take text as input, and transform it into a new text satisfying some constraints (e.g. length). In addition to information fusion, compression algorithms are another example of such methods (Knight and Marcu, 2000; Jing and McKeown, 2000).

Compression methods were developed for single-document summarization, and they aim to reduce a sentence by eliminating constituents which are not crucial for its understanding. These approaches are based on the observation that the "importance" of a sentence constituent can often be determined based on shallow features, such as its syntactic role and the words it contains. For example, in many cases a relative clause that is peripheral to the central point of the document can be removed from a sentence without significantly distorting its meaning. To determine which constituents can be reduced, these approaches typically use an aligned corpus containing pairs of original sentences and sentences which were manually compressed.

(Knight and Marcu, 2000) model reduction as a translation process using a noisy-channel model (Brown et al., 1993). In this model, a short (compressed) string is treated as a source and additions to this string are considered to be noise. The probability of a source string s is computed by the combination of a standard probabilistic context-free grammar score, which is derived from the grammar rules that yielded tree s, and a word-bigram score, computed over the leaves of the tree. The stochastic channel model creates a large tree t from a smaller tree s by choosing an extension template for each node based on the labels of the node and its children. In the decoding stage, the system searches for the short string s that maximizes $P(s|t)$,

which (for fixed t) is equivalent to maximizing $P(s) * P(t|s)$.

While this approach exploits only syntactic and lexical information, (Jing and McKeown, 2000) also rely on cohesion information, derived from word distribution in a text: phrases that are linked to a local context are kept, while phrases that have no such links are dropped. Another difference between these two methods is the extensive use of domain-independent knowledge sources in the latter. For example, a lexicon is used to identify which components of the sentence are obligatory to keep it grammatically correct. The corpus in this approach is used to estimate the degree to which the fragment is extraneous and can be omitted from a summary. A phrase is removed only if it is not grammatically obligatory, not linked to a local context, and has a reasonable probability of being removed by humans. In addition to reducing the original sentences, (Jing and McKeown, 2000) use a number of manually compiled rules to aggregate reduced sentences together; for example, reduced clauses might be conjoined with "*and*".

These two approaches were evaluated using different methods and were never compared against each other, but for both of them good performance was reported. (Jing and McKeown, 2000) directly compared reduction decisions made by their system to those of humans, and found that in 81.3% of the cases the reduction was a correct one. (Knight and Marcu, 2000) asked human judges to evaluate the quality of produced sentences in terms of grammaticality and their meaning. Their system significantly outperformed a baseline algorithm that produces compressions with highest word-bigram scores.

Information fusion exhibits similarities with compression algorithms in the ways it copes with the lack of semantic data in the generation process, relying on

shallow analysis of the input and statistics derived from a corpus. Clearly, the difference in the nature of both tasks and in the type of input they expect (single sentence versus multiple sentences) dictates the use of different methods. Having multiple sentences in the input poses new challenges — such as a need for sentence comparison —- but at the same time it opens up new possibilities for generation. While the output of existing compression algorithms is always a substring of the original sentence, information fusion may generate a new sentence which is not a substring of any of the input sentences. This is achieved by arranging fragments of several input sentences into one sentence.

The only other text-to-text generation approach with such a capability is that of Pang, Knight and Marcu . Their method operates over multiple English translations of the same foreign sentence, and is intended to generate novel paraphrases of the input sentences. Like information fusion, their method aligns parse trees of the input sentences and then uses a language model to linearize the derived lattice. The main difference between the two methods is in the type of the alignment: our algorithm performs local alignment, while the algorithm of (Pang, Knight, and Marcu, 2003) performs global alignment. The differences in alignment are caused by differences in input: (Pang, Knight, and Marcu, 2003) expect semantically equivalent sentences, while our algorithm operates over sentences which have only partial meaning overlap.

4.4 Sentence Fusion Algorithm

Sentence fusion follows the typical generation pipeline: content selection (what to say) and surface realization (how to say it). In traditional generation systems, a

content selection component chooses the semantic units to be verbalized, and a surface realization component translates these units into text by choosing appropriate syntactic constructions and wording. Having sentences as input prohibits the use of generation techniques operating over semantic input, but we can benefit from the textual information given in the input sentences for the tasks of syntactic realization, phrasing, and ordering. In fact, we use input sentences in two ways: to select the phrases conveying common information and to guide the way these phrases are combined into a fused sentence.

Our algorithm uses local alignment to identify repeated information across pairs of sentences in parsed form, from which we select fragments to be included in the fusion sentence. Instead of examining all possible ways to combine these fragments together, we select a sentence in the input which contains most of the fragments and transform its parsed tree into a desired form by eliminating non-essential information and augmenting it with information from other input sentences. Finally, we generate a sentence from this representation based on statistics derived from a large body of texts.

We describe the algorithm in three steps: identification of common information, fusion tree computation, and generation.

4.4.1 Identification of common information

First, we describe a routine which, given a pair of sentences, determines which sentence constituents convey information appearing in both sentences. This routine will be applied to all pairs of sentences in the input set of related sentences.

The intuition behind the routine is to compare all constituents of one sen-

tence to those of the other, and to select the most similar ones. Of course, how this comparison is done depends on the particular sentence representation used. A good sentence representation would emphasize sentence features that are relevant for comparison such as dependencies between sentence constituents, while ignoring irrelevant features such as constituent ordering. A representation which fits these requirements is a dependency based representation.

We will first detail how this representation is computed; describe a method for aligning dependency trees; and then present a method for selecting components conveying overlapping information.

Sentence Representation

In many NLP applications, the structure of a sentence is represented using phrase structure trees. An alternative representation is a *dependency tree* which describes the sentence structure in terms of dependencies between words. The similarity of the dependency tree to a predicate-argument structure makes it a natural representation for our comparison. This representation can be constructed from the output of a traditional parser. In fact, we developed a rule-based component that transforms the phrase-structure output of Collins's parser (Collins, 1997) to a dependency representation.

The process of comparing trees can be further facilitated if the dependency tree is abstracted to a canonical form which eliminates features irrelevant to the comparison. For example, the difference in grammatical features such as auxiliaries, number and tense have a marginal effect when comparing the meaning of sentences. Therefore, we represent in the dependency tree only non-auxiliary words with their

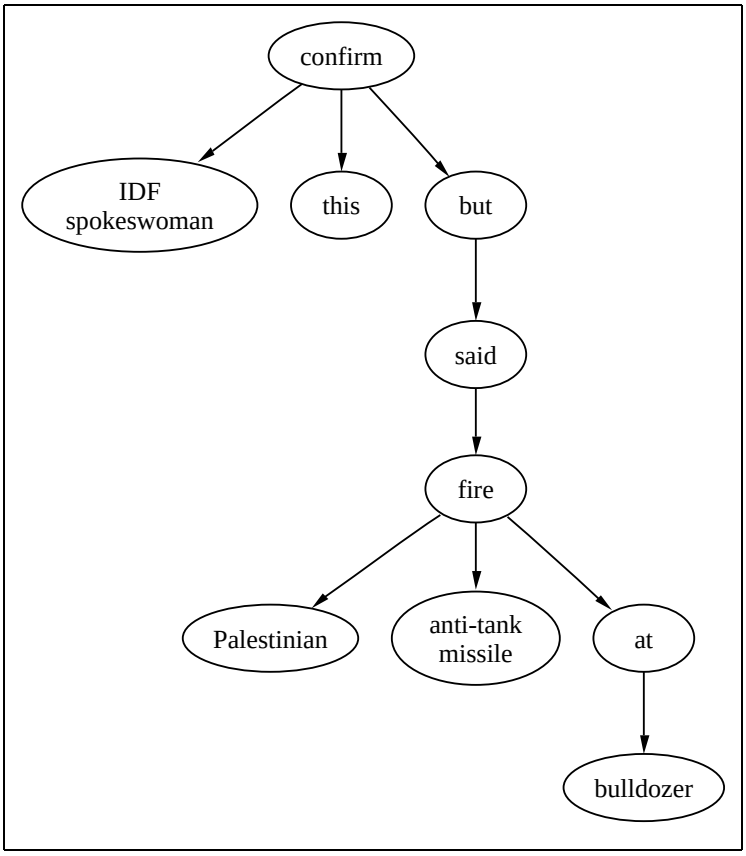

Figure 4.3: Dependency tree of the sentence "*The IDF spokeswoman did not confirm this, but said the Palestinians fired an anti-tank missile at a bulldozer on the site.*"

associated grammatical features; the eliminated auxiliary words can be recreated using these recorded features. In addition, we transform all the passive voice sentences to the active voice, changing the order of appropriate children. An example of a sentence and its dependency tree is shown in Figure 4.3.

Alignment

Our alignment of dependency trees is driven by two sources of information: a measure of similarity between two given words, and the similarity between the structure of

the dependency trees. More specifically, the similarity measures take into account more than word identity: these also identify similar words which appear as synonyms in WordNet or paraphrases according to the automatically constructed dictionary described in the previous chapter. In determining the structural similarity between two trees, we take into account the types of edges (which indicate the relationship between nodes); for example, it is unlikely that an edge connecting a subject and verb in one sentence corresponds to an edge connecting an adjective and noun in another sentence.

We now give an intuitive explanation of how our tree similarity function, denoted by Sim, is computed. If the optimal alignment of two trees is known, then the value of the similarity function is the sum of the similarity scores of aligned nodes and aligned edges. Since the best alignment of given trees is not known a priori, we select the maximal score among plausible alignments of the trees. Instead of exhaustively traversing the space of all possible alignments, we recursively construct the best alignment for trees of given depths, assuming that we know how to find an optimal alignment for trees of shorter depth. More specifically, at each point of the traversal we consider two cases. In the first case, two top nodes are aligned together and their children are aligned in an optimal way, applying the routine to shorter trees. In the second case, the top node of one tree is aligned with one of the children of the top node of the other tree; again we can apply our routine for this computation, since we decrease the length of one of the trees.

Before giving the precise definition of Sim, we introduce some notation. When T is a tree with root node v, we let $c(T)$ denote the set containing all children of v. For a tree T containing a node s, the subtree of T which has s as its root node is

denoted by T_s.

Given two trees T and T' with root nodes v and v', respectively, the similarity $Sim(T, T')$ between the trees is defined to be the maximum over the three expressions $NodeCompare(T, T')$, $\max\{Sim(T_s, T') : s \in c(T)\}$, and $\max\{Sim(T, T'_{s'}) : s' \in c(T')\}\}$. The function $NodeCompare(T, T')$ is defined by

$$NodeCompare(T, T') =$$
$$NodeSim(v, v') + \max_{m \in M(c(T), c(T'))} \left[\sum_{(s,s') \in m} EdgeSim((v, s), (v', s')) + Sim(T_s, T'_{s'}) \right]$$

where $M(A, A')$ is the set of all possible matchings between A and A', and a matching (between A and A') is a subset m of $A \times A'$ such that for any two distinct elements $(a, a'), (b, b') \in m$, both $a \neq b$ and $a' \neq b'$. Word and edge similarity scores were manually derived based on our linguistic intuition, and were then subsequently hand-tuned using a small development corpus. Intuitively, the maximization in the $NodeCompare$ formula searches for the best possible alignment for the children nodes of the given pair of nodes. $NodeSim(v, v')$ is equal to 1, when the words of the nodes v and v' are equal or are synonyms in WordNet; 0.5, when they are paraphrases according to our dictionary; and -0.01 otherwise. $EdgeSim(e, e')$ is equal to 0.3, when the edges are of identical type and they are both of either subject-verb or verb-object type; 0.2, in all other cases where edges are of the same type; and 0 otherwise. In the base case, when one of the trees has depth one, $NodeCompare(T, T')$ is defined to be $NodeSim(v, v')$.

The computation of the similarity function Sim is performed using bottom-up dynamic programming, where input trees with lower depth are computed first. The alignment routine returns the similarity score of the trees as well as the optimal

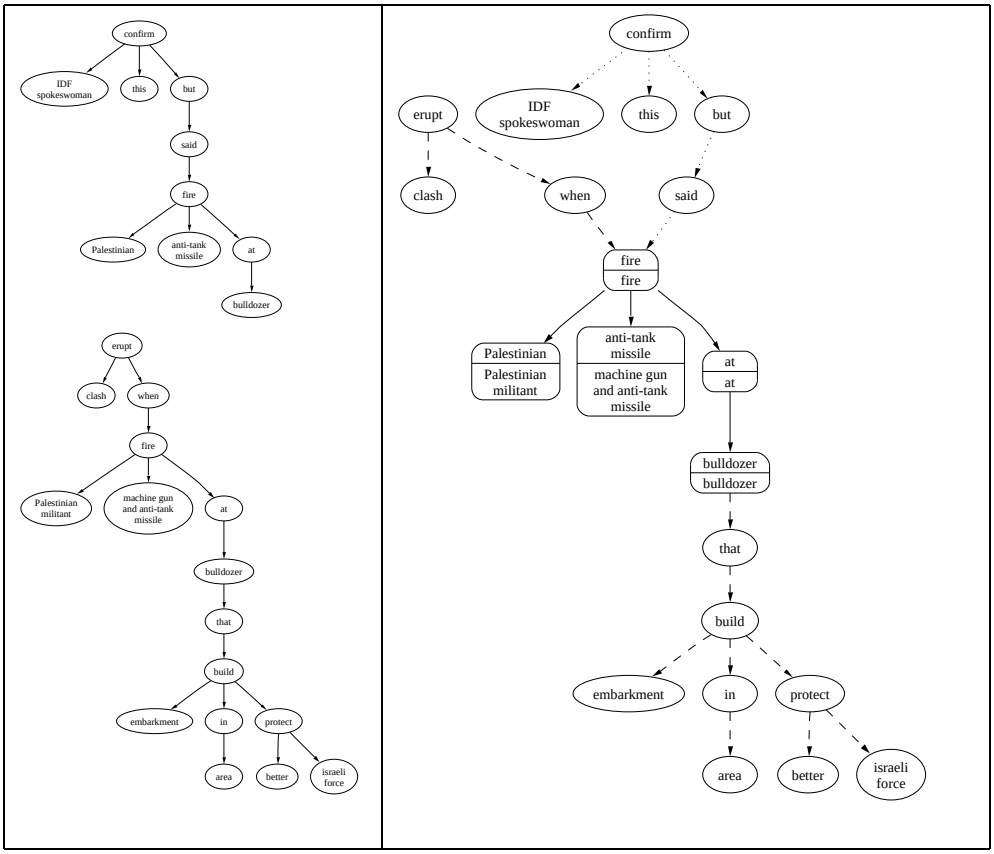

Figure 4.4: Two dependency trees and their alignment tree.

mapping between the subtrees of input trees. Figure 4.4 shows two dependency trees and their alignment.

As is evident from the *Sim* definition, we are only considering one-to-one node "matchings": every node in one tree is mapped to at most one node in another tree. While this restriction significantly decreases the complexity[1] of the optimal

[1]The complexity of our algorithm is polynomial in the number of nodes.
Let n_1 denote the number of nodes in the first tree, and n_2 denote the number of nodes in the second tree. We assume that the branching factor of a parse tree is bounded above by a constant. The function *NodeCompare* is evaluated only once on each node pair. Therefore, it is evaluated $n_1 * n_2$ times totally. Each evaluation is computed in constant time, assuming that values of the function for node children are known. Since we use memoization, the total time of the procedure is $O(n_1 * n_2)$.

alignment computation, it may lead to wrong alignments. Our manual analysis of paraphrased sentences described in Chapter 2 revealed that some paraphrases are not decomposable to words, forming one-to-many or many-to-many paraphrases. In our analysis, we observed that such alignments most frequently occur in pairs of noun-phrases and pairs including verbs with particles (e.g. ("*stand up*", "*rise*")). To improve our alignment, during a preprocessing stage we flatten subtrees containing such phrases into one node.

Another important property of our algorithm is that it produces a local alignment. Local alignment aims to map local regions with high similarity to each other rather than to create an overall optimal global alignment of the entire tree. This strategy is more meaningful when only partial meaning overlap is expected between input sentences, as in typical information fusion input. Only these high similarity regions, which we call intersection subtrees, are selected to be included in the fusion sentence.

4.4.2 Fusion Tree Computation

The next question we address is how to put together intersection subtrees. Obviously, among the many possible combinations, we are interested only in those combinations which yield semantically sound sentences and do not distort the information repeated in the input sentences. We can not explore every possible combination, since the lack of semantic information in the trees prohibits us from assessing the quality of the resulting sentences. Instead, we select a combination already present in the input sentences as a basis, and transform it into a fused sentence by removing extraneous information and augmenting the fused sentence with information from other sentences.

The advantage of this strategy is that when the initial sentence is semantically correct and the applied transformations preserve semantic correctness, the resulting sentence is a semantically correct one.

The fusion tree computation consists of three steps: selection of the *basis tree*, augmentation of the tree with alternative verbalizations, and pruning of the extraneous subtrees. The selection of the basis tree is guided by the number of intersection subtrees it includes; in the best case, it contains all such subtrees. The basis tree can be viewed as the centroid of the input sentences — a sentence which is the most similar to the other sentences in the input. Using the similarity function described in section 4.4.1, we identify a centroid by computing for each sentence the average similarity score between the sentence and the rest of the input sentences, and then selecting a sentence with a maximal score.

Next, we augment the basis tree with information present in the other input sentences. More specifically, we add alternative verbalizations for the nodes in the basis tree and the intersection subtrees which are not part of the basis tree. The alternative verbalizations are readily available from the pairwise alignments of the basis tree with other trees in the input computed in the previous section. For each node of the basis tree we record all verbalizations from the nodes of the other input trees aligned to a given node. A verbalization can be a single word, or it can be a phrase, if a node represents a noun compound. An example of a fusion tree, augmented with alternative verbalizations, is given in Figure 4.5. Even after this augmentation, the fusion tree may not include all of the intersection subtrees. The main difficulty in subtree insertion is finding its right placement, which is often determined by various sources of knowledge: syntactic, semantic and idiosyncratic.

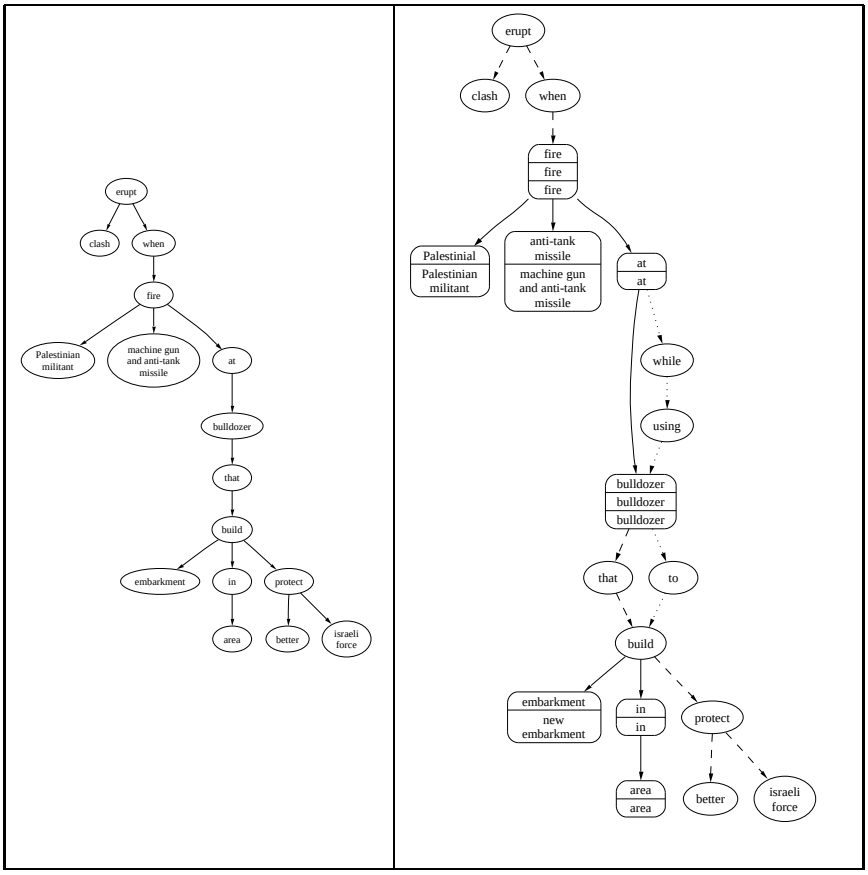

Figure 4.5: A basis tree before and after the augmentation

Finally, subtrees which are not part of the intersection are pruned off the basis tree. However, removing all such subtrees trees may result in an ungrammatical or semantically flawed sentence; for example, we might create a sentence without a subject. This overpruning may happen if either the input to the fusion algorithm is noisy, or the alignment failed to identify the similarity between some subtrees. Therefore, we perform more conservative pruning, deleting self-contained components which can be removed without leaving non-grammatical sentences. As previously observed in the literature (Jing and McKeown, 2000), such components include a

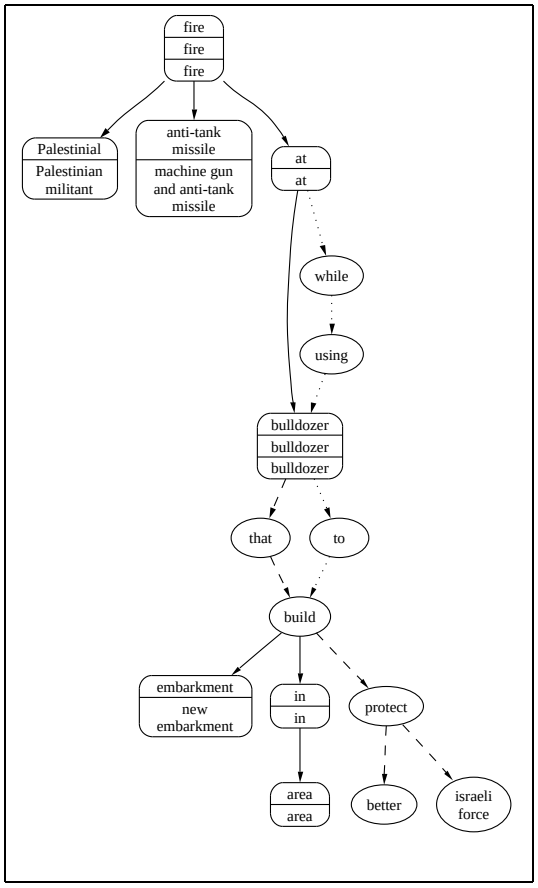

Figure 4.6: Pruned basis tree

clause in the clause conjunction, relative clauses, and some elements within a clause (such as adverbs and propositions). Once these subtrees are removed, the fusion tree construction is completed.

4.4.3 Generation

Finally, it remains to linearize a fusion tree into a sentence. Sentence generation includes selection of a tree traversal order and lexical choice among available alternatives. We don't have to consider all the possible traversals, since the number of

valid traversals is limited by ordering constraints encoded in the fusion tree. Since these constraints are inherited from a basis tree, they do not impose restrictions on the nodes inserted from other trees in the input. Therefore, the algorithm still has to choose among alternative verbalizations and it also has to select an appropriate order from among different orders of the inserted trees.

While the selection of words and phrases which appear in the basis tree is a safe choice, enriching the fusion sentence with alternative verbalizations has several benefits. For example, when the basis tree contains a noun phrase with anaphoric expressions (e.g. "*his visit*") and one of the alternative verbalizations is anaphora-free, substitution of the latter for the anaphoric expression increases the clarity of the produced sentence. In applications such as summarization where the length of the produced sentence is a factor, a shorter alternative is desirable. Besides cases where the substitution is preferable but not mandatory, there are cases where it is a required step for generation of a fluent sentence. As a result of subtree insertions and deletions, the words used in the basis tree may not be a good choice after the transformations. For example, if the phrase "*our correspondent*" is removed from the sentence "*Sharon told our correspondent that the elections were delayed ...*", a replacement of the verb "*told*" with "*said*" yields a more readable sentence.

In addition to lexical choice, there is another question concerning the inserted trees: even though their ordering is partially constrained by their original sentences, they still may be ordered in several possible ways. While the ordering of many sentence constituents is determined by their syntactic roles, some constituents, such as circumstantials, are free to move (Elhadad et al., 2001). In a typical language generation system, placement of such clauses is guided by their semantic type and

frequently is idiosyncratic.

Linearization of the fusion sentence involves the selection of the best phrasing as well as the determination of optimal ordering. Since we do not have sufficient semantic information to perform such selection, our algorithm is driven by corpus-derived knowledge. We generate all possible sentences[2] from the valid traversals of the fusion tree, and score their likelihood according to statistics derived from a corpus. This approach, originally proposed by (Langkilde and Knight, 1998), is a standard method in statistical generation. We trained a 3-gram model over 60M of news articles using the CMU-Cambridge Statistical Language Modeling toolkit (second version). The sentence with the lowest length-normalized entropy is selected as the verbalization of the fusion tree. Figure 4.7 shows several verbalizations produced by our algorithm from the central tree in Figure 4.6.

Sentence	Entropy
Palestinians fired an anti-tank missile at a bulldozer.	4.25
Palestinian militants fired machine-guns and anti-tank missiles at a bulldozer.	5.86
Palestinian militants fired machine-guns and anti-tank missiles at a bulldozer that was building an embankment in the area.	6.22
Palestinians fired anti-tank missiles at while using a bulldozer.	7.04
Palestinians fired anti-tank missile at a bulldozer to build a new embankment in the area.	5.46

Figure 4.7: Alternative linearizations of the fusion tree with the corresponding entropy values

[2]In practice, we sample only n ($n = 20$) paths for efficiency reasons.

4.5 Sentence Fusion Evaluation

We aim to evaluate the sentence fusion algorithm in terms of content selection and the grammaticality of the produced sentences. Traditionally (Knight and Marcu, 2000; DUC, 2002), these dimensions are evaluated separately to avoid confusion in judgment. Evaluating content selection is more problematic than evaluating grammaticality, due to the degree of subjectivity inherent in this judgment. However, unlike in a generic summarization application where the information is selected according to a underspecified salience criteria, the criteria for content selection in information fusion are more concrete. Therefore, comparison with an "ideal" output is a valid method to evaluate the quality of the output. In such an evaluation, a fusion sentence generated by a human is compared to one generated by the system. Among many possible ways to perform such a comparison, we chose to follow the DUC[3] evaluation procedure: the judge assesses overlap on the clause level between the human-created reference sentence and the system output, using the ratings "Full overlap" (2), "Partial overlap" (1) and "No overlap" (0). From the overlap data, we compute recall and precision. The grammaticality is rated in three categories: "Grammatical"(2), "Partially Grammatical"(1), and "Not Grammatical"(0).

In addition to the system generated sentence, we also included in the evaluation a fusion sentence generated by another human and two baselines. The first baseline is the shortest sentence among the theme sentences, which is obviously grammatical, and also it has a good chance of being representative of common topics conveyed in the input. The second baseline is produced by a simplification of our algorithm, where

[3]DUC (Document Understanding Conference) is a community-based evaluation of summarization systems organized by DARPA. Chapter 6 gives a background information on DUC.

paraphrase information is omitted during the alignment process. This baseline is included to capture the contribution of paraphrasing information to the performance of the fusion algorithm. The judge is given a reference sentence along with a system generated sentence, a sentence produced by another human, and two baselines. We scrambled the order of the outputs across test cases.

To evaluate our sentence fusion algorithm, we selected 75 sets of related sentences from material collected by Newsblaster. Each set varied from two to eight sentences, with 3.63 sentences on average. As we mentioned above, Simfinder does not always produce accurate themes[4]. Therefore, the human creating reference sentences had an option not to generate any if the theme sentences had little in common. Given 75 sets, the human judge generated 61 fusion sentences. An example of a theme for which no sentence was generated is shown in Figure 4.8. We evaluate the system performance only on these 61 sets. 24 out of 61 sentences produced by the algorithm combined phrases from several sentences, while the rest of the sentences were either subsequences of the original sentences (17) or were fully extracted (20)[5].

Figure 4.9 shows two sentences from the test corpus, along with input sentences.The examples are chosen so as to reflect good and bad performance cases.

The results in Figure 4.10 show compression rate, precision, and recall for each algorithm. The compression rate of a sentence was computed as the ratio of its output length to the average length of the theme input sentences. The generally close match between the reference sentence and the sentence generated by another

[4]To mitigate the effects of Simfinder noise in MultiGen, we induced a similarity threshold on input trees — trees which are not similar to the basis tree are not used in the fusion process. Typically, the output of the noisy theme is a basis sentence itself.

[5]We observed that sentence fusion reduces to sentence extraction when an input theme contains almost identical sentences.

The shares have fallen 60 percent this year.
They said Qwest was forcing them to exchange their bonds at a fraction of face value — between 52.5 percent and 82.5 percent, depending on the bond — or else fall lower in the pecking order for repayment in case Qwest went broke.
Qwest had offered to exchange up to $12.9 billion of the old bonds, which carried interest rates between 5.875 percent and 7.9 percent.
The new debt carries rates between 13 percent and 14 percent.
Their yield fell to about 15.22 percent from 15.98 percent.

Figure 4.8: An example of noisy Simfinder output.

human validates our hypothesis that an "ideal"-sentence based evaluation for content selection makes sense for the fusion task, because different humans produce very similar output for a given input.

These results confirm our hypothesis about the importance of paraphrasing information for the fusion process. Omission of paraphrases (baseline 2) causes a 9% drop in recall due to its inability to match equivalent phrases with different wording. The performance of the second baseline demonstrates that the shortest sentence is an inadequate substitution for fusion in terms of content selection as well as compression rate.

To further investigate the generation capacity of the information fusion algorithm, we conducted a separate evaluation concentrated on the well-formedness of the produced sentences. We randomly selected 44 summaries (182 sentences) from the Newsblaster run. On average, each sentence fused fragments from 1.8 theme sentences. The human judge identified 150 (82%) sentences as grammatical. Only 23 (53%) summaries did not contain any ungrammatical sentences.

The majority of mistakes originated from linearization component. Mistakes

#1	The forest is about 70 miles west of Portland.
#2	Their bodies were found Saturday in a remote part of Tillamook State Forest, about 40 miles west of Portland.
#3	Elk hunters found their bodies Saturday in the Tillamook State Forest, about 60 miles west of the family's hometown of Portland.
#4	The area where the bodies were found is in a mountainous forest about 70 miles west of Portland.
Generated	The bodies$_4$ were found$_2$ Saturday$_2$ in$_3$ the Tillamook$_3$ State$_3$ Forest$_3$ west$_2$ of$_2$ Portland$_2$.
#1	Four people including an Islamic cleric have been detained in Pakistan after a fatal attack on a church on Christmas Day.
#2	Police detained six people on Thursday following a grenade attack on a church that killed three girls and wounded 13 people on Christmas Day.
#3	A March 17 grenade attack on a Protestant church in Islamabad killed five people, including a U.S. Embassy employee and her 17 - year - old daughter.
#4	On March 17, a grenade attack on a Protestant church in Islamabad killed five people, including a U.S. Embassy employee and her 17 - year - old daughter.
Generated	On$_4$ March$_4$ 17$_4$ a grenade$_4$ attack$_4$ on$_4$ a protestant$_4$ church$_4$ in$_4$ Islamabad$_4$ killed$_4$ six$_2$ people$_2$.

Figure 4.9: Examples from the test set.

in these category include incorrect selection of determiners (2), wrong realization of negation constructions (8) and tense (7). Other mistakes result from the overcut during the pruning process (5) and the suboptimal lexical choice for the nodes of the augmented central tree (6). Figure 4.11 shows examples of mistakes in the fusion sentences.

Frequency	Compression	Precision	Recall	Grammaticality
System	84%	68%	73%	2.2
Baseline 1	74%	55%	61%	3
Baseline 2	80%	68%	64%	2.2
Human	63%	100%	92%	3

Figure 4.10: Evaluation results for human crafted fusion sentence, our system output, the shortest sentence in the theme (baseline 1) and a simplified version of our algorithm without paraphrasing information (baseline 2).

The coalition to have play a central role.
Earlier Thursday about 15 members of an elite police unit that was close to former President Slobodan Milosevic were arrested on suspicion they helped organize the killing along with the police unit.
North Korea earlier hinted test but Japanese officials said there was no indication it was prepared.

Figure 4.11: Examples of mistakes in generated sentences.

4.6 Conclusions and Future Work

As the evaluation shows, our algorithm outperforms the shortest sentence baseline in terms of content selection, without a significant drop in grammaticality. We also showed that augmenting the fusion process with paraphrasing knowledge improves the output by both measures. However, there is still a gap between our system and human performance.

Our manual analysis of the output revealed that some "mistakes" made by the algorithm can be attributed to problems with alignment. Our assumption about one-to-one mapping does not always hold; namely, two trees conveying the same meaning may not be decomposable into the node level mappings which our algorithm aims to compute. For example, the mapping between the sentences in Figure 4.12 expressed by the rule "*X denied claims by Y*" $\leftrightarrow$ "*X said that Y's claim was untrue*" cannot

be decomposed into smaller matching units.

Syria denied claims by Israeli Prime Minister Ariel Sharon...
The Syrian spokesman said that Sharon 's claim was untrue...

Figure 4.12: A pair of sentences which can not be fully decomposed.

Besides the high computational cost associated with many-to-many mapping, such a mapping requires a dictionary of clause level paraphrases which can not be computed with existing methods. We partially account for this phenomenon by collapsing noun phrases into composite nodes and comparing them against atomic and other composite nodes, allowing one-to-one or many-to-many matches for these nodes, but of course there is room for significant improvement in addressing this problem. Another limitation of our algorithm concerns the weights guiding the alignment process. We did our best to fit our development data by hand; a more systematic approach to parameter estimation may produce better results. In previous work, the estimation procedures relied on a large human-annotated corpus, which was not available for our task.

On the generation side, we can improve the quality of fusion output with more powerful language models. Currently, we restrict the types of possible insertions and deletions due to the weak capability of the language model to distinguish whether a sentence is a well-formed one or not. More than once, we observed that language models selected ungrammatical sentences, assigning a lower score to a better sentence. For example, the fifth sentence in Figure 4.7 is not a well-formed sentence; however, our language model gave it a better score than to its well-formed alternatives (the second and the third sentences). Therefore, we eliminated "high-risk" transformations,

which reduces the generative power of the algorithm. Recent research (Daume et al., 2002) has shown that syntax-based language models are more suitable for language generation; the study of such models is a promising direction to explore.

One of the limitations specific to our implementation of the sentence fusion algorithm is an inability to properly place punctuation[6]. We were unable to develop a set of rules which works in most cases. Punctuation placement is determined by a variety of features; considering all possible interactions of these features is hard. We believe a corpus-based approach is a promising approach to this problem.

The algorithm for fusion does not match the level of human performance; a pertinent question is whether its performance is sufficiently good to be used in applications, in particular in summarization systems. The overall evaluation of this system presented in Chapter 6 gives an additional confirmation that sentence fusion produces good results.

In the next chapter, we focus on ordering generated sentences into a summary.

[6]In our grammaticality evaluation (following the DUC procedure), the judge was asked to ignore punctuation.

Chapter 5

Strategies for Sentence Ordering in Multi-Document Summarization

In the previous chapter, we described a method for generation of summary sentences. This chapter investigates methods for ordering generated sentences into a coherent text [1].

5.1 Introduction

One issue that has received little attention in multi-document summarization is how to organize the selected information so that the output summary is coherent. Once all the relevant pieces of information have been selected across the input documents, the summarizer has to decide in which order to present them so that the whole text makes sense. In single document summarization, one possible ordering of the extracted information is provided by the input document itself. However, (Jing, 1998)

[1] This chapter is based on the JAIR article(Barzilay, Elhadad, and McKeown, 2002)

observed that, in single document summaries written by professional summarizers, extracted sentences do not always retain their precedence orders in the summary. Moreover, in the case of multiple input documents, this does not provide a useful solution: information may be drawn from different documents and therefore, no single document can provide an ordering. Furthermore, the order between two pieces of information can change significantly from one document to another.

In this chapter, we provide a corpus based methodology for studying ordering. Our goal was to develop a good ordering strategy in the context of multi-document summarization targeted for the news genre. The first question we addressed is the importance of ordering. We conducted experiments which show that ordering significantly affects the reader's comprehension of a text. Our experiments also show that although there is no single ideal ordering of information, ordering is not an unconstrained problem; the number of good orderings for a given text is limited. The second question addressed was the analysis and use of data to infer a strategy for ordering. Existing corpus based methods, such as supervised learning, are not easily applicable to our problem in part because of lack of training data. Given that there are multiple possible orderings, a corpus providing one ordering for each set of information does not allow us to differentiate between sentences which must be in a given order and sentences which happen to be together. This led us to develop a corpus of data sets, each of which contains multiple acceptable orderings of a single text. Such a corpus is expensive to construct and therefore, does not provide enough data for pure statistical approaches. Instead, we used a hybrid corpus analysis strategy that first automatically identifies commonalities across orderings. Manual analysis of the resulting clusters led to the identification of constraints on ordering. Finally, we

evaluated plausible ordering strategies by asking humans to judge the results.

Our set of experiments together suggests an ordering algorithm that integrates constraints from an approximation of the temporal sequence of the underlying events and relatedness between content elements. Our evaluation of plausible strategies measures the usefulness of a Chronological Ordering algorithm used in previous summarization systems (McKeown et al., 1999; Lin and Hovy, 2001) as well as an alternative, original strategy, Majority Ordering. Our evaluation shows that the two ordering algorithms alone do not yield satisfactory results. The first, Majority Ordering, is critically linked to the level of similarity of information ordering across the input texts. When input texts have different orderings, however, the algorithm produces unpredictable and unacceptable results. The second, Chronological Ordering produces good results when the information is event-based, and therefore, is temporally sequenced. When texts do not refer to events, but describe states or properties, this algorithm falls short.

Our automatic analysis reveals that topical relatedness is an important constraint; groups of related sentences tend to appear together. Our algorithm combines Chronological Ordering with constraints from topical relatedness. Evaluation shows that the augmented algorithm significantly outperforms either of the simpler methods alone. This strategy can be characterized as bottom-up since final ordering of the text emerges from how the data groups together, whether by related content or by chronological sequence. This contrasts with top-down strategies developed in traditional generation such as RST (Moore and Paris, 1993; Hovy, 1993), schemas (McKeown, 1985) or plans (Dale, 1992) which impose an external, rhetorically motivated ordering on the data.

In the following sections, we first show that the way information is ordered in a summary can critically affect its overall quality. We next describe the two naive ordering algorithms and evaluate them, followed by a study of multiple orderings produced by humans. This allows us to determine how to improve the Chronological Ordering algorithm using cohesion as an additional constraint. The last section describes the augmented algorithm along with its evaluation.

5.2 Impact of Ordering on the Overall Quality of a Summary

Even though the problem of ordering information for multi-document summarization has received relatively little attention, we hypothesize that good ordering is crucial to produce summaries of quality. The consensus architecture of the state of the art summarizers consists of a content selection module in which the salient information is extracted and a regeneration module in which the information is reformulated into a fluent text. Ideally, the regeneration component contains devices that perform surface repairs on the text by doing anaphora resolution, introducing cohesion markers or choosing the appropriate lexical paraphrases. Our claim in this chapter is that the multi-document summarization architecture needs an explicit ordering component. If two pieces of information extracted by the content selection phase end up together but should not, in fact, be next one to another, surface devices will not repair the impaired flow of information in the summary. An ordering strategy would help avoid this situation.

It is clear that ordering cannot improve the output of earlier stages of a sum-

marizer, among them content selection[2]; however, finding an acceptable ordering can enhance user comprehension of the summary and, therefore, its overall quality. Of course, surface devices are still needed to smooth the output summary, but this is beyond the scope of our work (but see (Schiffman, Nenkova, and McKeown, 2002)). In this section we show that the quality of ordering has a direct effect on user comprehension of the summary. To verify our hypothesis, we performed an experiment, measuring the impact of ordering on the user's comprehension of summaries.

We selected ten summaries produced by the Columbia Summarization system (McKeown et al., 2001a). It is composed of a router and two underlying summarizers — MultiGen and DEMS. Depending on the type of input articles to be summarized, the router selects the appropriate summarizer. We evaluated this system through the Document Understanding Conference 2001 (DUC)[3] evaluation, where summaries produced by several systems were graded by human judges according to different criteria, among them how well the information contained in the summary is ordered. To actually identify a possible impact of ordering on comprehension, we selected only summaries where humans judged the ordering as poor.[4] For each summary, we manually reordered the sentences generated by the summarizer, using the input articles as a reference. When doing so, we did not change the content — all the sentences in the reordered summaries were the same ones as in the originally produced summaries. This process yields ten additional reordered summaries and thus, overall our collection contains twenty summaries.

[2]No information is added or deleted once the content selection is performed.

[3]`http://www-nlpir.nist.gov/projects/duc/`

[4]The selected summaries were produced by the DEMS system. We didn't select any summary produced by MULTIGEN because it implemented our ordering algorithm at the time. DEMS on the other hand, had no specific ordering strategy implemented and thus provided us with the appropriate type of data.

Two subjects other than the experiment organizers participated in this experiment. Each summary was read by one participant without having access to the input articles. We distributed the summaries among the judges so that none of them read both an original summary and its reordering. They were asked to grade how well the summary could be understood, using the ratings "Incomprehensible," "Somewhat comprehensible" or "Comprehensible".

The results[5] are shown in Figure 5.1. Seven original summaries were considered incomprehensible by their judge, two were somewhat comprehensible, and only one original summary was fully comprehensible. The reordered summaries obtained better grades overall — five summaries were fully comprehensible, two were somewhat comprehensible, while three remained incomprehensible. To assess the statistical significance of our results, we applied the Fisher exact test to our data set, conflating "Incomprehensible" and "Somewhat comprehensible" summaries into one category to obtain a 2x2 table. This test is adapted to our case because of the reduced size of our data set. We obtained a p-value of 0.07 (Siegel and Castellan, 1988), which means that if reordering is not, in general, helpful, there is only a 7% chance that doing reordering anyway would produce a result this different in quality from the original ordering. This experiment indicates that a good ordering can improve the overall comprehensibility of a summary.

In the case of some low-scoring summaries, it is clear that poor ordering is the likely culprit. For instance, readers can easily identify that grouping the two following sentences is an unsuitable choice and could be misleading. *"Miss Taylor's health problems started with a fall from a horse when she was 13 and filming the*

[5]The set names are the ones used in the DUC evaluation.

Summary set	Original	Reordered
d13	Incomprehensible	Incomprehensible
d19	Somewhat comprehensible	Comprehensible
d24	Incomprehensible	Comprehensible
d31	Somewhat comprehensible	Comprehensible
d32	Incomprehensible	Somewhat comprehensible
d39	Incomprehensible	Incomprehensible
d45	Incomprehensible	Incomprehensible
d50	Incomprehensible	Comprehensible
d54	Incomprehensible	Somewhat comprehensible
d56	Comprehensible	Comprehensible

Figure 5.1: Impact of ordering on the user comprehension of summaries.

movie National Velvet. The recovery of Elizabeth Taylor, near death two weeks ago with viral pneumonia, was complicated by a yeast infection, her doctors said Friday." But in other cases, when information in a summary is poorly ordered and readers cannot make sense of the text, we observed through interviews with the readers that they tend to blame it on content selection rather than on ordering, even if the content is not the issue. Thus, the issue of ordering is not isolated; it can affect the overall quality of a summary.

In the following section, we describe different strategies for ordering the output sentences to obtain a quality summary.

5.3 Naive Ordering Algorithms Are Not Sufficient

When producing a summary, any multi-document summarization system has to choose in which order to present the output sentences. In this section, we describe two algorithms for ordering sentences suitable for multi-document summarization in the

news genre. The first algorithm, Majority Ordering (MO), relies only on the original orders of sentences in the input documents. The second one, Chronological Ordering (CO), uses time-related features to order sentences. This strategy was originally implemented in MULTIGEN and followed by other summarization systems (Radev, Jing, and Budzikowska, 2000; Lin and Hovy, 2001). In the MULTIGEN framework, ordering sentences is equivalent to ordering themes, and we describe the algorithms in terms of themes. This makes sense because, ultimately, the summary will be composed of a sequence of sentences, each one constructed from the information in one theme. Our evaluation shows that these methods alone do not provide an adequate strategy for ordering.

5.3.1 Majority Ordering

The Algorithm

In single document summarization, the order of sentences in the output summary is typically determined by their order in the input text. This strategy can be adapted to multi-document summarization. Consider two themes, Th_1 and Th_2; if sentences from Th_1 precede sentences from Th_2 in all input texts, then presenting Th_1 before Th_2 is likely to be an acceptable order. To use the majority ordering algorithm when the order between sentences from Th_1 and Th_2 varies from one text to another, we must augment the strategy. One way to define the order between Th_1 and Th_2 is to adopt the order occurring in the majority of the texts where Th_1 and Th_2 occur. This strategy defines a pairwise order between themes. However, this pairwise relation is not necessarily transitive. For example, given the themes Th_1, Th_2 and Th_3 and the following situation: Th_1 precedes Th_2 in a text, Th_2 precedes Th_3 in the same text or

in another text, and Th_3 precedes Th_1 in yet another text; there is a conflict between the orders (Th_1, Th_2, Th_3) and (Th_3, Th_1). Since transitivity is a necessary condition for a relation to be called an order, this relation does not form an order.

We, therefore, have to expand this pairwise relation to provide a total order. In other words, we have to find a linear ordering between themes which maximizes the agreement between the orderings provided by the input texts. For each pair of themes, Th_i and Th_j, we keep two counts; $C_{i,j}$ and $C_{j,i}$; $C_{i,j}$ is the number of input texts in which sentences from Th_i occur before sentences from Th_j, and $C_{j,i}$ is the same for the opposite order. The weight of a linear order $(Th_{i_1}, \ldots, Th_{i_k})$ is defined as the sum of the counts for every pair C_{i_l,i_m}, such that $i_l \leq i_m$ and $l, m \in \{1 \ldots k\}$. Stating this problem in terms of a directed graph where nodes are themes, and a vertex from Th_i to Th_j has the weight $C_{i,j}$. We call such a graph a precedence graph. We are looking for a path with maximal weight which traverses each node exactly once (see Figure 5.2).

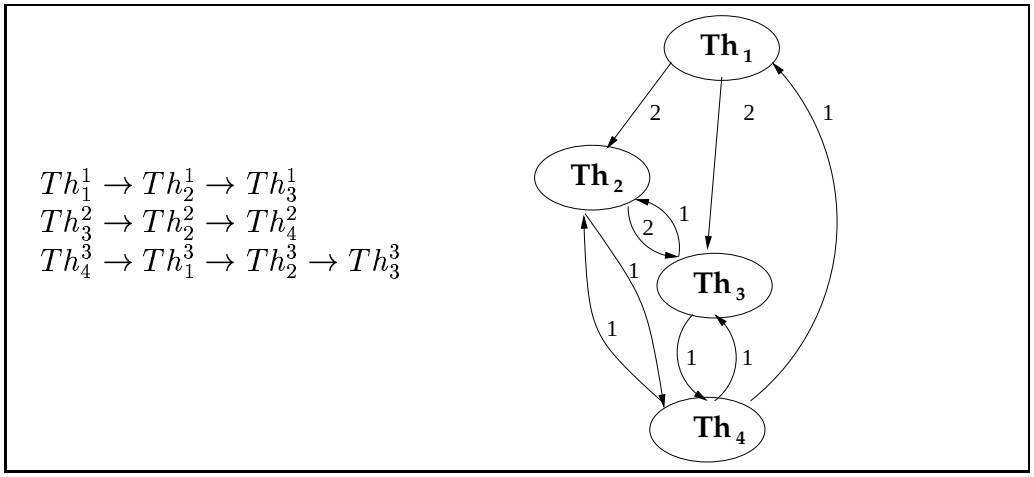

Figure 5.2: Three input theme orderings and their corresponding precedence graph. Th_i^j is the sentence part of the theme Th_i in the input ordering j.

The problem of finding a path with maximal weight has been addressed by (Cohen, Schapire, and Singer, 1999) in the task of learning orderings. They adopt a two-stage approach. In the first stage, given a training corpus of ordered instances and a set of features describing them, a binary preference function is learned. In the second stage, new instances are ordered so that agreement with the learned preference function is maximized. To do so, (Cohen, Schapire, and Singer, 1999) represent the preference function as a directed, weighted graph. Our precedence graph can be seen as such a graph where the preference function between the nodes Th_i and Th_j is $C_{i,j}$. The orderings from the input articles provide us directly with a preference function and, therefore, we do not need to learn it.

Unfortunately this problem is NP-complete; (Cohen, Schapire, and Singer, 1999) prove it by reducing from CYCLIC-ORDERING (Galil and Megiddo, 1977). However, using a modified version of topological sort provides us with an approximate solution. For each node, we assign a weight equal to the sum of the weights of its outgoing edges minus the sum of the weights of its incoming edges. We first pick up the node with maximum weight, ordering it ahead of the other nodes, delete it and its outgoing edges from the precedence graph and update properly the weights of the remaining nodes in the graph. We then iterate through the nodes until the graph is empty. (Cohen, Schapire, and Singer, 1999) show that this algorithm produces a tight approximation of the optimal solution. Currently MULTIGEN uses an implementation of this algorithm for its ordering component.

Figures 5.3 and 5.4 show examples of produced summaries. One feature of this strategy is that it can produce several orderings with the same weight. This happens when there is a tie between two opposite orderings. In this situation, this strategy

does not provide enough constraints to determine one optimal ordering; an ordering is chosen randomly among the orders with maximal weight.

The man accused of firebombing two Manhattan subways in 1994 was convicted Thursday after the jury rejected the notion that the drug Prozac led him to commit the crimes.
He was found guilty of two counts of attempted murder, 14 counts of first-degree assault and two counts of criminal possession of a weapon.
In December 1994, Leary ignited firebombs on two Manhattan subway trains. The second blast injured 50 people – 16 seriously, including Leary.
Leary wanted to extort money from the Transit Authority.
The defense argued that Leary was not responsible for his actions because of "toxic psychosis" caused by the Prozac.

Figure 5.3: A summary produced using the Majority Ordering algorithm, graded as Good.

Hemingway, 69, died of natural causes in a Miami jail after being arrested for indecent exposure.
A book he wrote about his father, "Papa: A Personal Memoir," was published in 1976.
He was picked up last Wednesday after walking naked in Miami.
"He had a difficult life."
A transvestite who later had a sex-change operation, he suffered bouts of drinking, depression and drifting, according to acquaintances.
"It's not easy to be the son of a great man," Scott Donaldson, told Reuters.
At the time of his death, he lived in the Coconut Grove district where he was well-known to its Bohemian crowd.
He had been due to appear in court later that day on charges of indecent exposure and resisting arrest.
He sometimes went by the name of Gloria and wore women's clothes.
The cause of death was hypertension and cardiovascular disease.
Taken to the Miami-Dade Women's Detention Center, he was found dead in his cell early on Monday, spokeswoman Janelle Hall said.
He was booked into the women's jail because he had a sex-change operation, Hall added.

Figure 5.4: A summary produced using the Majority Ordering algorithm, graded as Poor.

Evaluation

We asked three human judges (not including ourselves) to classify the quality of the order of information in 25 summaries produced using the MO algorithm into

three categories— Poor, Fair and Good. We use an operational definition of a Poor summary as a text whose readability would be significantly improved by reordering its sentences. A Fair summary is a text which makes sense, but reordering of some sentences can yield a better readability. Finally, a summary which cannot be further improved by any sentence reordering is considered a Good summary.

The judges were asked to grade the summaries taking into account only the order in which the information is presented. To help them focus on this aspect of the texts, we resolved dangling references beforehand. Figure 5.12 shows the grades assigned to the summaries — three summaries were graded as Poor, 14 were graded as Fair, and eight were graded as Good. We are showing here the majority grade that is selected by at least two judges. This was made possible because in our experiments, judges had strong agreement; they never gave three different grades to a summary.

The MO algorithm produces a small number of Good summaries, but most of the summaries were graded as Fair. For instance, the summary graded Good shown in Figure 5.3 orders the information in a natural way; the text starts with a sentence summary of the event, then the outcome of the trial is given, a reminder of the facts that caused the trial and a possible explanation of the facts. Looking at the Good summaries produced by MO, we found that it performs well when the input articles follow the same order when presenting the information. In other words, the algorithm produces a good ordering if the input articles' orderings have high agreement.

On the other hand, when analyzing Poor summaries, we observed that the input texts have very different orderings. By trying to maximize the agreement of the input texts' orderings, MO produces a new ordering that does not occur in any input text. The ordering is, therefore, not guaranteed to be acceptable. An example

of a new produced ordering is given in Figure 5.4. The summary would be more readable if several sentences were moved around. An example of a better ordering is given in Figure 5.5. In this summary, the three sentences related to the fact that the subject had a sex-change operation are grouped together, while in the one produced by the majority ordering algorithm, they are scattered throughout the summary.

Hemingway, 69, died of natural causes in a Miami jail after being arrested for indecent exposure.
The cause of death was hypertension and cardiovascular disease.
He was picked up last Wednesday after walking naked in Miami.
He had been due to appear in court later that day on charges of indecent exposure and resisting arrest.
Taken to the Miami-Dade Women's Detention Center, he was found dead in his cell early on Monday, spokeswoman Janelle Hall said.
He was booked into the women's jail because he had a sex-change operation, Hall added.
A transvestite who later had a sex-change operation, he suffered bouts of drinking, depression and drifting, according to acquaintances.
He sometimes went by the name of Gloria and wore women's clothes.
"He had a difficult life."
"It's not easy to be the son of a great man," Scott Donaldson, told Reuters.
At the time of his death, he lived in the Coconut Grove district where he was well-known to its Bohemian crowd.
A book he wrote about his father, "Papa: A Personal Memoir," was published in 1976.

Figure 5.5: One possible better ordering for the summary graded as Poor.

This algorithm can be used to order sentences accurately if we are certain that the input texts follow similar organizations. This assumption may hold in limited domains where documents have a fixed organization of the information. However, in our case, the input texts we are processing do not have such regularities. Looking at the daily statistics of Newsblaster[6]), which collects clusters of related articles to be synthesized into one summary, we notice that the typical cluster size is seven. But every day there are several clusters which contain more than 20 and up to 70 articles

[6]See Chapter 4 for system description

to be summarized into single summaries[7]. With such a big number of input articles, we cannot assume that they will all have similar ordering of the information. MO's performance critically depends on the agreement of orderings in the input texts; we, therefore, need an ordering strategy which can fit any input data. From here on, we will focus only on the Chronological Ordering algorithm and techniques to improve it.

5.3.2 Chronological Ordering

The Algorithm

Multi-document summarization of news typically deals with articles published on different dates, and articles themselves cover events occurring over a wide range of time. Using chronological order in the summary to describe the main events helps the user understand what has happened. It seems like a natural and appropriate strategy. As mentioned earlier, in our framework, we are ordering themes; using this strategy, we, therefore, need to assign a date to themes. To identify the date an event occurred requires a detailed interpretation of temporal references in articles. While there have been recent developments in disambiguating temporal expressions and event ordering (Wiebe et al., 1998; Mani and Wilson, 2000; Filatova and Hovy, 2001), correlating events with the date on which they occurred is a hard task. In our case, we approximate the theme time by its first publication time; that is, the first time the theme has been reported in our set of input articles (see Figure 5.6). It is an acceptable approximation for news events; the first publication time of an event usually corresponds to its occurrence in real life. For instance, in a terrorist attack

[7] These giant clusters correspond to the "hot topics" of the day in the news.

story, the theme conveying the attack itself will have a date previous to the date of the theme describing a trial following the attack.

Theme 5	
Oct 5, 11:35am	Hours after the crash, U.S. officials said that the tragedy had been caused by an S-200 missile fired by Ukraine during military exercises on the Crimean Peninsula.
Oct 6, 6:13am	U.S. officials said immediately after the crash that they had evidence the passenger jet was hit by a Ukrainian missile.
Oct 5, 10:20am	But U.S. officials said that the crash had been caused by an S-200 missile fired mistakenly by Ukrainian forces during military exercises on the Crimean Peninsula.

Figure 5.6: A theme with its corresponding sentences. The time theme is shown underlined; it is the earliest publication time of the sentences.

Articles released by news agencies are marked with a publication time, consisting of a date and a time with two fields (hour and minutes). Articles from the same news agency are thus guaranteed to have different publication times. This is also quite likely for articles coming from different news agencies. During the development of MULTIGEN, we processed hundreds of articles, and we never encountered two articles with the same publication time. Thus, empirically the publication time serves as a unique identifier over articles. As a result, when two themes have the same publication time, it means that they both are reported for the first time in the same article.

Our Chronological Ordering (CO) algorithm takes as input a set of themes and orders them chronologically whenever possible. Each theme is assigned a date corresponding to its first publication. To do so, we select for each theme the sentence that has the earliest publication time. We call it the time stamp sentence and assign

its publication time as the time stamp of the theme. This establishes a partial order over the themes. When two themes have the same date (that is, they are reported for the first time in the same article) we sort them according to their order of presentation in this article. This results in a total order over the input themes. Figures 5.7 and 5.8 show examples of summaries produced using CO.

One of four people accused along with former Pakistani Prime Minister Nawaz Sharif has agreed to testify against him in a case involving possible hijacking and kidnapping charges, a prosecutor said Wednesday.
Raja Quereshi, the attorney general, said that the former Civil Aviation Authority chairman has already given a statement to police.
Sharif's lawyer dismissed the news when speaking to reporters after Sharif made an appearance before a judicial magistrate to hear witnesses give statements against him. Sharif has said he is innocent.
The allegations stem from an alleged attempt to divert a plane bringing army chief General Pervez Musharraf to Karachi from Sri Lanka on October 12.

Figure 5.7: A summary produced using the Chronological Ordering algorithm graded as Good.

Thousands of people have attended a ceremony in Nairobi commemorating the first anniversary of the deadly bombings attacks against U.S. Embassies in Kenya and Tanzania.
Saudi dissident Osama bin Laden, accused of masterminding the attacks, and nine others are still at large.
President Clinton said, "The intended victims of this vicious crime stood for everything that is right about our country and the world".
U.S. federal prosecutors have charged 17 people in the bombings.
Albright said that the mourning continues.
Kenyans are observing a national day of mourning in honor of the 215 people who died there.

Figure 5.8: A summary produced using the Chronological Ordering algorithm graded as Poor.

Evaluation

Following the same methodology we used for the MO algorithm evaluation, we asked three human judges (not including ourselves) to grade 25 summaries generated by the system using the CO algorithm applied to the same collection of input texts. The results are shown in Figure 5.12: ten summaries were graded as Poor, eight were graded as Fair and seven were graded as Good.

Our first suspicion was that our approximation deviates too much from the real chronological order of events and, therefore, lowers the quality of sentence ordering. To verify this hypothesis, we identified sentences that broke the original chronological order and restored the ordering manually. Interestingly, the displaced sentences were mainly background information. The evaluation of the modified summaries shows no visible improvement.

When comparing Good (Figure 5.7) and Poor (Figure 5.8) summaries, we notice two phenomena: first, many of the badly placed sentences cannot be ordered based on their temporal occurrence. For instance, in Figure 5.8, the sentence quoting Clinton is not one event in the sequence of events being described, but rather, a reaction to the main events. A tool assigning time stamps would assign to this sentence the date at which Clinton made his statement. This is also true for the sentence reporting Albright's reaction. Assigning a date to a reaction, or more generally to any sentence conveying background information, and placing it into the chronological stream of the main events does not produce a logical ordering. The ordering of these themes is, therefore, not covered by the CO algorithm. Furthermore, some sentences cannot be assigned any time stamp. For instance, the sentence, *"The vast, sparsely inhabited Xinjiang region, largely desert, has many Chinese military and nu-*

clear installations and civilian mining." describes a state rather than an event and, therefore, trying to describe it in temporal terms is invalid. Thus, the ordering cannot be improved at the temporal level.

The second phenomenon we observed is that Poor summaries typically contain abrupt switches of topics and are generally incoherent. For instance, in Figure 5.8, quotes from US officials (third and fifth sentences) are split, and sentences about the mourning (first and sixth sentences) appear too far apart in the summary. Grouping them together would increase the readability of the summary. At this point, we need to find additional constraints to improve the ordering.

5.4 Improving the Ordering: Experiments and Analysis

In the previous section, we showed that using naive ordering algorithms does not produce satisfactory orderings. In this section, we investigate through experiments with humans how to identify patterns of orderings that can improve the algorithm.

5.4.1 Collecting a corpus of multiple orderings

Sentences in a text can be ordered in a number of ways, and the text as a whole will still convey the same meaning. But the majority of possible orders are likely to be unacceptable because they break conventions of information presentation. One way to identify these conventions is to find commonalities among different acceptable orderings of the same information. Extracting regularities in several acceptable orderings can help us specify ordering constraints for a given input type. There is no

naturally occurring existing collection of summaries for multiple documents that we aware of[8]. But even such a collection would not be sufficient since we want to analyze a collection of multiple summaries over the same set of articles. We created our own collection of multiple orderings produced by different humans. Using this collection, we studied common behaviors and mapped them to strategies for ordering.

Our collection of multiple orderings, along with our test corpus is available at `http://www.cs.columbia.edu/~noemie/ordering/`. We collected ten sets of articles for this collection. Each set consisted of two to three news articles reporting the same event. For each set, we manually selected the intersection sentences, simulating MULTIGEN[9]. On average, each set contained 8.8 intersection sentences. The sentences were cleaned of explicit references (for instance, occurrences of "the President" were resolved to "President Clinton") and connectives, so that participants would not use them as clues for ordering. Ten subjects participated in the experiment, and they each built one ordering per set of intersection sentences. Each subject was asked to order the intersection sentences of a set so that they form a readable text. Overall, we obtained 100 orderings, ten alternative orderings per set. Figure 5.9 shows the ten alternative orderings collected for one set.

We first observed that a surprisingly large portion of the orderings are different. Out of the ten sets, only two sets had some identical orderings (in one set, two orderings were identical while in the other set, there were two pairs of identical orderings). This variety in the produced orderings can be interpreted as suggesting that not all the orderings were actually valid or that the task was maybe too hard for the

[8]In a recent attempt, NIST for the DUC conference collected sets of articles to summarize and one summary per set.

[9]We performed a manual simulation to ensure that ideal data was provided to the subjects of the experiments.

Participant 1	D B G I H F C J A E
Participant 2	D G B I C F A J E H
Participant 3	D B I G F J A E H C
Participant 4	D C F G I B J A H E
Participant 5	D G B I H F J A C E
Participant 6	D G I B F C E H J A
Participant 7	D B G I F C H E J A
Participant 8	D B C F G I E H A J
Participant 9	D G I B E H F A J C
Participant 10	D B G I C F A J E H

Figure 5.9: Multiple orderings for one set in our collection. A, B, ..., J stand for sentences. Underlined are automatically identified blocks.

subjects to allow them to produce reasonable orderings. In fact, all the subjects were satisfied with the orderings they produced. Furthermore, we manually went through all the 100 orderings, and all appeared to be valid. In other words, there are many acceptable orderings given one set of sentences. This confirms the intuition that we do not need to look for a single ideal total ordering but rather construct an acceptable one.

Looking at these various orderings, one might also conclude that any ordering would do just as well as any other. One piece of evidence against this statement is that, as shown in section 5.2, some orderings yield incomprehensible texts and thus should be avoided. Furthermore, for a text with n sentences, there are $n!$ possible orderings, but only a small fraction of those are actually valid orderings. One way to validate this claim would be to enumerate all the possible orderings of a single text and evaluate each one of them. This would be doable for very small texts (a text of 5 sentences has 120 possible orderings) but not for texts of a reasonable size. A more feasible way to validate our claim is to get multiple orderings of the same text from a large number of

subjects. We asked subjects to order one text of eight sentences. There is a maximum of 40,320 possible orderings for these sentences. While 50 subjects participated, we only obtained 21 unique orderings, showing that the number of acceptable orderings does not grow as fast as the number of participants. We can conclude that only a small fraction of all possible orderings of the information in a text contains orderings that render a readable text.

5.4.2 Analysis

The several alternative orderings produced for a single summary exhibit commonalities. We noticed that, within the multiple orderings of a set, some sentences always appear together. They do not appear in the same order from one ordering to another, but they share an adjacency relation. From now on, we refer to them as blocks. For each set, we identify blocks by automatically clustering sentences across orderings. We use as a distance metric between two sentences, the average number of sentences that separate them over all orderings. In Figure 5.9, for instance, the distance between sentences D and G is 2. The blocks identified by clustering are: sentences B, D, G and I; sentences A and J; sentences C and F; and sentences E and H.

We observed that all the blocks in the experiment correspond to clusters of topically related sentences. These blocks form units of text dealing with the same subject. In other words, all valid orderings contain blocks of topically related sentences. The notion of grouping topically related sentences is known as cohesion. As defined by (Hasan, 1984), cohesion is a device for "linking together" different parts of the text. Studies show that the level of cohesion has a direct impact on reading comprehension (Halliday and Hasan, 1976). Therefore, good orderings are cohesive;

this is part of what makes the summary readable. Conversely, the evaluation of the CO algorithm showed that the summaries that were judged invalid contain abrupt switches of topic. In other words, orderings that are not cohesive are graded poorly. There is a correlation between the quality of the ordering and cohesion. Incorporating the cohesion constraint into our ordering strategy by opportunistically grouping sentences together would be beneficial. Cohesion is achieved by surface devices, such as repetition of words and coreferences. We describe next how we include cohesion in the CO algorithm based on these surface features.

5.5 The Augmented Algorithm

Disfluencies arise in the output of the CO algorithm when topics are distributed over the whole text, violating cohesion properties (McCoy and Cheng, 1991). A typical scenario is illustrated in Figure 5.10. The inputs are texts T_1, T_2, T_3 (ordered by publication time). A_1, A_2 and A_3 belong to the same theme, whose intersection sentence is A, and similarly for B and C. The themes A and B are topically related, but C is not related. Summary S_1, based only on chronological clues, contains two topical shifts; from A to C and back from C to B. A better summary would be S_2, which keeps A and B together.

5.5.1 The Algorithm

Our goal is to remove disfluencies from the summary by grouping together topically related themes. The main technical difficulty in incorporating cohesion in our ordering algorithm is to identify and to group topically related themes across multiple

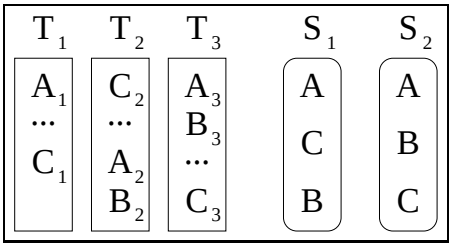

Figure 5.10: Input texts $T_1T_2T_3$ are summarized by the Chronological Ordering (S_1) or by the Augmented algorithm (S_2).

documents. In other words, given two themes, we need to determine if they belong to the same cohesion block. For a single document, topical segmentation (Hearst, 1994) could be used to identify blocks, but this technique is not a possibility for identifying cohesion between sentences across multiple documents. Segmentation algorithms typically exploit the linear structure of an input text; in our case, we want to group together sentences belonging to different texts.

Our solution consists of the following steps. In a preprocessing stage, we segment each input text (Kan, Klavans, and McKeown, 1998) based on word distribution and coreference analysis, so that given two sentences within the same text, we can determine if they are topically related. Assume we have two themes A and B, where A contains sentences $(A_1 \ldots A_n)$, and B contains sentences $(B_1 \ldots B_m)$. Recall that a theme is a set of sentences conveying similar information drawn from different input texts. We denote $\#AB$ to be the number of pairs of sentences (A_i, B_j) which appear in the same text, and $\#AB^+$ to be the number of sentence pairs which appear in the same text and are in the same segment.

In the first stage, for each pair of themes A and B, we compute the ratio $\#AB^+/\#AB$ to measure the relatedness of two themes. This measure takes into

account both positive and negative evidence. If most of the sentences in A and B that appear together in the same texts are also in the same segments, it means that A and B are highly topically related. In this case, the ratio is close to 1. On the other hand, if among the texts containing sentences from A and B, only a few pairs are in the same segments, then A and B are not topically related. Accordingly, the ratio is close to 0. A and B are considered related if this ratio is higher than a predetermined threshold. We determined experimentally its value to be 0.6.

This strategy defines pairwise relations between themes. A transitive closure of this relation builds groups of related themes and, as a result, ensures that themes that do not appear together in any article but which are both related to a third theme will still be linked. This creates an even higher degree of relatedness among themes. Because we use a threshold to establish pairwise relations, the transitive closure does not produce elongated chains that could link together unrelated themes. We are now able to identify topically related themes. At the end of the first stage, they are grouped into blocks.

In a second stage, we assign a time stamp to each block of related themes using the earliest time stamp of the themes it contains. We adapt the CO algorithm described in 5.3.2 to work at the level of the blocks. The blocks and the themes correspond to, respectively, themes and sentences in the CO algorithm. By analogy, we can easily show that the adapted algorithm produces a complete order of the blocks. This yields a macro-ordering of the summary. We still need to order the themes inside each block.

In the last stage of the augmented algorithm, for each block, we order the themes it contains by applying the CO algorithm to them. Figure 5.11 shows an

example of a summary produced by the augmented algorithm.

This algorithm ensures that cohesively related themes will not be spread over the text and decreases the number of abrupt switches of topics. Figure 5.11 shows how the Augmented algorithm improves the sentence order compared with the order in the summary produced by the CO algorithm in Figure 5.8; sentences quoting US officials are now grouped together, and so are the descriptions of the mourning.

Thousands of people have attended a ceremony in Nairobi commemorating the first anniversary of the deadly bombings attacks against U.S. Embassies in Kenya and Tanzania. Kenyans are observing a national day of mourning in honor of the 215 people who died there.

Saudi dissident Osama bin Laden, accused of masterminding the attacks, and nine others are still at large. U.S. federal prosecutors have charged 17 people in the bombings.

President Clinton said, "The intended victims of this vicious crime stood for everything that is right about our country and the world". Albright said that the mourning continues.

Figure 5.11: A summary produced using the Augmented algorithm. Related sentences are grouped into paragraphs.

5.5.2 Evaluation

Following the same methodology used to evaluate the MO and the CO algorithms, we asked the judges to grade 25 summaries produced by the Augmented algorithm. Results are shown in Figure 5.12.

The manual effort needed to compare and judge system output is extensive considering that each human judge had to read three summaries for each input set as well as skim the input texts to verify that no misleading information was introduced in the summaries. We collected a corpus of 25 sets of articles for evaluation. Overall, there were 75 summaries to be evaluated. The size of our corpus is comparable with

the collection used for the DUC evaluation (30 sets of articles). This evaluation shows a significant improvement in the quality of the orderings from the CO algorithm to the Augmented algorithm. To assess the significance of the improvement, we used the Fisher exact test, conflating Poor and Fair summaries into one category (p-value of 0.04). The augmented algorithm also shows an improvement over the MO algorithm (p-value of 0.07).

	Poor	Fair	Good
Majority Ordering	3	14	8
Chronological Ordering	10	8	7
Augmented Ordering	3	8	14

Figure 5.12: Evaluation of the the Majority Ordering, the Chronological Ordering and the Augmented Ordering.

5.6 Related Work

Finding an acceptable ordering has not been studied before in domain independent text summarization. In single document summarization, summary sentences are typically arranged in the same order that they were found in the full document, although (Jing, 1998) reports that human summarizers do sometimes change the original order. In multi-document summarization, the summary consists of fragments of text or sentences that were selected from different texts. Thus, there is no complete ordering of summary sentences that can be found in the original documents.

In domain dependent summarization, it is possible to establish possible orderings *a priori*. A valid ordering is traditionally derived from a manual analysis of a corpus of texts in the domain, and it typically operates over a set of semantic con-

cepts. A semantic representation of the information is usually available as input to the ordering component. For instance, in the specific domain of news on the topic of terrorist attacks, summaries can be constructed by first describing the place of the attack, followed by the number of casualties, who the possible perpetrators are, etc.

Another alternative when ordering information, still in the domain dependent framework, is to use a more data driven approach, which produces a more flexible output. *A priori* defined simple ordering strategies are combined together by looking at a set of features from the input. (Elhadad and McKeown, 2001) use such techniques to produce patient specific summaries of technical medical articles. Examples of features which influence the ordering are presence of contradiction or repetition, relevance to the patient characteristics, or type of results being reported. A linear combination of these features assigns a weight to each semantic predicate to be included in the output, allowing them to be ordered. In this case, the features are domain dependent and have been identified through corpus analysis and interviews with physicians. In the case of a domain independent system, it would be an entire new challenge to define and compute such a set of features.

Producing a good ordering of information is also a critical task for the generation community, which has extensively investigated the issue (McKeown, 1985; Moore and Paris, 1993; Hovy, 1993; Bouayad-Agha, Power, and Scott, 2000; Mooney, Carberry, and McCoy, 1990). One approach is top-down, using schemas (McKeown, 1985) or plans (Dale, 1992) to determine the organizational structure of the text. This approach postulates a rhetorical structure which can be used to select information from an underlying knowledge base. Because the domain is limited, an encoding can be developed of the kinds of propositional content that match rhetorical ele-

ments of the schema or plan, thereby allowing content to be selected and ordered. Rhetorical Structure Theory (RST) allows for more flexibility in ordering content by establishing relations between pairs of propositions. Constraints based on intention (Moore and Paris, 1993), plan-like conventions (Hovy, 1993), or stylistic constraints (Bouayad-Agha, Power, and Scott, 2000) are used as preconditions on the plan operators containing RST relations to determine when a relation is used and how it is ordered with respect to other relations. Another approach (Mooney, Carberry, and McCoy, 1990) is bottom-up and is used to group together stretches of text in a long, generated document by finding propositions that are related by a common focus. Since this approach was developed for a generation system, it finds related propositions by comparisons of proposition arguments at the semantic level. In our case, we are dealing with a surface representation, so we find alternative methods for grouping text fragments.

More recent approaches (Duboue and McKeown, 2001; Kan and McKeown, 2002) automatically estimate constraints on information ordering in limited domains, at the content planning stage. Using a collection of semantically tagged transcripts written by domain experts, (Duboue and McKeown, 2001) identify basic adjacency patterns contained within a plan, as well as their ordering. (Kan and McKeown, 2002) applies Majority Ordering to semantic tags in bibliography entries in order to learn ordering in this domain. MultiGen generates summaries of news on any topic. In such an unconstrained domain, it would be impossible to enumerate the semantics for all possible types of sentences which could match the elements of a schema, a plan or rhetorical relations. For instance, Duboue and McKeown build their content planner based on a set of 29 semantic categories; in our case, there is no

such regularity in the input information. Furthermore, it would be difficult to specify a generic rhetorical plan for a summary of news. Instead, content determination in MULTIGEN is opportunistic, depending on the kinds of similarities that happen to exist between a set of news documents. Similarly, we describe here an ordering scheme that is opportunistic and bottom-up, depending on the cohesion and temporal connections that happen to exist between selected text.

Our ordering component takes place after the content selection of the information in a pipeline architecture, in contrast to generation systems, where usually the ordering and the content selection come in tandem. This separation might come at a cost — if there is no good ordering to the given extracted information, it is not possible to go back to the content selection to extract new information. In summarization, content selection is driven by salience criteria. We believe that the same ordering strategy should work with different content selectors, independently of their salience criteria. Therefore, we choose to keep the two components, selection and ordering, as two separate modules.

5.7 Conclusions and Future Work

In this chapter we investigated information ordering constraints in multi-document summarization in the news genre. We evaluated two alternative ordering strategies, Chronological Ordering (CO) and Majority Ordering (MO). Our experiments show that MO performs well only when all input texts follow similar organization of the information. In the domains where this constraint holds, MO would be an appropriate and highly effective strategy. But in the news genre we cannot make this assumption;

thus it is not an appropriate solution.

The Chronological Ordering (CO) algorithm can provide an acceptable solution for many cases, but is not sufficient when summaries contain information that is not event based. Our experiments, using a corpus that we collected of multiple alternative summaries each of multiple documents, show that cohesion is an important constraint contributing to ordering. Moreover, they also show that appropriate ordering of information is critical to allow for easy comprehension of the summary and that it is not the case that all possible orderings of information are acceptable. We developed an operational algorithm that integrates cohesion as part of the CO algorithm, and implemented it as part of the MultiGen summarization system. Our evaluation of the system shows significant improvement in summary quality.

While in this thesis we focused on augmenting the CO algorithm, we believe that MO is a promising strategy and should not be neglected. It is clear that different forms of summarization are useful in different situations, depending on the intended purpose of the summary and on the types of documents summarized. For our future work, we plan to build on the approach we used for the DUC 2001/2002 evaluations (DUC, 2001; DUC, 2002), where we developed a summarizer that would use different algorithms for summary generation depending on the type of input text. We suspect that ordering strategies may differ also, depending on the type of summary. Our work will first investigate whether we can use our augmented algorithm for other summary types. If the algorithm does not yield good orderings, we will investigate through corpus analysis other summary type specific constraints. We suspect that our augmented algorithm may apply, for instance, to biographical summaries, since the information being summarized is a mixture of event-based information that can

be chronologically ordered along with descriptive information about the person. It is unclear whether it can apply to other types of summaries such as summaries of different events, since pieces of information may not be temporally related to each other. We also plan to identify the types of summaries which would benefit from using the MO algorithm or an augmented version of it (the same way the CO algorithm was augmented with the cohesion constraint).

This chapter concludes the presentation of our summarization strategy. In the next chapter, we present an overall system evaluation.

Chapter 6

Overall Evaluation

Beauty is in the eye of the beholder.

6.1 Summarization Evaluation Methods

The evaluation of an NLP system is a key part of any research or development effort and yet it is probably the most likely to arouse controversy. In the field of automatic summarization, there is no consensus on what is a good way to evaluate summaries, but there is wide agreement that the techniques which are commonly used today are problematic (Jing et al., 1998; Spark Jones, 1999; Marcu and Gerber, 2001).

The inherited difficulty of summary evaluation comes from the fact that there is no single right answer to a summarization task, unlike in other NLP tasks, such as parsing. Different people tend to have different perceptions of what is important in a given text, and consequently they will produce different summaries given the same text. Thus, a naive comparison will not suffice to perform evaluation: if a system produced a summary which is not identical to the "ideal" one produced by a human,

we still do not know whether the automatically produced summary is a good or a bad one. Direct assessment of the summary by a human judge suffers from the same problem — the evaluator may not "like" a summary because it differs from his understanding about what the summary should include. Researchers (Spark Jones, 1999; Teufel, 2001) have argued that the variability between produced summaries can be reduced if human judges know what the summaries are for; or, in other words, in which task and context the summaries will be used. However, so far little progress has been made in identifying such tasks. Several commonly used tasks such as information retrieval and classification do not allow one to reliably distinguish between the performance of different systems, since humans can usually complete these tasks independently of the system output quality.

In this thesis, we do not propose a new evaluation strategy which can overcome the limitations of existing techniques. Instead, we used two existing evaluation strategies which are very different in their nature, hoping that in this case "the whole is greater then the sum of its parts", because each method gives a different perspective on system performance. The first one was "ideal-summary" based evaluation conducted within the Document Understanding Conference (DUC). Despite its limitations, this community-wide evaluation allows us to compare our system with state of the art summarizers. The second evaluation was performed in the context of a particular information access task, aiming to determine whether or not the summaries we produce help users to efficiently access news.

6.2 DUC Evaluation

DUC provided for the first time a framework for the quantitative evaluation of multi-document summarization systems on a standalone basis, unconnected with specific application tasks. Different summarization approaches are compared through precision and recall measures.

6.2.1 DUC Evaluation Background

The DUC multi-document summarization evaluation involved 59 document sets. Each set contained between five and 15 news documents, with an average of ten documents. The document sets were drawn from the TREC collections, and their contents fall into one of the following four types: (1) single event in any domain tracked over short period (at most a seven day window); (2) single event tracked over a long period of time, usually about a particular person; (3) multiple events of a similar nature; and (4) discussion of an issue with some related events. Examples of these four set types, respectively, are an earthquake in Iraq, Elizabeth Taylor's bout with pneumonia, various marathons, and gun control.

For each test data and each target summary size[1] (50, 100, 200, 400 words), one automatically-generated abstract[2] and one extract were submitted from each participating system; in addition, two extracts and one abstract were created by humans. The human judges were given the summaries automatically produced by competing systems and an extract created by humans (*peer extract*). The judges were asked to compare them with a human-created abstract (*model abstract*) and one

[1] The evaluation also included 10-word abstract, which was generated as a headline. Our system is not built to generate headlines, therefore Columbia did not participate in this track.

[2] Terms "abstract" and "extract" distinguish between generated and extracted summaries.

of the human-created extracts (*model extract*). Although two model extracts were available for almost all the document sets, only one comparison was performed for each peer summary for a given summary size.

For each data set and target abstract size, the author of the model abstract assessed the degree of match between the model abstract and various peer abstracts. First, qualitative measures pertaining to the comparison between a model and a peer abstract as a whole were reported on a scale between zero and four. These measures were grammaticality, cohesion and organization.

To calculate quantitative measures of overlap between system- and human-created summaries, the human-created summaries were segmented by hand into *model units* (MUs), which are informational units that should each express one self-contained fact in the ideal case. These units might be sentence clauses; however, they are often sentences. Automatically generated summaries were automatically segmented into *peer units* (PUs), which are always sentences. Subsequently, the assessor located the PUs that covered the content of each MU, if any, and assigned an estimate of the degree of match, between one and four[3].

Content of peer summaries was evaluated using precision and recall[4]. Precision is calculated for each peer summary as the number of PUs matching some MU divided by the number of PUs in the peer summary. Recall was measured as the number of MUs matched divided by the total number of MUs in the model summary.

[3]Matching grades were at least one, since otherwise no PUs were reported for that MUs

[4]Note that there is some argument about how to calculate precision and recall (see (McKeown et al., 2001b) for more details)

6.2.2 Our results

The Columbia Summarizer for DUC 2002, which includes MultiGen as a component, is a composite system that uses different strategies depending on the type of documents in the input set. Our system, MultiGen, was used on only the sets about a single event. Following the DUC guidelines, we assumed that all document sets containing documents that were published within seven days are on a single event; 30 such documents were sent to MultiGen. We considered doing an additional test for similarity between the documents in a document set. We would have done this using the similarity metrics that we currently use for document clustering in the tracking and clustering stage of Newsblaster. However, given the lack of training data for the single event type, we felt that thresholds for Newsblaster could not be reliably determined.

We also implemented the ability to produce extractive summaries, in addition to abstractive summaries. This meant extracting a representative sentence from each theme instead of generating a sentence from the intersection of similar phrases.

Extracts For the extracts, we measured precision and recall, both micro-averaging across all sentences for the produced summaries and model summaries, respectively, in the entire evaluation collection; and macro-averaging, by computing precision and recall for each summary and averaging those across the collection. The results are shown in Table 6.1.

In the evaluation of extracted summaries of all sizes, our system, 24, came in second with respect to precision and third with respect to recall. System 21 beat us on both recall and precision, while system 19 beat us on recall but not on precision.

System code	Recall		Precision	
	Macro-averaged	Micro-averaged	Macro-averaged	Micro-averaged
19	20.70% (1)	21.30% (1)	20.66% (3)	21.14% (3)
20	15.21% (5)	15.78% (5)	14.82% (5)	15.26% (5)
21	20.63% (2)	20.48% (2)	24.90% (1)	25.84% (1)
24	**18.23% (3)**	**17.91% (3)**	**22.11% (2)**	**22.26% (2)**
28	15.83% (4)	16.05% (4)	18.12% (4)	19.23% (4)

Table 6.1: Evaluation scores on extracts for the top five systems, across all summary sizes. Systems listed in order of system code, with Columbia's scores in bold. Ranks shown in parentheses among all 10 systems submitting extracts.

System code	Recall		Precision	
	Macro-averaged	Micro-averaged	Macro-averaged	Micro-averaged
19	18.62% (1)	18.38% (1)	18.67% (3)	18.51% (3)
20	12.43% (5)	12.22% (5)	12.86% (5)	12.78% (5)
21	17.24% (2)	16.26% (2)	21.18% (1)	20.91% (1)
24	**15.67% (3)**	**14.65% (3)**	**19.49% (2)**	**19.05% (2)**
28	12.93% (4)	12.32% (4)	15.01% (4)	15.23% (4)

Table 6.2: Evaluation scores on extracts for the top five systems, on 200 word summaries. Systems listed in order of system code, with Columbia's scores in bold. Ranks shown in parentheses among all 10 systems submitting extracts.

System 28 ranks consistently fourth by all measures, and system 20 fifth; these two systems are clearly separated from the top three by at least two percentage points. This relative ranking also holds if we look at the subsets of 200 word extracts and 400 word extracts separately (Tables 6.2 and 6.3). Micro- or macro-averaging makes very little difference in the relative performance of the top five systems in the vast majority of cases; the one exception is recall for system 21 which moves from second to first on the 400 word summaries when micro-averaging is used.

Looking at all summaries independent of size, humans did better than systems

System code	Recall		Precision	
	Macro-averaged	Micro-averaged	Macro-averaged	Micro-averaged
19	22.78% (2)	22.81% (1)	22.66% (3)	22.46% (3)
20	18.00% (5)	17.61% (5)	16.78% (5)	16.39% (7)
21	24.02% (1)	22.65% (2)	28.61% (1)	28.31% (1)
24	**20.80% (3)**	**19.58% (3)**	**24.73% (2)**	**23.80% (2)**
28	18.73% (4)	17.97% (4)	21.23% (4)	21.19% (4)

Table 6.3: Evaluation scores on extracts for the top five systems, on 400 word summaries. Systems listed in order of system code, with Columbia's scores in bold. Ranks shown in parentheses among all 10 systems submitting extracts.

System code	Coverage		Precision		Topic-related unmarked units
	Macro-avg	Micro-avg	Macro-avg	Micro-avg	
19	21.20% (1)	18.72% (1)	74.52% (2)	71.11% (2)	38.56% (6)
20	16.75% (4)	14.12% (5)	57.19% (6)	56.75% (6)	39.58% (5)
24	**17.90% (2)**	**17.68% (2)**	**69.84% (3)**	**69.73% (3)**	**39.77% (4)**
26	17.01% (3)	15.53% (3)	65.96% (4)	64.94% (5)	46.69% (1)
28	15.61% (5)	15.42% (4)	79.72% (1)	78.90% (1)	31.19% (7)

Table 6.4: Evaluation scores on abstracts for the top five systems, across all summary sizes using length-adjusted mean coverage. Systems listed in order of system code, with Columbia's scores in bold. Ranks shown in parentheses among all 8 systems submitting abstracts.

in most cases on recall (seven out of nine), but by only a small margin (the biggest margin was 7.13 percentage points). On precision, only four out of nine humans beat the top system when micro-averaging is used and two when macro-averaging is used. The difference in the best case, 3.28 percentage points, is even smaller. The numbers of humans exceeding system performance on recall and precision remains relatively constant when we focus on either the 200 word or 400 word summaries, although the difference between the best humans and the top system increased in the former case

System code	Coverage		Precision		Topic-related unmarked units
	Macro-avg	Micro-avg	Macro-avg	Micro-avg	
19	27.83% (1)	25.22% (1)	74.52% (2)	71.11% (2)	38.56% (6)
20	15.40% (5)	17.53% (5)	57.19% (6)	56.75% (6)	39.58% (5)
24	**17.87% (4)**	**19.37% (4)**	**69.84% (3)**	**69.73% (3)**	**39.77% (4)**
26	22.28% (2)	22.24% (2)	65.96% (4)	64.94% (5)	46.69% (1)
28	22.09% (3)	22.09% (3)	79.72% (1)	78.90% (1)	31.19% (7)

Table 6.5: Evaluation scores on abstracts for the top five systems, across all summary sizes using unmodified mean coverage. Systems listed in order of system code, with Columbia's scores in bold. Ranks shown in parentheses among all 8 systems submitting abstracts.

(to 9.74 points for macro-averaged recall, for example), and reduced it in the latter case.

Abstracts For the abstracts, we computed both unadjusted and length-adjusted coverage using the definitions provided by NIST, using both micro- and macro-averaging as defined earlier. We also calculated precision (micro- and macro-averaged), and we have included our score on related but unmarked units, which indicates how many of the system summary sentences were related to the topic of the summary. For macro-averaging we used the mean coverage within each summary rather than the median. Since a large percentage of model units are not covered at all in any peer summary, the median is often very low and obscures differences in coverage between systems for the model units that they do cover.

Table 6.4 shows the length-adjusted scores across all articles, where we rank second in coverage and third in precision, regardless of whether micro- or macro-averaging is used. We are fourth in the score of related but unmarked units. System 19 has done the best on abstracts, ranking first on coverage and second on precision,

using these calculations. However, system 19 does worse on unmarked, related units ranking sixth. System 28 (which ranked fourth on extracts) is a distant third here, ranking first on precision but fourth (macro-averaging) or fifth (micro-averaging) on coverage and seventh on the related but unmarked units. System 21, which performed best on extracts, did not participate in the abstracts evaluation.

If we use instead the unmodified coverage metric, we score somewhat lower on coverage, ranking fourth under both micro- and macro-averaging (Table 6.5). System 19 retains the top position, and system's 28 position improves slightly to third.

We did not perform as well in grammaticality and cohesion due to the fact that linearization component of the system was changed a short time before the competition and, as a result, the full debugging of the system was not completed on time.

6.2.3 Issues with the DUC Evaluation

Despite the careful organization of the evaluation effort, DUC is still an "ideal" summary based evaluation with all of its intrinsic shortcomings. In the multi-document case, the topic diversity of input documents in a set only aggravate the problem. Our analysis revealed that some sets from the single event category contain documents only loosely connected through the common theme. For example, the set of articles D096 of the type "single-event" contains articles on several disjoint topics related to the Super Bowl, ranging from game results to food sold during the event. This set is not an exception in its category; the DUC "single event" type sets are versatile in their structure and topic. This makes it even harder to decide what is "important" in the whole set without any guidelines about what a summary should contain. Conse-

quently, agreement in an extraction task between humans is quite low — Kappa equals 0.2, which corresponds to a random agreement. This results in an ironic situation, where for some texts, automatically generated summaries received a better score than human crafted summaries. This observation is consistent with the detailed statistical analysis we performed for DUC 2001 (McKeown et al., 2001b). In that analysis, we showed that the input set and the human evaluator were the most important factor in predicting the summary score — much more important than the system that did the summarization, and even more important than whether summarization was done by an automated system or human.

We should also notice that DUC participants came up with different, in some cases even controversial, analyses of the results due to different evaluation measures used. Not surprisingly, a list of top performing systems varied from one participant to another.

Another important question which is left unaddressed in the DUC evaluation is whether the summaries produced by the systems are of sufficiently quality to be helpful to users in information access tasks. The evaluation described in the next section aims to answer this question.

6.3 Newsblaster Evaluation

For over a year, MultiGen has ran as part of Columbia Newsblaster, a news browsing system, which aims to help a user keep abreast of current news. In addition to summarization, this system integrates a number of technologies, among them clustering and text classification. While popularity of the Newsblaster system, as revealed by

the number of regular visitors, is evidence that it is useful, we wanted to evaluate the contribution of the summarization component to the functionality of the overall system.

The first issue we address in this evaluation is whether our summaries are really helpful in efficient news access. MultiGen uses quite a sophisticated strategy to produce summaries, but it is still an open question as to whether users prefer natural-language summaries to more simple, but robust, representations of text. For example, the system can more reliably extract keywords from a text than summarize a text. The quality of a summary may vary from one input to another, although it should provide a better indication of the text content than a list of words. Thus, to justify the selection of summaries for Newsblaster, we performed an experiment which allowed users to have either a summary or a list of keywords in their configuration of the system. By monitoring their preferences, we can assess how valuable our summaries are in the context of the information access task.

We complemented this evaluation with a more traditional method, aiming to directly capture user satisfaction with the summaries. In this evaluation, users were asked to evaluate the content and fluency of summaries. We first give more details on the organization of the evaluation and then present the results.

6.3.1 Newsblaster Evaluation Background

The evaluation was performed through a mirror website identical to the continuously running Newsblaster system, but with links to our evaluation materials (described below). The evaluation website was advertised to our colleagues in the Department of Computer Science as well as to a more general lay audience within and outside the

University. All the transactions with this website were recorded, and user identities were tracked through a cookie mechanism, which allowed us to determine unique IDs (by augmenting IP addresses) for each computer system that accessed the site. The participants in the experiment were asked to check the news using Newsblaster for a week, and to answer questions about the system and summaries several times during the week.

During the week of the experiment (January 14, 2002 to January 21, 2002), 94 subjects accessed the system with an average of 2.87 accesses per person. 48 users accessed the system only once and 46 users regularly used the system (number of accesses between two and 41, average number of accesses 4.8). During the first interaction with the system, users were presented with questions about their information needs. We asked how frequently an evaluator checks news and which means he or she uses to obtain news.[5] Almost all of our evaluators regularly check the news: 17 users check news daily, seven read news several times a week and only two evaluators check the news rarely. The majority of the users (16) obtain their news mostly from on-line sources; the rest access news through print (eight) and broadcast media (two). This data shows that the majority of the respondents fit the profile of the target audience of our system—users who regularly follow the news through on-line sources. All of our evaluations were performed using judgments from humans different from the designers of the experiment.

[5]Since these questions were optional, only 26 users answered them.

6.3.2 Usability Evaluation

Initially, every user was randomly assigned either a summary or a keywords list. The system prominently displays an option for changing preferences on the front page and when selected, gives a brief description of the different options. When the user selects a new configuration, the system remembers the choice for future usage. However, the user can change his preferences during any session using a preference button; all changes are recorded in the system log.

Our initial random settings for the 46 users had 24 users starting with the natural language summary option. At the end, 35(76%) users preferred summaries over the keywords. These results are statistically significant at the 3% or less level, and provide strong evidence for user preference of summaries over the keywords. In particular, this percentage of user who selected summary rises with even more regular visitors: 82% for people who used the site at least 3 or at least 4 times, and 90% for people with more than four visits.

One caveat of this experiment should be noted: a number of users after the experiment mentioned that they had difficulty changing the interface to a new setting, and as a result they stayed with their default setting.

6.3.3 Direct Summary Evaluation

The evaluation of summary quality was performed through a questionnaire which users accessed from a link on the main page. The users were asked to perform the evaluation after browsing the site. The questionnaire page opened in a new window, which had links referring the user to the specific page about which the questions were asked (either a link to the main index page, or a particular summary page).

Question	Answer Scale	Answer Distribution
Does the summary content give you a good idea of what the articles are about?	no	9
	mostly	21
	yes	30
How would you describe the organization (e.g., order of information, flow of information) of the summaries?	poor	7
	adequate	22
	good	31
How would you describe the clarity of the summaries?	poor	5
	adequate	21
	good	34

Table 6.6: User satisfaction questionnaire and answer distributions.

Newsblaster generates 34.7 summaries per day on average by MultiGen and DEMS (Schiffman, Nenkova, and McKeown, 2002); two summaries among them were randomly selected for evaluation every day. To ensure that the users carefully examined the summary, they were asked to evaluate one summary per day; on average, each summary was evaluated by 7.5 judges. Overall, we collected 61 judgments for 14 summaries during the week of the evaluation. Each summary in our test set summarized 25.8 articles on average. A summary was evaluated along three dimensions: content, organization and clarity. The results [6] are shown in the last three questions of Table 6.6. The majority of the summaries got the highest rating in all three categories, and only very few summaries received low marks.

Interestingly, there is a high correlation between summary grades in all three categories — if a user rated the summary content high, he frequently rated the readability and organization of the summary high. We hypothesize that users were unable

[6] These results correspond to both DEMS and MultiGen, since during the experiment we did not track the identity of a summarizer.

to look beyond their overall perception of the summary to give a more detailed breakdown of the different qualities of the summary.

6.4 Discussion

In this chapter we described two evaluation efforts aiming at capturing the overall performance of our system. Each of these methods has known limitations; however, looking at the results of both techniques together gives us a better view of system functionality.

In both evaluations, our system performed well. In the DUC evaluation, we scored in second or third place according to all measures of content selection. The Columbia Newsblaster evaluation revealed that our human judges preferred summaries to a list of keywords in the context of an information access task. The users of the system also gave high scores to the system in terms of content, fluency and organization. The difference in scores[7] between the two evaluations can be attributed to two factors: first, we asked the questions in the context of a particular information access task, rather than comparing system-produced summaries to ideal human-generated ones. Second, the data set from which we generated the summaries was particularly suited to the requirements of MultiGen, in the sense that it included multiple related articles. In contrast, the DUC evaluation datasets frequently included only loosely related articles.

We hope that future research will yield a better methods for evaluation of summarization tasks.

[7]The absolute scores of our system in the Newsblaster evaluation are higher that the scores of the DUC 2002 evaluation

Chapter 7

Conclusions and Future Work

7.1 Summary of Main Contributions

In this thesis, we presented information fusion, a novel method for text-to-text generation. Given an input set of multiple documents about the same event, this method identifies common information across input sentences and uses language generation to synthesize them into a summary. The use of generation to merge similar information is a new approach that improves the quality of the resulting summaries, reducing repetition and increasing fluency. Unlike traditional generation methods, information fusion does not require an elaborate semantic representation of the input, but instead relies on knowledge automatically extracted from the input documents and a large text corpus.

Identification of similar information requires coping with the variety of ways in which a piece of information can be expressed. Thus, one of the main sources of knowledge required for information fusion is paraphrasing information. Paraphras-

ing, an important language phenomenon, is largely unaccounted for in the linguistic literature, and was not directly addressed in previous NLP research. We developed a method for paraphrase acquisition from a large body of text and use derived paraphrases in the information fusion algorithm.

Below we summarize technical contributions related to information fusion and paraphrase acquisition:

1. **Sentence fusion**

   Sentence fusion is a novel text-to-text generation technique which, given a set of similar sentences, produces a new sentence containing the information common to most sentences. The generation is performed by reusing and altering phrases from input sentences. Such an algorithm needs to identify the fragments conveying common information and to combine them into a sentence. We showed that both of these tasks can be achieved using shallow analysis techniques. More specifically, our algorithm utilizes a method for sentence alignment, paraphrasing information and a language model. According to our evaluation, this method generates a grammatical sentence which accurately synthesizes input phrases in most cases.

2. **Sentence ordering**

   We developed a corpus-based methodology for studying sentence ordering in multi-document summarization. We conducted experiments which show that ordering significantly affects a reader's comprehension of a text. Our experiments also show that although there is no single ideal ordering of information, the number of good orderings for a given text is limited. Given that there are

multiple acceptable orderings, a text providing one ordering of a set of information does not allow us to differentiate between sentences which "must be" together in all acceptable orderings and sentences which happen to be together in the particular text. This led us to develop a corpus of data sets, each of which contains multiple acceptable orderings of a single text. Using sequence analysis methods to study these orderings, we developed an ordering algorithm which utilizes cohesion and chronological information derived from input texts.

3. **Corpus for paraphrase acquisition**

   Empirical investigation of the paraphrasing phenomenon requires a corpus that contains many instances of paraphrases. We proposed using a collection of texts which are either parallel or comparable. More specifically, we collected a parallel corpus of multiple English translations of the same source text written in a foreign language and a comparable corpus of multiple news articles about the same event. This corpus has already been used by other researchers (Ibrahim, 2002), and we hope that this corpus becomes popular in the field as a means of investigating paraphrasing phenomena.

4. **Algorithm for paraphrase acquisition**

   We developed an unsupervised method for corpus-based identification of paraphrases, which applies a co-training method to contextual and lexico-syntactic features. In contrast to earlier work, our approach allows for identification of multi-word paraphrases, in addition to single-word paraphrases, as well as extraction of compositional rules. We found that our method can handle translations along a continuum of similarity, ranging from parallel translations to

comparable corpora. The method significantly outperforms state-of-the-art MT techniques applied to the paraphrase extraction task. We also showed that augmenting information fusion with paraphrasing information derived by our algorithm improves its results.

5. **Design and implementation of a multi-document summarization system**

   The methods developed within this thesis are embodied in MultiGen, a domain-independent multi-document summarization system. Currently MultiGen operates as part of Columbia's Newsblaster system. Every day, Newsblaster downloads news articles from a variety of sources, clusters articles by topic, and generates a cohesive, readable automatic summary of each document cluster. Newsblaster has an active set of followers, with tens of thousands of hits a day.

7.2 Limitations and Future Work

In our multi-document summarization system, the principal criterion for the selection of summary content is based on the repetition of information in input articles. While this criterion typically results in reasonable summaries, it is naive to expect that a single criterion would work well in all cases. The summarization literature suggests that other types of information such as lexical cohesion and the discourse structure of the input documents should also play an important role in content selection. Even though MultiGen uses lexical chains to filter themes, our summarization system can greatly benefit from additional sources of knowledge to guide the content selection process.

Another serious limitation of our summarization strategy is that we do not take into account discourse constraints during the summary generation process. In our system, each sentence is generated in isolation, independently from what is said before and what will be said after. This negatively influences the selection of referring expressions. For example, all summary sentences may contain the full description of a named entity (e.g. "*President of Columbia University Lee Bollinger*"), while the use of shorter descriptions such as "*Bollinger*" or anaphoric expressions in some summary sentences would increase its readability (Schiffman, Nenkova, and McKeown, 2002). These discourse-based constraints can be incorporated into the sentence fusion algorithm, since our alignment-based representation of themes often contains several alternative descriptions of the same object.

An important goal for future work on sentence fusion is to increase the flexibility of this component. In our current implementation, we took a conservative approach which eliminates some valid combinations of input phrases in order to ensure a well-formed output. This approach permits the possibility of a noisy alignment; furthermore, the language model does not effectively discriminate between grammatical and non-grammatical sentences. We believe that the process of aligning theme sentences can be improved by learning the similarity function, instead of using manually assigned weights. An interesting question is how such a similarity function can be induced in an unsupervised fashion. We can also improve the flexibility of the fusion algorithm by using a more powerful language model. Recent research (Daume et al., 2002) showed that syntax-based language models are more suitable for language-generation tasks; the study of such models is a promising direction to explore.

The remainder of this section is focused on discussing future work related to

paraphrasing. Our method for paraphrase acquisition identifies mainly phrasal lexical paraphrases and local compositional rules. An obvious future research direction is a method for extracting compositional lexico-syntactic paraphrases from a parallel or comparable corpus. Phrasal paraphrases extracted by our method could facilitate the learning of such rules, since knowledge of phrasal equivalence helps to reveal structural similarity of the sentences which contain them. However, learning structural paraphrases poses a number of new challenges. How can we automatically determine the granularity of structural rules? For example, the correct mapping between the pair of sentences "*Syria denied claims by Sharon*" and "*Syria said that Sharon's claim was untrue*" is encoded by the rule "*X denied claims by Y*" $\leftrightarrow$ "*X said that Y's claim was untrue*". A method should be able to select this rule among many other possibilities (e.g. "*X denied*" $\leftrightarrow$ "*X said that*" and "*claim by Y*" $\leftrightarrow$ "*Y's claim was untrue*"). Existing approaches for learning structural paraphrases address this problem by predefining their syntactic typology and the allowable number of arguments. An interesting direction for future research is to develop a method which is not restricted to an *a priori* specified paraphrase type.

Our method uses a parallel or comparable corpus as a source of paraphrases. Another possibility is to use the parallel corpus as a seed for paraphrase extraction from other resources such as large-scale knowledge sources and non-parallel corpora. Since numerous manually-created lexical resources provide classification of lexico-semantic relations between English words, it seems natural to use these thesauri as sources for paraphrases. We only need to know what relations in these resources can produce paraphrases. Our analysis revealed that the vast majority of noun paraphrases belong to the same hyperonym tree in WordNet. However, the distance

between the nodes varies greatly from one pair to another. Obviously, the distance between paraphrases depends on the semantic features of the tree to which the pair belongs. Paraphrases extracted from our corpus can inform the identification of WordNet distances which corresponds closely to the paraphrasing relations.

A more ambitious goal is to use lexical resources to generalize lexico-syntactic paraphrasing rules derived from a parallel corpus. Assume that a pair "*X wrote Y*" ↔ "*X is the author of Y*" is extracted from a parallel corpus. Using "*compose*", a WordNet synonym of "*write*", allows us to induce a new pattern — "*X composed Y*" ↔ "*X is the author of Y*". Such a substitution does not work for every synonym: "*X wrote Y*" can not be paraphrased as "*X is the generator of Y*", even though "*generator*" is a synonym of "*author*".

In this thesis, we focused on the paraphrase acquisition problem. An important future research direction is a method for automatic generation of multiple paraphrases of a given sentence (Barzilay and Lee, 2003). Text-to-text generation systems could employ such a mechanism to produce candidate sentence paraphrases that other system components could filter for length, sophistication level, and so forth. Another interesting application would be to expand existing corpora by providing several versions of their component sentences. This could, for example, aid machine-translation evaluation, where it has become common to evaluate systems by comparing their output against a bank of several reference translations for the same sentences (Papineni et al., 2002). Studies of paraphrases across several domains (Iordanskaja, Kittredge, and Polguere, 1991; Robin, 1994; McKeown, Kukich, and Shaw, 1994) revealed that sentence-level paraphrasing involves more than word-for-word or phrase-by-phrase substitution applied in a domain-and context-independent fashion.

In our future work, we will investigate the induction of generative models from a parallel and comparable corpus for the sentence rewriting task.

Appendix A

Input Example

A.1 Set of Related Articles

Article 1

On 13 Oct 2000, it was reported that at least 30 people in this northern Uganda town have died in recent weeks of a hemorrhagic fever that authorities fear may be caused by the Ebola or Marburg virus.

Blood samples from victims of the outbreak, which has produced tell-tale bleeding from every orifice in many patients here, are being flown to high-security laboratories in South Africa and at the Centers for Disease Control and Prevention in Atlanta.

Investigators who rushed here this week said tests may identify the fever's cause by early next week.

They said several elements of the outbreak, including the infection of medical personnel, suggest involvement of a member of the family Filoviridae, the family of

viruses that includes the Ebola and Marburg viruses.

A total of 3 of the 10 people who died of the virus since 30 Sep 2000 in one Gulu hospital were nursing students who probably cared for infected patients.

Ugandan health officials said at least 20 others have perished with similar symptoms at a second hospital or at home.

In one cluster of huts not far from the center of town, 8 people have died since 20 Sep 2000.

Officials from the Uganda Ministry of Health and the World Health Organization told residents to avoid direct contact with the sick and to rush them to local hospitals where doctors have tried to establish isolation wards.

Ebola and Marburg are the most lethal of the hemorrhagic fever viruses that have emerged in recent decades, notably in Africa.

Ebola virus was first identified in 1976 in the Congo and Sudan, both of which border Uganda, and has since flared up at several other sites. The first sustained outbreak of either virus continues in northeast Congo, where local men working an abandoned gold mine have suffered a steady stream of Marburg virus infections since mid-1998.

Article 2

An outbreak of a mystery disease is centered on the northern town of Gulu. About 30 people have died from an as-yet unidentified disease in northern Uganda.

Health experts, including officials from the World Health Organisation left for Uganda's northern Gulu District on Fri 13 Oct 2000 to investigate the outbreak. The Ugandan Ministry of Health said on Thursday that 42 cases of the disease have been reported.

Symptoms include fever, muscle pains and bleeding from the mouth, nose and anus. Those symptoms have led doctors to suspect that the disease is a hemorrhagic fever, of which the best known are Marburg and Ebola.

The first death occurred on 17 Sep 2000. The victim was reportedly a soldier who had recently served in the Democratic Republic of Congo. An agency (AFP) report states that many other victims have lived near the army barracks.

Blood samples have been sent to South Africa and the United States for testing in an effort to identify the disease.

Article 3

The outbreak of hemorrhagic fever is centered on the northern town of Gulu. Ugandan health authorities are battling to contain an outbreak suspected to be caused by the deadly Ebola virus which has killed at least 31 people in the north of the country. The highly contagious disease, which broke out in the district of Gulu, causes its victims to bleed to death. A further 57 people are known to have contracted the disease but doctors fear that many in remoter villages may have died before they could get medical help.

Efforts to tackle the outbreak have been hampered by the lack of adequate medical facilities, and the effects of a rebel activity in the region. The government and the World Health Organization have sent fact-finding missions to Gulu to investigate the outbreak, but so far have given little practical help.

Symptoms of the mystery illness include fever, muscle pains and bleeding from the mouth, nose and anus. This is the first recorded suspected Ebola outbreak in Uganda and medical officials say they do not yet know how the disease reached Gulu.

But there has been intense speculation in the local press that the virus could have been passed by Ugandan soldiers who have recently returned from postings in the Democratic Republic of Congo. One of the first victims, who died on 17 Sep 2000, was reported to be a soldier who had recently returned from such a posting.

Days later, a woman bled to death after giving birth in a hospital in Gulu. During the following weeks, 7 of her family and friends who attended her burial service were also dead. Doctors believe they could have contracted the disease after washing their hands in the same water at her funeral.

So far 10 people have died in hospital, including 3 nurses treating the sick. The other victims have succumbed in their villages before they could get to medical help.

New arrivals continue to arrive at Gulu's 2 hospitals with 5 more people admitted on Saturday alone. Hospital staff are struggling to deal with the outbreak, but lack basic necessities like adequate protective clothing.

The situation is made worse by the fact that Gulu is at the heart of a 12-year insurgency by rebels based in neighboring Sudan.

A.2 Themes

Theme 1

On 13 Oct 2000, it was reported that at least 30 people in this northern Uganda town have died in recent weeks of a hemorrhagic fever that authorities fear may be caused by the Ebola or Marburg virus.

About 30 people have died from an as-yet unidentified disease in northern Uganda.

Ugandan health authorities are battling to contain an outbreak suspected to be caused by the deadly Ebola virus which has killed at least 31 people in the north of the country.

Theme 2

Blood samples from victims of the outbreak, which has produced tell-tale bleeding from every orifice in many patients here, are being flown to high-security laboratories in South Africa and at the Centers for Disease Control and Prevention in Atlanta.

Blood samples have been sent to South Africa and the United States for testing in an effort to identify the disease.

Theme 3

Investigators who rushed here this week said tests may identify the fever's cause by early next week.

Health experts, including officials from the World Health Organisation left for Uganda's northern Gulu District on Fri 13 Oct 2000 to investigate the outbreak.

The government and the World Health Organization have sent fact-finding missions to Gulu to investigate the outbreak, but so far have given little practical help.

Theme 4

A total of 3 of the 10 people who died of the virus since 30 Sep 2000 in one Gulu hospital were nursing students who probably cared for infected patients.

So far 10 people have died in hospital, including 3 nurses treating the sick.

Theme 5

An outbreak of a mystery disease is centered on the northern town of Gulu.

The outbreak of hemorrhagic fever is centered on the northern town of Gulu.

Theme 6

Symptoms include fever, muscle pains and bleeding from the mouth, nose and anus.

Symptoms of the mystery illness include fever, muscle pains and bleeding from the mouth, nose and anus.

Theme 7

The first death occurred on 17 Sep 2000. The victim was reportedly a soldier who had recently served in the Democratic Republic of Congo.

One of the first victims, who died on 17 Sep 2000, was reported to be a soldier who had recently returned from such a posting.

References

Agichtein, Eugene and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*, pages 85–94.

Al-Onaizan, Yaser, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz Joseph Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation. Technical report, CLSP/JHU. Final Report, JHU Workshop.

Arbatchewsky-Jumarie, Nadia, Louise Dagenais, Leo Elnitsky andLidija Iordanskaja, Marie-Noelle Lefebvre, and Suzanne Mantha. 1984. *Dictionnaire explicatif et combinatoire du franais contemporain.* Les Presses de l'Universit de Montreal.

Baayen, Harald, Richard Piepenbrock, and Hedderik van Rijn, editors. 1993. *The CELEX Lexical Database(CD-ROM).* Linguistic Data Consortium, University of Pennsylvania.

Barzilay, Regina and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Barzilay, Regina, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Barzilay, Regina and Lillian Lee. 2003. Learning to paraphrase: An unsupervised

approach using multiple-sequence alignment. In *Proceedings of HLT-NAACL*, pages 16–23.

Barzilay, Regina and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the ACL/EACL*, pages 50–57.

Barzilay, Regina, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the ACL*, pages 550–557.

Blum, Avrim and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 92–100.

Bouayad-Agha, N., R. Power, and D. Scott. 2000. Can text structure be incompatible with rhetorical structure? In *Proceedings of the First International Conference on Natural Language Generation (INLG'2000)*, Mitzpe Ramon, Israel.

Brin, Sergey. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, pages 172–183.

Brown, Peter F., Stephen Della Pietra, Vincent Della Pietra, and Robert Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chen, Stanley F. and Joshua T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the ACL*, pages 310–318.

Chomsky, Noam. 1957. *Syntactic Structures.* The Hague: Mouton Publishers, Paris.

Clark, Eve V. 1992. Conventionality and contrasts: pragamtic principles with lexical consequences. In Andrienne Lehrer and Eva Feder Kittay, editors, *Frame, Fields, and Contrasts: New Essays in Semantic and Lexical Organization.* Lawrence Erlbaum Associates, pages 171–188.

Clough, Paul, Robert Gaizauskas, Scott Piao, and Yorick Wilks. 2002. Meter: Measuring text reuse. In *Proceedings of the ACL*, pages 152–159.

Cohen, William, Robert Schapire, and Yoram Singer. 1999. Learning to order things. *Journal of Artificial Intelligence*, **10**:243–270.

Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the ACL/EACL*, pages 16–23.

Collins, Michael and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of EMNLP.*

Cruse, A. D. 1986. *Lexical Semantics.* Cambridge University Press.

Dale, Robert. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes.* MIT Press, Cambridge, MA.

Daume, Hal, Kevin Knight, Irene Langkilde-Geary, Daniel Marcu, and Kenji Yamada. 2002. The importance of lexicalized syntax models for natural language generation tasks. In *Proceedings of INLG.*

de Beaugrande, Robert and Wolfgang V. Dressler. 1981. *Introduction to Text Linguistics.* Longman, New York, NY.

Dras, Mark. 1999. *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text.* Ph.D. thesis, Macquarie University, Australia.

Duboue, Pable and Kathleen McKeown. 2001. Empirically estimating order constraints for content planning in generation. In *Proceedings of the ACL/EACL*, pages 172–179.

2001. Proceeding of the first document understanding conference (duc).

2002. Proceeding of the second document understanding conference (duc).

Duclaye, Florence, Francois Yvon, and Olivier Collin. 2002. Using the web as a linguistic resource for learning reformulations automatically. In *Proceedings of LREC*, volume 2, pages 390–396.

Edmonds, Philip and Graeme Hirst. 2002. Near synonymy and lexical choice. *Computational Linguistics*, 28(2):105–144.

Elhadad, Michael, Yael Netzer, Regina Barzilay, and Kathleen McKeown. 2001. Ordering circumstantials for multi-document summarization. In *Proceedings of BISFAI.*

Elhadad, Noemie and Kathleen R. McKeown. 2001. Generating patient specific summaries of medical articles. In *Proceedings of the NAACL 2001 Workshop on Automatic Summarization.*

Filatova, E. and E. Hovy. 2001. Assigning time-stamps to event-clauses. In *Proceedings of the AACL/EACL 2001 Workshop on Temporal and Spatial Information Processing.*

Frege, Gottlob. 1892. *Über Sinn and Bedeutung.* Zeitscrift für Philosophie und Philosophie Kritik.

Gale, William, Kenneth Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *In Working Notes, AAAI Fall Symposium Series, Probabilistic Approaches to Natural Language*, pages 54–60.

Gale, William and Kenneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the ACL*, pages 1–8.

Galil, Zvi and Nimrod Megiddo. 1977. Cyclic ordering is np-complete. *Theoretical Compter Science*, **5**:179–182.

Gleitman, Lila R. and Henry Gleitman. 1970. *Phrase and Paraphrase: Some Innovative Uses of Language.* W.W. Norton & Company, New York.

Halliday, Michael. 1985. *An introduction to functional grammar.* Edward Arnold, UK.

Halliday, Michael and Ruqaiya Hasan. 1976. *Cohesion in English.* Longman.

Harabagiu, Sanda, George Miller, and Dan Moldovan. 1999. Wordnet 2 — a morphologically and semantically enhanced resource. In *Proceedings of SIGLEX*, pages 1–8.

Harabagiu, Sanda and Dan Moldovan. 2000. Enriching the wordnet taxonomy with contextual knowledge acquired from text. In S. Shapiro and L. Iwanska, editors, *Natural Language Processing and Knowledge Representation: Language*

for Knowledge and Knowledge for Language. AAAI/MIT Press, pages 301—334.

Harris, Zellig. 1981a. Co-occurence and transformation in linguistic structure. In Henry Hiz, editor, *Papers on Syntax*. D.Reidel Publishing Company. (first published in 1957).

Harris, Zellig. 1981b. The two systems of grammar: Report and paraphrase. In Henry Hiz, editor, *Papers on Syntax*. D.Reidel Publishing Company. (first published in 1969).

Hasan, Ruqaiya. 1984. Coherence and cohesive harmony. In J. Flood, editor, *Understanding Reading Comprehension*. International Reading Association, Newark, DE.

Hatzivassiloglou, Vasileios, Judith Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of EMNLP*.

Hatzivassiloglou, Vasileios and Kathleen McKeown. 1993. Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning. In *Proceedings of the ACL*, pages 172–182.

Hearst, Marti. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the ACL*, pages 9–16.

Hindle, Donald. 1990. Noun classificatin from predicate-argument structures. In *Proceedings of the ACL*, pages 268–275.

Hovy, Eduard. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63. Special Issue on NLP.

Ibrahim, Ali. 2002. Extracting paraphrases from aligned corpora. Master's thesis, MIT.

Iordanskaja, Lidija, Richard Kittredge, and Alain Polguere, 1991. *Natural language Generation in Artificial Intelligence and Computational Linguistics*, chapter 11. Kluwer Academic Publishers.

Jacquemin, Christian. 1999. Syntagmatic and paradigmatic representations of term variations. In *Proceedings of the ACL*, pages 341–349.

Jacquemin, Christian, Judith Klavans, and Evelyne Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proceedings of the ACL*, pages 24–31.

Jing, Hongyan, Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1998. Summarization evaluation methods: Experiments and analysis. In *AAAI Symposium on Intelligent Summarization*, Stanford University, CA.

Jing, Hongyang. 1998. Summary generation through intelligent cutting and pasting of the input document. Technical report, Columbia University.

Jing, Hongyang and Kathleen McKeown. 2000. Cut and paste based summarization. In *Proceedings of the NAACL*.

Kan, Min-Yen, Judith Klavans, and Kathleen McKeown. 1998. Linear segmentation

and segment relevence. In *Proceedings of 6th International Workshop of Very Large Corpora (WVLC-6).*

Kan, Min-Yen and Kathleen R. McKeown. 2002. Corpus-trained text generation for summarization. In *Proceedings of INLG*, Arden House, NJ.

Knight, Kevin and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceeding of AAAI*, pages 703–710.

Kupiec, Julian M., Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of SIGIR*, pages 68–73.

Labov, William. 1972. *Language in the inner city; studies in the Black English vernacular.* University of Pennsylvania Press, Philadelphia.

Landis, J. Richard and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

Langkilde, Irene and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of ACL/COLING*, pages 704–710.

Lees, Robert. 1960. *The Grammar of English Nominalizations.* Indiana University Center in Antropology, Folklore and Linguistics.

Lin, Chin-Yew and E. Hovy. 2001. Neats: A multidocument summarizer. In *Proceedings of the First Document Understanding Workshop (DUC).*

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*, pages 768–774.

Lin, Dekang and Patrick Pantel. 2001. Dirt — discovery of inference rules from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2001*, pages 323–328.

Livant, W. H. 1961. Productive grammatical operations: The noun-compounding of 5-years-olds. *Language Learning*, 12:15–26.

Lyons, John. 1977. *Semantics*. Cambridge University Press.

Mani, Inderjeet and Eric Bloedorn:. 1997. Multi-document summarization by graph search and matching. In *Proceedings of AAAI*, pages 622–628.

Mani, Inderjeet and George Wilson. 2000. Robust temporal processing of news. In *Proceedings of the ACL*.

Manning, Christopher D. and Hinrich Schutze. 2000. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Marcu, Daniel. 1997. From discourse structures to text summaries. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 82–88.

Marcu, Daniel and Laurie Gerber. 2001. An inquiry into the nature of multidocument abstracts, extracts, and their evaluation. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization*.

McCawley, James. 1978. Conversational implicature and the lexicon. In Peter Cole, editor, *Syntax and Semantics 9: Pragmatics*. Academic Press.

McCoy, Kathleen and John Cheng. 1991. Focus of attention: Constraining what can be said next. In C. Paris, W. Swartout, and W. Mann, editors, *Natural*

Language Generation in Artificial Intelligence and Computational Linguistics. Kluwer Academic Publishers.

McKeown, Kathleen. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, England.

McKeown, Kathleen, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *Proceedings of HLT*.

McKeown, Kathleen, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulati0n: Progress and prospects. In *Proceedings of AAAI*, pages 453–360.

McKeown, Kathleen, Karen Kukich, and James Shaw. 1994. Practical issues in automatic documentation generation. In *Proceedings of ANLP*, pages p. 7–14, Stuttgart, Germany, October.

McKeown, Kathleen R., Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Min Yen Kan, Barry Schifman, and Simone Teufel. 2001a. Columbia multi-document summarization: Approach and evaluation. In *Proceedings of the Document Understanding Workshop (DUC).*

McKeown, Kathleen R., Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Min Yen Kan, Barry Schifman, and Simone Teufel. 2001b. Columbia multi-

document summarization: Approach and evaluation. In *Proceedings of the Document Understanding Conference(DUC01)*.

Melamed, Dan I. 2001. *Empirical Methods for Exploiting Parallel Texts*. MIT press.

Melcuk, Igor. 1988. *Dependency Syntax: Theory and Practice*. Albany:State University of New York Press.

Melcuk, Igor. 1996. Lexical functions: A tool for the description of lexical relations in a lexicon. In Leo Wanner, editor, *Lexical Functions in Lexicograpy and Natural Language Processing*. John Benjamin Publishing Company, pages 37–95.

Metropolis, Nicholas C., Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward H. Teller. 1953. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087—1092.

Mikheev, Andrei. 1997. The LTG part-of-speech tagger. University of Edinburgh.

Miller, G.A., R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, 3(4):235–245.

Mooney, David, Sandra Carberry, and Kathleen McCoy. 1990. The generation of high-level structure for extended explanations. In *Proceeding of COLING*, pages 276–281.

Moore, Johanna and Cecile Paris. 1993. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Journal of Computational Linguistics*, 19(4).

Paciorek, Chris and Roni Rosenfeld. 2000. Minimum classification error training in exponential language models. In *Proceedings of NIST/DARPA Speech Transcription Workshop,*.

Paice, Chris D. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, **26**:171–186.

Pang, Bo, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of HLT-NAACL*, pages 181–188.

Papineni, Kishore A., Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the ACL*, pages 183–190.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Kennery. 1997. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, second edition.

Quine, Willard Van Orman. 1985. Two dogmas of empiricism. In A. P. Martinich, editor, *Philosophy of Language.* Oxford Unviversity Press, chapter Four.

Quirk, Randolph, Sidney Greenbaum, Geoffret Leech, and Jan Svartnik. 1985. *A Comprehensive Grammar of English.* Longman Group Ltd, London, UK.

Radev, Dragomir, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the ANLP/NAACL 2000 Workshop on Automatic Summarization.*

Radev, Dragomir R. and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, **24**(3):469–500, September.

Ravichandran, Deepak and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the ACL*, pages 41–47.

Riloff, Ellen and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level boot-strapping. In *Proceedings of AAAI*, pages 1044–1049.

Robin, Jacques. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background: Corpus-Based Analysis, Design, Implementation, and Evaluation.* Ph.D. thesis, Department of Computer Science, Columbia University, NY.

Schiffman, Barry, Ani Nenkova, and Kathleen R. McKeown. 2002. Experiments in multidocument summarization. In *Proceedings of HLT.*

Schutze, Hinrich. 1992. Context space. In *Working Notes, AAAI Fall Symposium on Probabilistic Approaches to Natural Language.*

Shinyama, Yusuke, Satoshi Sekine, Kiyoshi Sudo, and Ralph Grishman. 2002. Automatic paraphrase acquistion from news articles. In *Proceedings of HLT*, pages 40–46.

Siegel, Sidney and N.John Castellan. 1988. *Non Parametric Statistics for Behavioral Sciences.* McGraw-Hill.

Spark Jones, Karen. 1999. Automatic summarizing: factors and directions. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Summarization.* The MIT Pres, Cambridge, Massachusetts, pages 1–12.

Späth, Helmuth. 1985. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples.* Ellis Horwood, Chichester, West Sussex, England.

Sudo, Kiyoshi and Satoshi Sekine. 2001. Automatic pattern acquistion for Japanese information extraction. In *Proceedings of HLT*, pages 40–46.

Teufel, Simone. 2001. Task-based evaluation of summary quality: Describing relationships between scientific papers. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization.*

Walker, Marilyn. 1992. Redundancy in collaborative dialogue. In *Proceeding of COLING.*

Walker, Marilyn. 1993a. *Informational Redundancy and Resource Bounds in Dialogue.* Ph.D. thesis, University of Pennsylvania.

Walker, Marilyn. 1993b. When given information is accented: Repetition, paraphrase and inference in dialogue. In *Proceedings of Linguistic Society of America.*

Wechsler, Robert. 1998. *Performing Without a Stage: The Art of Literary Translation.* Catbird Press.

Wiebe, Janyce, Tom O'Hara, Thorsten Ohrstrom-Sandgren, and Kenneth McKeever. 1998. An empirical approach to temporal reference resolution. *Journal of Artificial Intelligence*, **9**:247–293.

Yang, Yiming, Tom Pierce, and Jaime Carbonell. 1998. A study on retrospective and on-line event detection. In *Proceedings of SIGIR*, pages 28–36.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the ACL*, pages 189– 196.