Generating Natural Language Summaries from Multiple On-Line Sources: Language Reuse and Regeneration

Dragomir R. Radev

TR CUCS-025-99

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

1999

Generating Natural Language Summaries from Multiple On-Line Sources: Language Reuse and Regeneration

Dragomir R. Radev

The abundance of newswire on the World-Wide Web has resulted in at least four major problems, which seem to present the most interesting challenges to users and researchers alike: size, heterogeneity, change, and conflicting information.

Size: several hundred newspapers and news agencies maintain their Web sites with thousands of news stories in each.

Heterogeneity: some of the data related to news is in structured format (e.g., tables); more exists in semi-structured format (e.g., Web pages, encyclopedias, textual databases); while the rest of the data is in textual form (e.g., newswire).

Change: most Web sites and certainly all news sources change on a daily basis.

Disagreement: different sources present conflicting or at least different views of the same event.

We have approached the second, third, and fourth of these four problems from the point of view of text generation. We have developed a system, SUMMONS, which when coupled with appropriate information extraction technology, generates a specific genre of natural language summaries of a particular event (which we call briefings) in a restricted domain. The briefings are concise, they contain facts from multiple and heterogeneous sources, and incorporate evolving information, highlighting agreements and contradictions among sources on the same topic.

We have developed novel techniques and algorithms for combining data from multiple sources at the conceptual level (using natural language understanding), for identifying new information on a given topic; and for presenting the information in natural language form to the user. We named the framework that we have developed for these problems *language reuse and regeneration* (LRR). Its novelty lies in the ability to produce text by collating together text already written by humans on the Web.

The main features of LRR are: increased robustness through a simplified parsing/generation component, leverage on text already written by humans, and facilities for the inclusion of structured data in computer-generated text.

The present thesis contains an introduction to LRR and its use in multi-document summarization. We have paid special attention to the techniques for producing conceptual summaries of multiple sources, to the creation and use of a LRR-based lexicon for text generation, to a methodology used to identify new and old information in threads of documents, and to the generation of fluent natural language text using all the components above.

The thesis contains evaluations of the different components of SUMMONS as well as certain aspects of LRR as a methodology. A review of the relevant literature is included as a separate chapter.

Contents

List of Figures

List of Tables

Acknowledgments

This thesis would not have been possible without the support of a multitude of people. First, my wife Axinia managed to cope with my hectic schedule and provide me with love and understanding. My daughter, Laura, inspired me by being the sweetest and most intelligent creature around. Kathleen McKeown, my thesis adviser, led me through the ocean of Natural Language Processing and gave me the once-in-a-lifetime opportunity to become a real researcher.

So many other people helped me throughout my 5+ years at Columbia and deserve my most sincere gratitude.

- my parents Elena and Radko, my sister Irina, and the rest of my family for the moral support from thousands of miles away,
- my thesis committee: Alfred Aho, Eduard Hovy, John Kender, and Julian Kupiec for reading my thesis and giving me useful feedback to make sure that the thesis becomes what it is now,
- Luis Gravano for being a good friend and mentor,
- the Columbia Digital Libraries group: Shih-Fu Chang, Alejandro Jaimes, Vasileios Hatzivassiloglou, Carl Sable, John Smith, Seungyup Paek, and Kazi

Zaman for the fruitful discussions,

- my officemates: Aya Aner, Regina Barzilay, Dinkar Bhat, Yong Feng, and Ofer Wainberg for bearing with me,
- my collaborators from industry: Evelyne Tzoukermann from Lucent Technologies Bell Labs, Karen Kukich and Rebecca Passonneau from Bellcore, and Wlodek Zadrozny from IBM TJ Watson Center,
- the Natural Language group: Pascale Fung, Hongyan Jing, Min-Yen Kan, Judith Klavans, Olga Merport, Shimei Pan, James Shaw, Eric Siegel, and Nina Wacholder for giving interesting practice talks and attending mine,
- my project students: Srikant Krishna, Amy Lau, Efrat Levy, Barry Schiffman, and Christopher Small for following my advice even when they had other plans in mind,
- the FUF/Surge dream team: Michael Elhadad and Jacques Robin,
- my favorite dean: Zvi Galil who believed in me from the very beginning,
- the staff of the Department of Computer Science: Rosemary Addarich, Alice Cueba, Melbourne Francis, Patricia Hervey, Martha Zadok, Renate Valencia, and Mary van Starrex, for being model administrators and disproving Columbia's red-tape image,
- the technical support folks from CRF (especially Ion Badulescu, Ashutosh Dutta, Sarmistha Dutta, Alex Shender, and Erez Zadok) for satisfying all my whims (especially in the middle of the night and on weekends and holidays),

- my undergraduate professors Iwan Tabakow and Danny Kopec for showing me how interesting Computer Science can be, and

- my friends Ivaylo Elenkov, Anton Kuzmanov, Stefan Tsonchev, and Daniel Nikovski for providing me with moral support.

I have certainly omitted some important people from this list. My apologies to all of them. I am also grateful for the useful comments from the anonymous reviewers and the editors of the *Computational Linguistics* Special Issue on Natural Language Generation and the various conferences and workshops in which I presented parts of the work included here. Select material in the thesis is based upon work supported by the National Science Foundation under Grants No. GER-90-24069, IRI-96-19124, IRI-96-18797, and CDA-96-25374. Any opinions, findings, and conclusions or recommendations expressed in the dissertation are those of the author and do not necessarily reflect the views of the National Science Foundation. Some parts of the work were also supported by a grant from Columbia University's Strategic Initiative Fund sponsored by the Provost's Office.

На Аксиния и Лора

(To Axinia and Laura)

Chapter 1

Introduction

1.1 Description of the problem

One of the major problems with the Internet is the abundance of information and the resulting difficulty for a typical computer user to find and read all existing documents on a specific topic. A recent study has shown that there are in excess of 360 million Web sites [Lawrence and Giles, 1998]. Even within the domain of current news, the user's task is infeasible. There exist now (as of September 1998) more than 100 sources of live newswire on the Internet, mostly accessible through the World-Wide Web [Berners-Lee, 1992]. Some of the most popular sites include news agencies and television stations like Reuters News [Reuters, 1998], CNN's Web site [CNN, 1998], and ClariNet's e.News on-line newspaper [ClariNet, 1998], as well as on-line versions of print media such as the New York Times on the Web [NYT, 1998].

For most classes of users of news, it is practically impossible to go through megabytes of news every day to select articles they wish to read. Even in the cases when the user can actually select all news relevant to the topic of his interest, he will still be faced with the problem of selecting a small subset that he can actually read in a limited time from the immense corpus of news available. Hence, there is a need for information retrieval as well as for summarization facilities.

There currently exist more than 40 information retrieval services on the World-Wide Web, such as DEC's AltaVista [AltaVista, 1998], Lycos [Lycos, 1998], and DejaNews [DejaNews, 1998], all of which allow keyword searches for recent news. However, only recently have there been practical results in the area of text summarization.

Summaries can be used to determine if any of the retrieved articles are relevant (thereby allowing the user to avoid reading those that are not) or can be read in place of the articles to learn about information of interest to the user. Existing summarization systems (e.g., [Preston and Williams, 1994, NetSumm, 1998, Kupiec *et al.*, 1995, Rau *et al.*, 1994]) typically use statistical techniques to extract relevant sentences from a document. This domain-independent approach produces a summary of a single article at a time which can indicate to the user what the article is about. In contrast, the work presented in this thesis focuses on generating a briefing, that is a type of informative summary that *briefs* the user on information in which he has indicated interest[1]. Such briefings pull together information of interest from multiple sources, aggregating information to provide generalizations, similarities, and differences across articles, and changes in perspective across time. Briefings do not necessarily fully summarize the articles retrieved, but they update the user on information he has specified is of interest.

[1] We will use the terms briefing and summary interchangeably in this thesis.

Definition 1 *: A* **briefing** *is a concise summary of the factual matter of a set of news articles on the same or related events.*

Some characteristics that distinguish a briefing from the general concept of a summary are:

- Briefings are used to keep a person up to date on a certain *event* while what summaries (or abstracts) typically summarize an *article*. Thus, briefings need to convey information about the event using appropriate historical references and the context of prior news.

- Briefings focus on certain types of information that are present in the source text in which the reader has expressed interest. They deliberately ignore facts that are tangential to the user's interests, whether or not these facts are the focus of the article. In other words, briefings are more user-centered than general summaries; the latter convey information that the writer has considered important, whereas briefings are based on information that the user is looking for.

- Briefings may include information drawn from sources other than the articles being summarized, for example from an encyclopedia or database.

1.2 Approach and summary of contributions

An inherent problem to summarizers based on sentence extraction[2] is the potential lack of discourse-level fluency in the output. The extracted sentences fit well

[2]Such summarizers produce a summary by picking sentences from the input text based on some criteria, such as position and use of certain words. For an overview of such techniques, refer to [Paice, 1990].

together only in the case they are adjacent in the source document. Discontinuous segments require some sort of text planning, and possibly text massaging (see Section 1.6). Because SUMMONS uses language generation techniques to plan and produce the content and wording of the summary at a paragraph level based on information extracted from input articles, it has all the necessary information to produce a fluent surface summary.

Several other problems with sentence extracts are listed below. Note that some of them arise in single document summarization while others arise when trying to summarize from multiple documents.

- One document - one summary: summarization of multiple articles based on sentence extraction merely involves the concatenation of the summaries of the individual articles.
- No explicit comparisons: when two articles refer to the same event and fact from different perspectives, the summaries do not reflect it.
- No tracking of an event over time: it is hard to see an event in its progression.

The rest of this section lists this dissertation's major contributions. The rest of Chapter 1 describes each of them in detail.

- The use of information extraction in SUMMONS (Section 1.3).
- Summarization of multiple articles (Section 1.4).
- Summarization of multiple types of sources (Section 1.5).
- Language reuse and regeneration (Section 1.6).

- Automated acquisition of additional knowledge resources for use in generation (Section 1.7).

1.3 The use of information extraction in SUMMONS

Most research issues described in the thesis are illustrated with examples from our prototype system, called SUMMONS[3] [McKeown and Radev, 1995, Radev, 1996, Radev and McKeown, 1997], which introduces novel techniques in the following areas:

- It *briefs* the user on a *news event* using tools related to information extraction, conceptual combination, and text generation.
- It *combines* information from *multiple* news articles into a coherent summary using symbolic techniques.
- It *augments* the resulting summaries using descriptions of entities obtained from on-line sources.

Many NLP systems are developed in restricted domains and SUMMONS is not an exception:

> "The richness and breadth of natural language means that any attempt at a computational treatment has to narrow its focus in various respects. Apart from concentrating on particular linguistic phenomena, it is usual to also concentrate on a particular domain of application." [Dale, 1992]

[3] SUMMONS stands for SUMMarizing Online NewS articles.

We have chosen the domain of news on terrorism for several reasons. First, there is already a large body of related research projects in information extraction, knowledge representation and text planning in the domain of terrorism. For example, earlier systems developed under the DARPA Message Understanding Conference (MUC) [Riloff and Lehnert, 1994, Lehnert *et al.*, 1993, Fisher *et al.*, 1995, Grishman *et al.*, 1992, Ayuso *et al.*, 1992, Rau *et al.*, 1992] were in the terrorist domain. We can build on these systems without having to start from scratch. Second, this domain is important to a variety of users, including casual news readers, journalists, and security analysts. Third, SUMMONS is being developed as part of a general environment for illustrated briefing over live multimedia information [Aho *et al.*, 1998]. Of all MUC system domains, terrorist articles are more likely than other domains that were explored (such as mergers and acquisitions or management succession) to have a variety of related pictorial images. Finally, the dynamics of a terrorist event and its development and ramifications present the problems that we were interested in addressing: how to account for conflicting or complementary sources of information at the conceptual level and how to realize linguistically that disagreement or complementation.

In order to extract information of interest to the user, SUMMONS makes use of components from several MUC systems. The output of such modules is in the form of templates that represent certain pieces of information found in the source news articles, such as victims, perpetrators, date, location, or type of event (a MUC template is a list of attribute-value pairs that summarize the important semantic roles in a news story on terrorism). By relying on these systems, the task we have addressed is happily more restricted than direct summarization of full text. This has allowed us to focus on issues related to the combination of information in the templates and the generation of text to express the result.

Before we are able to port our system to other domains, we would first need to develop new templates and the information extraction rules required for them. While this is a task we leave to those working in the information extraction field, we note that there do exist tools for semi-automatically acquiring such rules [Lehnert *et al.*, 1993, Soderland *et al.*, 1995, Fisher *et al.*, 1995]. This fact helps to alleviate the otherwise knowledge-intensive nature of the task. We should note, however, that we have built some tools for domain-independent information extraction. For example, our work on extracting descriptions of individuals, locations and organizations and representing them in a formalism that facilitates reuse of the descriptions in summaries can be used in any domain.

In the remainder of this chapter, we highlight the novel techniques of SUMMONS and explain why they are important for our work.

1.4 Summarization of multiple articles

Given the omnipresence of on-line news services, one can expect that any interesting news event will be covered by several, if not most, services. If different sources present exactly the same information, the user clearly only needs to have access to one of them. Practically, this assumption doesn't hold, as different sources provide updates from different perspectives and at different times. An intelligent summarizer's task is to obtain as much information from the multiple sources as possible, combine it, and present it in a concise form to the user. For example, if two sources of information report a different number of casualties in a particular incident, SUMMONS must report the contradiction and attribute the contradictory information to their sources, rather than select one of the contradictory pieces without presenting the alternative to the user.

With a few exceptions (as explained in Chapter 10), all existing summarizers provide summaries of single articles by extracting sentences from them. If such systems were applied to a series of articles, they might be able to extract sentences that have words in common with the other articles, but they would be unable to indicate how sentences that were extracted from different articles were similar. Moreover, they would certainly not be able to indicate significant differences between articles. In contrast, our work focuses on processing of information from multiple sources to highlight agreements and contradictions as part of the summary.

1.5 Summarization of multiple types of sources

A problem related to summarizing multiple news articles is the problem summarizing material that is not fully in textual (news) format. For example, a summary that mentions an entity (e.g., a person, a place, or an organization) may be enhanced to include some background about the entity. Such background information may be available in the source articles, but often the best (or the only) source of such information may be in a non-textual source of information, such as an encyclopedia or a table of countries and their political leaders. We classify the types of information sources that can be used to extract content for a summary according to two criteria:

- *historical vs. current newswire*: **current newswire** includes the articles from which the main content of the summary is produced, while **historical newswire**, though on a similar topic, is only used to add background information to the summary (e.g., how the whole story began), due to chronological and logical restrictions.

- *textual vs. non-textual sources*: **textual sources** are in free-text format and can include news articles or encyclopedia entries, while **non-textual sources** are in a structured form and are typically represented as database relations or ontologies.

We should note that a few components of SUMMONS produce non-textual sources of information from textual ones using information extraction.

1.6 Language reuse and regeneration

The contributions listed in Sections 1.3– 1.5 are directly related to knowledge-based summarization. On the other hand, this thesis also attempts to address a more general linguistic problem which, while clearly applicable to summarization (as shown in the pages to come), can have a potentially significant impact on Natural Language Generation in general.

We introduce the dual concepts of **language reuse** and **language regeneration** which we will collectively refer to as **language reuse and regeneration (LRR)**. We worked on LRR to be able to exploit text already written by humans.

Language reuse involves two components: a *source* text, written by a human, and a *target* text, that is to be automatically generated by a computer, partially making use of structures reused from the *source* text. Surface structures are extracted automatically from the source text, along with the appropriate syntactic, semantic, and pragmatic constraints under which they are used.

Language regeneration is related to language reuse with the relaxation of one constraint: in language regeneration, the text to be reused is first processed by some transformation and only the modified text is reused in the produced text. The transformation is required because some phrases can only be reused after some small modifications are made to them (e.g., "his successor" –> "Deng Xiaoping's successor").

This thesis defines and motivates these dual concepts, and provides examples of how they are used to produces summaries by SUMMONS.

1.7 Automatic acquisition of lexical resources for use in generation

We show how the summary generated using symbolic techniques can be enhanced so that it includes descriptions of entities (such as people, places, or organizations) it contains. The descriptions are automatically extracted from on-line sources of past news using domain-tailored information extraction techniques. SUMMONS adds these descriptions to the generated summaries to put the entities in context.

1.8 Structure of the thesis

In addition to the current introductory chapter, this thesis includes fourteen chapters, organized in four parts, as well as five appendices.

Part I covers only the generation of the so-called **base summary**, i.e., the information generated derives solely from the current news articles being summarized, without additional enhancements. It describes the method used to generate summaries from multiple documents. Chapter 2 provides an architectural overview of SUMMONS. In Chapter 3, we describe how we collected and analyzed the corpora of news used in the design of SUMMONS. Chapter 4 describes the domain model that is used in SUMMONS as well as knowledge representation issues involved. Chapters 5 and 6 provide a description of the algorithms and the grammar used to generate individual sentences and present the algorithm used to combine these sentences in paragraphs using a discourse planner.

Part II introduces the concept of language reuse and regeneration (LRR), which allows SUMMONS to circumvent the need for a sophisticated natural language generation component and rather to include in the summaries text that is already in sentential form.

In Chapters 7 and 8, we present the motivation and the basics of the LRR theory and describe a case study that illustrates its usefulness — the context-based generation of noun phrase descriptions of the entities participating in the summary text.

The final portion of the thesis (**Part III**) addresses all outstanding issues related to the dissertation that are not covered elsewhere. Chapter 9 discusses the status of the components of SUMMONS and the system as a whole. A brief review of related work that is not explicitly mentioned elsewhere in the thesis is included in Chapter 10. Chapter 11 addresses some current and anticipated applications of LRR. We conclude the thesis with Chapter 12 where we summarize (no pun intended) the most important contributions of this work.

To shorten the thesis, we moved some of the information from the body of the text to five appendices (Appendix A to Appendix E). The reader can use the appendices for a better understanding of certain points in the discussion, however the thesis is complete even without the appendices.

1.9 Typographical conventions

- small caps : names of programs (e.g., SUMMONS)
- italics : linguistic examples in text (e.g., *Deng Xiaoping was born on August 22, 1904 in Paifang Village.*)
- bold face : first occurrence of a technical term (e.g., **language reuse**)
- sans serif : knowledge representation language (e.g., isa(gun,weapon))
- roman : system output (e.g., On January 12th 1990, ACAN-EFE reported that terrorists kidnapped Hector Oqueli Colindras in Guatemala City.)
- italics : mathematical and logical formulas (e.g., $\exists x, y_{x \neq y} : \mathbf{P}(\mathbf{T}[x].i_1, \mathbf{T}[y].i_2)$)

Part I

Multi-Document Summarization

The five chapters included in this part discuss the methodology for representing information about news events and for producing multi-sentence briefings from multiple articles on the same event. The culmination of this part is Chapter 5, in which we describe the pattern/action operators that we have designed to identify and express conceptual differences, similarities, and generalizations among the articles. The first three chapters in the part serve as background to Chapter 5. They introduce the knowledge engineering and representation issues which allow us to introduce the planning operators.

Chapter 2 presents an overview of our research system, SUMMONS, and relates it to other cognate systems. In Chapter 3 we describe the approach that we used to collect a corpus of news stories and to analyze them in order to extract linguistic and conceptual techniques to use in summarization. Chapter 4 introduces the domain model which we use to represent the entities and events involved in a sequence of news articles. Chapter 5 shows how the operators are used to plan paragraphs in coherent discourse to highlight the similarities, differences, or generalizations among the sources of information. In that chapter we also provide a taxonomy of the operators used to generalize information at a conceptual level and to decide how that information should be conveyed to the user from a linguistic point of view. At the end of the part, Chapter 6 describes how the individual sentences of the output of SUMMONS are produced.

Chapter 2

SUMMONS overview

2.1 Introduction

As mentioned in the previous chapter, SUMMONS produces short summaries of terrorist events. Since many information extraction systems for the terrorist domain were developed under the MUC program[MUC4, 1992], we focused on using the output format of such systems as the input to our system. This way we factor out the problems of information extraction and understanding already dealt with in previous work.

Under this assumption, the input to SUMMONS is actually a cluster of templates containing information about the perpetrator, victim, location, etc. about the stories in an event. The three stages of SUMMONS , after preprocessing, are:

1. **Information extraction** (performed by the MUC system). At this stage, information is extracted from a series of news articles on the same terrorist event and it is stored in a set of MUC templates.

Reuters reported that 18 people were killed in a bombing in Jerusalem *Sunday*. *The next day*, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel radio. Reuters reported that *at least 12 people* were *actually* killed and *105* wounded. *Later the same day*, Reuters reported that Hamas had claimed responsibility for the act.

Figure 2.1: Sample output from SUMMONS.

2. **Conceptual combination**. Different operators are tried on the set of templates until its pattern component of one of them matches the data and the corresponding action component is applied to modify the set of templates 2.2.1.

3. **Text generation**. At this stage, SUMMONS produces natural language text using FUF and SURGE on the output of the conceptual combination stage.

We will be discussing the exact interaction between the different components of SUMMONS in the following chapters. Here, we will limit ourselves to a general overview of the system and use an example as illustration. The example summary shown in Figure 2.1 is actually produced by SUMMONS (italics are added here for descriptive purposes). This paragraph summarizes four articles about two separate terrorist acts that took place in Israel in March of 1996. To create this, SUMMONS used two different planning operators.

The core of SUMMONS consists of the pipeline that starts with a cluster of templates on the same or related events and which ends with a paragraph-length summary. It is shown in Figure 2.4. The left-hand side of the figure describes how a base summary is produced while the right-hand side shows the addition of a LRR module which allows for the generation of **enhanced summaries**. We will describe each of the components of the architecture in turn and conclude by discussing what additional processing is done so that SUMMONS can function in its entirety.

The summarization component of SUMMONS is based on the traditional language generation system architecture [McKeown, 1985], also [McDonald and Pustejovsky, 1986, Hovy, 1988]. A typical language generator is divided into two main components, a **content planner**, which selects information from an underlying knowledge base to include in a text, and a **linguistic component**, which selects words to refer to concepts contained in the selected information and arranges those words, appropriately inflecting them, to form an English sentence. The content planner produces a conceptual representation of text meaning (e.g., a frame, a logical form, or an internal representation of text) and typically does not include any linguistic information. The linguistic component uses a lexicon and a grammar of English to realize the conceptual representation into a sentence. The lexicon contains the vocabulary for the system and encodes constraints about when each word can be used. As shown in Figure 2.4, the content planner used by SUMMONS determines what information from the input MUC templates should be included in the summary using a set of planning operators that are specific to summarization and to some extent, the terrorist domain. Its linguistic component determines the phrases and surface syntactic form of the summary. The linguistic component consists of a lexical chooser, which determines the high level sentence structure of each sentence and the words which realize each semantic role, and the FUF/SURGE sentence generator [Elhadad, 1991, Elhadad, 1993].

As mentioned earlier, input to SUMMONS is a set of templates, where each template represents the information extracted from one or more articles by a message understanding system. We should note that we based our implementation both on output templates from actual MUC systems and on templates that we encoded manually[1] to include terrorist events that have taken place after the period of time covered in MUC-4, such as the World Trade Center bombing, the Hebron Mosque massacre and more recent incidents in Israel and the disaster in Oklahoma City. These incidents were not handled by the original message understanding systems. We also created by hand a set of templates unrelated to real newswire articles which we used for testing some techniques of our system.

SUMMONS's summarization component generates a base summary, which contains facts extracted from the input set of articles. The base summary is later enhanced with additional facts from online structured databases with descriptions of individuals extracted from previous news to produce the enhanced summary. The additional information comes from the LRR module[2] shown on the right-hand side of Figure 2.4.

The base summary is a paragraph consisting of one or more sentences, where the length of the summary is controlled by a variable input parameter. The enhanced summary (base summary with added descriptions of entities) is generated if another input parameter (related to the user model — see Chapter 5) is set. Figure 2.2 shows an enhanced summary corresponding to the base summary shown in Figure 2.1. The enhanced summary is needed because, unlike the original documents being summarized, the generated briefing is quite devoid of details and the

[1]The MUC corpus contains only two hundred templates which we considered insufficient for our research effort.

[2]the LRR module is explained in detail in Part 2 of the thesis.

Reuters reported that 18 people were killed in a bombing in Jerusalem the capital of Israel *Sunday*. *The next day*, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel radio. Reuters reported that *at least 12 people* were *actually* killed and *105* wounded. *Later the same day*, Reuters reported that the radical Muslim group Hamas had claimed responsibility for the act.

Figure 2.2: Enhanced summary produced by SUMMONS.

addition of contextual information enhances the generated text.

2.2 SUMMONS as a text generation system

We now describe each of the components of the core SUMMONS architecture shown in Figures 2.3 and 2.4: clustering, conceptual combination, and generation of summaries.

2.2.1 Clustering

Figure 2.3 shows an important stage in the production of multi-document summaries. Since different sources write about the same event as well as about a multitude of other events, SUMMONS contains two components that address these issues. First, a MUC system is used to used to produce templates from all related articles. Articles are then grouped according to topic [Radev *et al.*, 1999] into clusters.

Each of the clusters becomes the input of the text generation module of SUMMONS (Figure 2.4). That figure is partially adapted from Michael Elhadad's thesis [Elhadad, 1993].

In addition to the conceptual operators described elsewhere, what makes

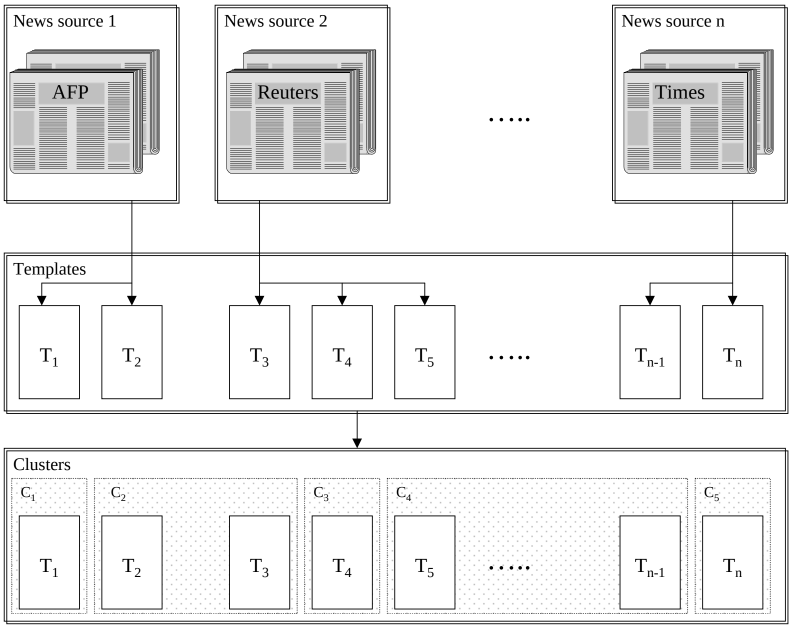

Figure 2.3: Online processing: Information extraction and clustering.

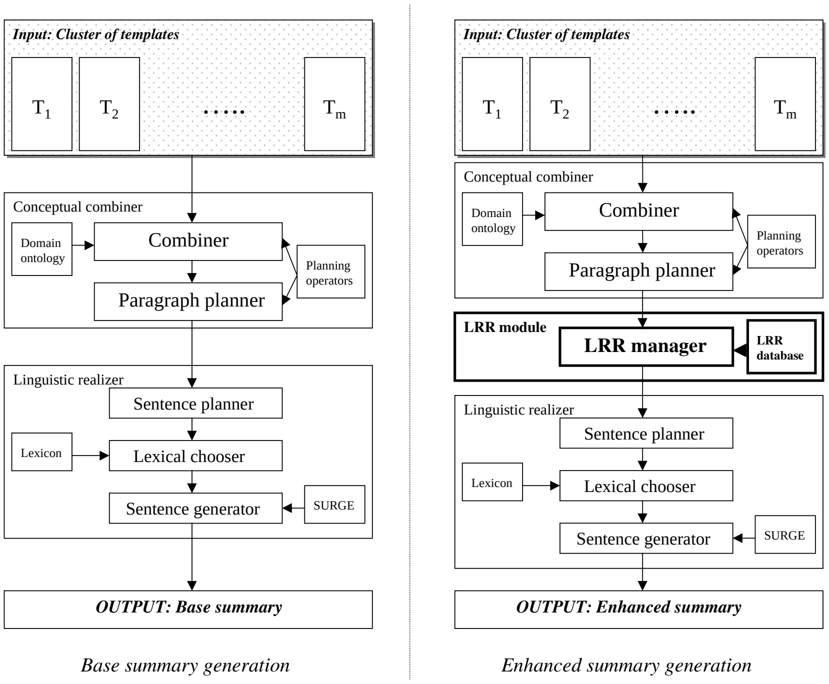

Figure 2.4: Two uses of SUMMONS in summary generation: base (left) and enhanced (right) summary generation.

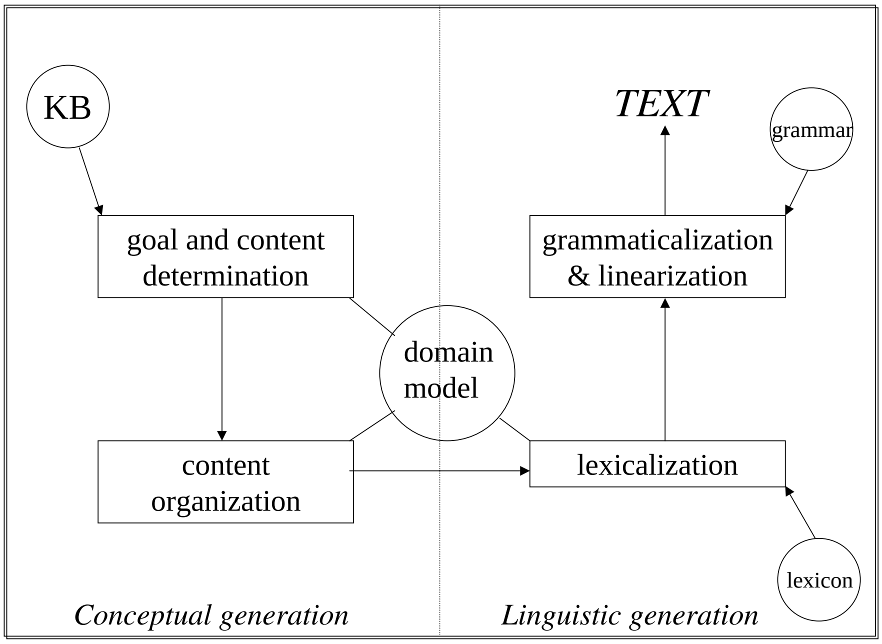

Figure 2.5: Text generation architecture (adapted from Elhadad'93).

SUMMONS significantly different is the addition of the language reuse component (on the right-hand side of Figure 2.5). The LRR component will be described fully in part II of this thesis.

2.2.2 Conceptual combination

This module consists of two components: the content combiner and the paragraph planner.

The **content combiner** uses **planning operators** Contradiction, Change of Perspective, etc., described in detail in Chapter 5 to identify which input templates exhibit the relationships. This allows the generation component to produce appropriate coherence markers (cue phrases).

The **paragraph planner** decides how information from the multiple inputs is apportioned among the sentences of the output. Chapter 5 describes this component in detail. The paragraph planner is also in charge of guiding the subsequent linguistic stages, by setting "realization switches" [McKeown *et al.*, 1995] that will eventually decide on the choice of connectives such as "the next day", "however", and "actually".

The **domain ontology** describes the relationships between entities and events in the domain of terrorism. Such information is used by the content combiner to relate, for example, a bombing in Tel Aviv with another in Jerusalem by virtue of both of them taking place in the same country. How SUMMONS represents the domain ontology is shown in Chapter 4. Some portions of the ontology are included in Appendix A.

2.2.3 Linguistic realizer

The **sentence planner** decides within each sentence how information should be realized syntactically. For example, if there is a source of information, the actual account of a terrorist event is realized as a subordinate clause, while in the absence of such a source, the account is instead realized as the main clause of the sentence.

The **lexical chooser** picks the proper words to express a given concept. It decides whether SUMMONS will generate *a bombing took place in X* or *Z bombed X* or whether the noun *explosion* will be used instead of the verb *blow up*. The constraints on the usage of alternative constructions are encoded in the **lexicon**.

As a **sentence generator** we use Michael Elhadad's reusable SURGE grammar [Elhadad, 1993]. It takes as input the output of the lexical chooser and produces linearized text.

2.3 Gathering additional information to enhance summaries

In order to build the lexical and conceptual resources that are used to generate the enhanced summaries, SUMMONS uses a range of techniques. The two types of sources (textual and structured) were mentioned in the previous chapter. SUMMONS uses a combination of language reuse and information extraction to build the three knowledge sources shown at the bottom of Figure 2.6.

The **domain model** (DM) contains information about the geography of different countries. SUMMONS builds the DM from an on-line encyclopedia [Probert, 1998] and from an on-line geographical database (The World Factbook [Agency, 1997]).

How this is done is described in Chapter 3.

The **knowledge base** (KB) is built from older news articles and consists of template representations produced by the MUC systems.

The **language reuse database** (LRDB) is built using information extraction techniques (Chapter 7). It has a phrasal lexicon [Kukich, 1983a, Jacobs, 1985] that maps a named entity (person, place, or organization) to all the noun phrases used to describe it in older news. These noun phrases are used in text generation of descriptions by SUMMONS. The LRR database is described in full detail in Part II.

We note that some of these stages take place on-line (that is, when a set of templates is ready to be summarized) while others are performed off-line (on a periodical basis, before actual summarization can take place). The off-line stage is shown in Figure 2.6.

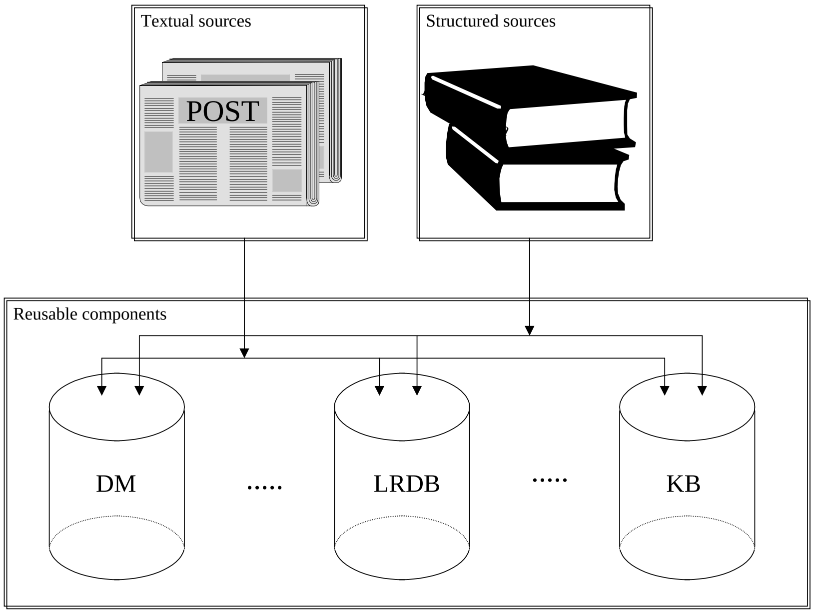

Figure 2.6: Offline processing.

Chapter 3

Data collection and corpus analysis

3.1 Introduction

Before we started work on SUMMONS, we collected phrases from corpora to empirically guide the design of the summarizer's architecture and the generation of connected text. We studied the discourse and syntactic structures used to convey information from multiple sources.

In order to produce plausible and understandable summaries, we used available on-line corpora as models, including the Wall Street Journal and current newswire and briefings from Reuters and the Agence France-Presse as well as a corpus of terrorist summaries. The corpora of news and summaries that we analyzed are 2.5 MB in size. We also manually grouped 300 articles in threads related to single events or series of similar events such as the April 1995 Oklahoma City bombing, the February 1993 bombing at the World Trade Center, and the kidnapping of the Salvadoran politician Hector Oqueli Colindras in January 1990.

From the corpora that we collected, we extracted manually, and after careful investigation, several hundred language constructions or phrases which we judged appropriate to include in the types of summaries that we want to generate. Some of the phrases are shown in Figure 3.2. In addition to the summary phrases collected from the corpus, we also tried to incorporate as many phrases as possible that have relevance to the message understanding conference domain. Due to domain variety, such phrases were scarce in the newswire corpora, forcing us to collect them from other sources (e.g., modifying templates that we acquired from the summary corpora to provide a wider coverage). Examples of such phrases include "in comparison", "finally", and "on the other hand" which are used by the paragraph planner to link sentences in smooth discourse.

Since one of the features of a briefing is conciseness, we have tried to assemble small paragraph summaries which in essence describe a single event and the change of perception of the event over time, or a series of related events with no more than a few sentences. Luckily, such summaries were available from the CSTI corpus (see next section).

3.2 Corpora used

This section explains how the corpora that we analyzed helped us to build SUMMONS. We mostly used articles from the North American News (NANTC) corpus and the BRIEF and TERROR corpora that we collected from ClariNet [ClariNet, 1998]. The Chronology of Significant Terrorist Incidents (CSTI[1]) [PGT97, 1997], which is

[1] The actual articles corresponding to the summaries were not available to us.

28 January 1993, Peru. Terrorists exploded a car bomb in front of the IBM headquarters building in Lima. Major damage was caused and eleven passersby and employees were injured. Later that day, a car bomb detonated at another Coca-Cola facility in Lima, causing only slight material damage.
4 February 1993, Egypt. A molotov cocktail bomb was lobbed at a tour bus as South Korean passengers waited to embark at a hotel outside Cairo. The Islamic extremist terrorist group Al-Gama'a al-Islamiyya claimed responsibility for the attack.
5 August 1995, Greece. A small improvised bomb detonated at a Citibank branch in Athens, causing minor damage. The Anti-Regime Nuclei (ARN) later claimed responsibility.
26 October 1996, Colombia. Leftist rebels abducted a French geologist and a Colombian engineer in Meta Department. No one claimed responsibility, but authorities suspect the National Liberation Army (ELN) or the Revolutionary Armed Forces of Colombia (FARC).
14 February 1997, Venezuela. Six armed Colombian guerrillas kidnapped a US oil engineer and his Venezuelan pilot in Apure. According to authorities, the FARC is responsible for the kidnapping.

Figure 3.1: Sample terrorist event summaries from the CSTI corpus.

a government-sponsored initiative organized at the Naval Postgraduate School, is particularly relevant. It contains a corpus of brief descriptions of terrorist events that occurred in the last few years (1993 - 62 events, 1994 - 57 events, 1995 - 81 events, 1996 - 83 events, 1997 - 95 events). We preserved the 1993 corpus for evaluation (see Chapter 9). Each event is described in a one paragraph summary, consisting of between one and nine sentences. Some examples are shown in Figure 3.1. Section 3.3 describes the paragraphs in more detail.

3.3 Analysis of the CSTI corpus

The summaries shown in Figure 3.1 follow some well defined patterns. Consider the first summary. It contains information about two events: two individual bombings in different locations of the same city. The first bombing results in serious damage to the building while the second one causes only minor damage to another facility. From a discourse point of view, each of the four sentences used are of a different well-defined type. The first sentence announces the first incident. The next two sentences are used to add detail (elaborate) on the first one. The final sentence describes the second event.

The sentences are quite stereotypical. We have been able to classify them into several categories depending on their rhetorical relation: fact, announcement, elaboration, responsibility, etc. Examples of the nine rhetorical categories are shown in Figure 3.2. All these examples are from the CSTI corpus and have also been generated by SUMMONS.

More than 95% of all sentences in the CSTI corpus fall into one of these categories. By making use of the CSTI corpus, we were able to reduce the problem of generating summary discourse to the problem of generating a sequence of such message types and by fine-tuning them to produce coherent discourse using **realization switches**. Unlike schemas [McKeown, 1985] or RST [Hovy, 1988, Marcu, 1997, Moore and Paris, 1989], the multi-sentence generation process in SUMMONS is guided primarily by the chronological order of the templates and only rarely requires more than one sentence type per input template.

Two distinctions exist between the texts in the CSTI corpus and the summaries produced by SUMMONS. The former depict one single point of view and

Message type	Example
fact	Three civilians were killed in Tegucigalpa, Honduras on Friday.
assign responsibility	Radio Venceremos reported that several heavily armed men in civilian clothes were responsible for the crime.
claim responsibility	The London-based Islamic Front for the Liberation of Bahrain claimed the bombing.
report	ACAN-EFE reported that terrorists kidnapped Hector Oqueli Colindras in Guatemala City.
total	A total of four bombings took place in Afghanistan over the last week.
denial of responsibility	The London-based Islamic Front for the Liberation of Bahrain denied responsibility for the bombing.
no responsibility	No one claimed responsibility for the attack.
elaboration	Two people were killed and three injured in the incident.
description	Sinn Fein is the political arm of the Irish Republican Army (IRA).

Figure 3.2: Examples of corpus-based message types (the nine types described in the table cover 95% of the sentences in the CSTI corpus.

describe one event, while the latter represent multiple viewpoints and may describe more than one event.

The realization switches are used by the text generation grammar to make low-level linguistic decisions such as the choice of connectives, the generation of anaphora, or the choice of active and passive voice.

Consider the third summary from Figure 3.2. It consists of two sentences: the first one presents a fact, while the second one is an adds specific information about the organization that claimed responsibility for the bombing. Upon analysis of the original stories from the TERROR corpus related to the summaries shown above, we realized that the information in the summaries often came from two, three, or more different stories. The explanation for this phenomenon is simple. When an event of terrorism happens, often the first reports only indicate its location and type, often giving little or no information about its effect, let alone its alleged perpetrators. Such information typically comes later, in follow-up stories. Sometimes a source announces that its previously released figures are no longer accurate and proceeds to update them with the most current ones. Often different news sources present complementary (or even conflicting) information. For example, the identity of an alleged perpetrator and the effect of the incident may appear in different articles.

With the above in mind, we focused our research on two main issues — combining the different accounts into one single, coherent story, and planning the discourse structure of the text that describes it. These techniques are presented in more detail in the following three chapters.

Chapter 4

The domain model

4.1 Introduction

The problem of multi-source summarization requires that the architecture of SUMMONS consider current (central) sources of information and historical (dynamically updated) sources separately.

We make the basic distinction between current, historical, and ontological sources of information.

Current sources include the text of the articles being summarized. They are always in textual form (possibly with some markup such as HTML which has to be removed during preprocessing). The current sources are the basis for generating the base summary. Knowledge representation issues related to current sources are discussed in Sections 4.3 and 4.4. An example of a current source of information would be the template representation of a news story (e.g., one that announces that a certain politician has been kidnapped).

Historical sources provide information that enhances the generated summaries. This includes both textual and structured sources. Examples of historical sources of information include a database containing the list of positions held by a particular person during the years or newswire from previous weeks and months. Historical sources of information are discussed in Section 4.5.

Ontological sources are either external or internal. **External ontological sources** are provided in a more or less structured format by an external source and can be accessed in a dynamic fashion by SUMMONS during summarization. For example, an on-line encyclopedia such as the CIA World Factbook [Agency, 1997] is an external ontological source. Since requests for information from external ontological sources are processed in real time, it is possible that the knowledge in them changes while SUMMONS is active. On the other hand, **internal ontological sources** are part of the the knowledge base of SUMMONS. Such sources include the domain ontologies provided in the MUC 4 specification — these ontologies cover the possible values of most of the slots of the MUC templates. For example, the value of the *instrument-type* slot must be from a pre-defined hierarchical list of possible weapons. Section 4.6 covers ontological sources of information.

4.2 The functional unification formalism (FUF)

This section discusses the internal representation used by SUMMONS.

4.2.1 Relationship between FUF and SUMMONS

We have chosen to implement the domain knowledge and the other information in a combination of Lisp and a functional grammar formalism.

Functional grammars present several features tailored to a knowledge-based generation system. They treat in a unified fashion the discourse, semantic, syntactic, and lexical constraints on the generation process. They don't require overcomplicated grammars and they allow for easy modularization of the linguistic code.

An excellent introduction to functional grammar can be found in [Halliday, 1985]. We are using a variant of functional grammars known as FUG (functional unification grammar), as part of an implemented package, FUF [Elhadad, 1993]. Elhadad introduces two features that made the development of SUMMONS easier - typed features and modular grammar organization.

4.2.2 Representing linguistic information in FUF

The basic unit of a FUG is the functional description (FD). FDs are used to represent both the input and the output to a grammar as well as the grammar itself.

An FD is defined as a recursive attribute-value list in which values are of one of three types:

- an atom, or
- a path, or
- another FD

An atom either indicates a linguistic value (e.g., "singular" or a lexical item such as "Al Gore". A path is a link to another portion of the grammar and is marked by curly braces (e.g., {*template incident-day*}). We have written a program that converts MUC templates to FUF FDs.

The following sections illustrate how FDs are used to represent knowledge.

4.3 Representing current information

We have used a representation scheme for news stories based on the templates already used in the MUC systems (see Figure 4.1). This representation is related to semantic case frames [Danlos, 1987] and the representation used in EPICURE [Dale, 1992]. All data extracted from the news articles is stored in a knowledge base (written in Lisp and FUF). Each story is represented as a recursive knowledge base entity (KB entity).

The MUC templates classify the semantic information extracted from a news article into five semantic groups: *message, incident, perp, phys_tgt*, and *hum_tgt*. To these we have added three more: *prim_src, sec_src*, and *now*. Thus, each story is represented in a hierarchical way in a fashion similar to the one shown in Figures 4.2– 4.5 (the remaining five sub-templates are not shown). As an illustration, the sub-template shown in Figure 4.2 contains eight slot groups:

- **message** - some meta-information about the template, such as the name of the MUC system which produced it.
- **incident** - the main facts about the incident (e.g., location, type, and date).
- **perp** - information about the perpetrator (e.g., the individual or organization perpetrator).
- **phys_tgt** - facts about the physical target of the attack (if applicable) such as its type and location.

```
;; Main MUC Template
;; Last Modified June 18, 1998
;; Dragomir R. Radev
0.  MESSAGE: ID (char)
1.  MESSAGE: TEMPLATE (int)
2.  INCIDENT: DATE (int)
3.  INCIDENT: LOCATION (char)
4.  INCIDENT: TYPE (char)
5.  INCIDENT: STAGE OF EXECUTION (char)
6.  INCIDENT: INSTRUMENT ID (char)
7.  INCIDENT: INSTRUMENT TYPE (char)
8.  PERP: INCIDENT CATEGORY (char)
9.  PERP: INDIVIDUAL ID (char)
10.  PERP: ORGANIZATION ID (char)
11.  PERP: ORGANIZATION CONFIDENCE (char)
12.  PHYS TGT: ID (char)
13.  PHYS TGT: TYPE (char)
14.  PHYS TGT: NUMBER (int)
15.  PHYS TGT: FOREIGN NATION (char)
16.  PHYS TGT: EFFECT OF INCIDENT (char)
17.  PHYS TGT: TOTAL NUMBER (int)
18.  HUM TGT: NAME (char)
19.  HUM TGT: DESCRIPTION (char)
20.  HUM TGT: TYPE (char)
21.  HUM TGT: NUMBER (int)
22.  HUM TGT: FOREIGN NATION (char)
23.  HUM TGT: EFFECT OF INCIDENT (char)
24.  HUM TGT: TOTAL NUMBER (int)
25.  PRIM SRC: SOURCE (char)
26.  PRIM SRC: REPORT (char)
27.  PRIM SRC: TIME (char)
28.  PRIM SRC: DATE (int)
29.  PRIM SRC: DAY (char)
30.  PRIM SRC: MONTH (char)
31.  PRIM SRC: YEAR (int)
32.  SEC SRC: SOURCE (char)
33.  SEC SRC: REPORT (char)
34.  SEC SRC: TIME (char)
35.  SEC SRC: DATE (int)
36.  SEC SRC: DAY (char)
37.  SEC SRC: MONTH (char)
38.  SEC SRC: YEAR (int)
39.  INCIDENT: TIME (char)
40.  INCIDENT: DAY (char)
41.  INCIDENT: MONTH (char)
42.  INCIDENT: YEAR (int)
43.  NOW: TIME (char)
44.  NOW: DATE (int)
45.  NOW: DAY (char)
46.  NOW: MONTH (char)
47.  NOW: YEAR (int)
```

Figure 4.1: Blank MUC-4 Template, extended to include source information as well as the current date and time.

- **hum_tgt** - information about the human victims (if any) - name, type, number, etc.
- **prim_src** - the primary source of the article.
- **sec_src** - the secondary source of the article.
- **now** - the current date and time.

The last three slot groups are not part of the original MUC templates but were added during the development of SUMMONS for completeness.

The possible values for the different slots are described in the MUC instructions [MUC4, 1992]. Since we added three additional semantic groups related to the sources of information (*prim_src* and *sec_src*) as well as the current date (*now*), we defined the scope of the potential fill values ourselves.

Using terminology from [Dale, 1992], some of the slots in the templates can be filled with *events* (e.g., incident types), others with *entities* (such as people, locations or sources of information) or *temporal structures* (e.g., the date of the incident, the date of the primary or the secondary reports, and the current date).

We enriched the templates by adding four slots: the primary source, the secondary source, and the times at which both sources made their reports[1]. The source of the report is essential for discovering and reporting contradictions and generalizations, because often different reports of an event are in conflict. Also, source information can indicate the level of confidence of the report, particularly when reported information changes over time. For example, if several secondary

[1]The primary source is usually a direct witness of the event, while the secondary source is most often a press agency or journalist, reporting the event.

message	*message*
incident	*incident*
perp	*perp*
phys_tgt	*phys_tgt*
hum_tgt	*hum_tgt*
prim_src	*prim_src*
sec_src	*sec_src*
now	*now*

Figure 4.2: Top-level KB entity including all eight sub-templates used. Each news article is represented as such a KB entity.

id	"vehicle"
type	"other: vehicle"
number	1
foreign_nation	""
effect_of_incident	"destroyed: vehicle"
total_number	1

Figure 4.3: KB entity corresponding to the *phys_tgt* sub-template.

sources all report the same facts for a single event, citing multiple primary sources, it is more likely that this is the way the event really happened, while if there are many contradictions between reports, it is likely that the facts are not yet fully known.

4.4 Representing clusters of stories

Clusters of stories on the same or different events are described by lists of KB templates, called **lots**. An example of a KB representation of a sequence of these stories

$$\begin{bmatrix} \text{incident\_category} & \text{“terrorist act”} \\ \text{individual\_id} & \text{“urban guerrillas”} \\ \text{organization\_id} & \text{“Nationalist Republican Alliance”} \\ \text{organization\_confidence} & \text{“suspected or accused: National Republican Alliance”} \end{bmatrix}$$

Figure 4.4: KB entity corresponding to the *perp* sub-template.

$$\begin{bmatrix} \text{date} & \begin{bmatrix} \text{day} & 1 \\ \text{month} & \text{“June”} \\ \text{year} & 1988 \end{bmatrix} \\ \text{location} & \text{“El Salvador: San Salvador (Department)”} \\ \text{type} & \text{“attack”} \\ \text{stage\_of\_execution} & \text{“accomplished”} \\ \text{instrument\_id} & \text{“”} \\ \text{instrument\_type} & \text{“”} \end{bmatrix}$$

Figure 4.5: KB entity corresponding to the *incident* sub-template.

$$\begin{bmatrix} \text{incident-type} & \text{“killing”} \\ \text{incident-location} & \text{“Sri Lanka’s main business district”} \\ \text{hum\_tgt-number} & 20 \\ \text{hum\_tgt-description} & \text{“person”} \\ \text{sec\_src-report} & \text{“report”} \\ \text{sec\_src-date} & 15 \\ \text{sec\_src-year} & 1997 \\ \text{sec\_src-month} & \text{“October”} \\ \text{sec\_src-source} & \text{“Agence France-Presse”} \end{bmatrix}$$

Figure 4.6: Template 1.

$$\begin{bmatrix} \text{template} & \begin{bmatrix} \text{incident-type} & \text{“killing”} \\ \text{hum\_tgt-number} & 9 \\ \text{hum\_tgt-description} & \text{“person”} \\ \text{sec\_src-report} & \text{“announce”} \\ \text{sec\_src-source} & \text{“AFP”} \end{bmatrix} \end{bmatrix}$$

Figure 4.7: Template 2.

is shown in Figures 4.6, 4.7, and 4.8. The first template reports that according to Agence France-Presse, 20 people are reported killed in Sri Lanka's main business district on October 15, 1997. The second template shows that according to the same source (AFP), nine people are killed in the terrorist act. The additional slots (the "meta" sub-template) are added by the planning operators in order to express the change of perspective (20 people becomes nine people).

Definition 2 *: A* **lot** *is a list of templates (or FDs) on the same or related events, sorted in chronological order.*

$$\begin{bmatrix} \text{template} & \begin{bmatrix} \text{incident-type} & \text{``killing''} \\ \text{hum\_tgt-number} & 9 \\ \text{hum\_tgt-description} & \text{``person''} \\ \text{sec\_src-report} & \text{``announce''} \\ \text{sec\_src-source} & \text{``AFP''} \end{bmatrix} \\ \text{meta} & \begin{bmatrix} \text{sec\_src-day} & \text{``later''} \\ \text{hum\_tgt-number} & \begin{bmatrix} \text{classifier} & \text{``exactly''} \end{bmatrix} \\ \text{incident-type} & \begin{bmatrix} \text{classifier} & \text{``actually''} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 4.8: Template 2 after realization switches have been added.

Instead of producing separate sentences for each individual template in the lot, SUMMONS uses the planning operators (already mentioned, but which will be described in detail in Chapter 5). The operators look for patterns in the input templates and rearrange them to produce appropriate text.

Operators influence linguistic generation by setting up the realization switches. They are added to the templates as a separate sub-template ("meta") — see Figure 4.7. In that example, three realization switches (*sec_src-day*, *hum_tgt-number* and *incident-type*) are set. The value of the *sec_src-day* realization switch is set by the planning operator because the date and the time of the second template are chronologically after the ones in the first template. The realization switches are used to help the generation component select appropriate discourse phrases. Other types of realization switches decide whether an absolute or relative date will be generated; whether the sentence should be in active or passive voice; whether to include additional modifiers to certain template slots (e.g., "*another* five people" rather than "five people" only).

```
(setf (gethash ``Jacques Chirac'' *profile*)
  '(
    (``mayor of Paris'' ``UPI'' ``31-Dec-1995'')
    (``president of France'' ``Reuters'' ``01-Jan-1996'')))
```

Figure 4.9: Sample KB entity for a description of an entity.

4.5 Representing historical information

Similar to the topical event information, we use Lisp code (as well as FUF) to represent historical information such as descriptions of entities (see Figure 4.9 and Appendix D). Information from these KB entities is also added to the summarization process through operators. In the example, the noun phrase descriptions of a named entity are stored along with the dates when they were extracted.

A sample KB entity related to a description of an entity is shown in Figure 4.9.

4.6 Representing ontological information

Ontological information about the domain of terrorism is also represented using the FUF formalism. One of the features of FUF is the **feature type** [Elhadad, 1993] which we have used to represent isa relationships, such as isa(gun, weapon). Figure 4.10 shows a sample ontology for the possible values of the instrument-type slot of the template. ISA ontologies are used in generalizations when applying planning operators (see Chapter 5).

Other ontologies used in SUMMONS include the ontology of geographical locations (cities, regions, countries, etc.) from the World Factbook (see Figure 4.11)

```
(define-feature-type weapon (gun explosive))
(define-feature-type gun (mortar machine_gun))
(define-feature-type explosive (bomb grenade))
(define-feature-type bomb (vehicle_bomb mine))
```

Figure 4.10: Ontology corresponding to the instrument-type slot.

and the ontology of incident types (kidnapings, killings, bombings, etc.). For a full description of the specification of the MUC4 domain, we refer the reader to [MUC4, 1992] and [Sundheim, 1992].

```
(country ((name "El Salvador")
          (capital "San Salvador")
          (map ((url
                 "http://www.odci.gov/cia/publications/95fact/es.gif")))
          (type republic)
          (divisions ((name "department")
                      (number 14)
                      (list ("Ahuachapan"
                             "Cabanas"
                             "Chalatenango"
                             "Cuscatlan"
                             "La Libertad"
                             ...
                             "Usulutan"))))
          (ports ((list ("Acajutla"
                         "Puerto Cutuco"
                         "La Libertad"
                         "La Union"
                         "Puerto El Triunfo"))))
          (executive ((president ((name "Armando CALDERON SOL")
                                  (elected "010694")))
                      (vice-president ((name "Enrique BORGO Bustamante")
                                       (elected "unknown")))))))
```

Figure 4.11: FD representation of the worldbook entry for El Salvador.

Chapter 5

Multiple document summarization

5.1 Introduction

Chapter 5 presents the core of the work on multi-document summarization. This chapter describes how information from multiple documents is combined together depending on whether certain types of logical relationships (such as Agreement and Contradiction) exist among the input templates. SUMMONS produces text based on the templates of all the input articles and identifies logical relationships by selecting appropriate wording and marking discourse structure.

The focal point of multi-document summarization is the notion of a **planning operator**. A planning operator serves two purposes: to identify the logical relationships among templates in a *lot* by comparing the templates, and to ensure that the generated text will both be grammatical and contain the appropriate information from the inputs. Thus, the planning operators work at both the conceptual and lexical levels of text generation. We first give an example of their use in discourse generation and then focus on the way in which they are used to conceptualize

```
if x was mentioned in the previous sentence
then use a pronoun
else if x is in D then use a definite noun phrase
else use an indefinite noun phrase
```

Figure 5.1: Pronoun generation algorithm described in Dale'92 (D is the list of already defined concepts).

```
if x was the date in the previous sentence
and y is the date in the current sentence
and x is the day before y
then use on the next day instead of y
else use y
```

Figure 5.2: One possible algorithm for the generation of date expressions.

and realize information from multiple sources.

5.2 Examples of discourse algorithms and motivation for operators

Since SUMMONS generates multi-sentence text, it needs to make a large number of discourse-generation decisions. For example, to generate an expression to refer to an object, it could use the algorithm shown in Figure 5.1, while to generate an expression that refers to a date, it could use the algorithm in Figure 5.2. A similar algorithm for locations is shown in Figure 5.3. We name these two sample algorithms the **date algorithm** and the **location algorithm**, since we will refer back to them in the following two sections.

A similar algorithm can be designed for the generation of referring expressions for organizations, events, etc. We found it counter-productive to express a

```
if (isa (x, city)) and x was the location in the previous sentence
and x is also the location in the current sentence
then use in the same city
else use y
```

Figure 5.3: A possible algorithm for the generation of location expressions.

large number of similar algorithms using a procedural approach. We opted for a declarative method in which similar algorithms are expressed as **discourse operators**. In the next two sections, we formally describe the concept of discourse operators and show how the two algorithms (location and date) can be expressed in the form of operators.

An operator $\mathbf{O}$ is a tuple $(\mathbf{I}, \mathbf{A})$, where $\mathbf{I}$ is an **input condition** and $\mathbf{A}$ is an **action**. It is applied on a list of templates (or *lot*) $\mathbf{L}$. Whenever the predicate $\mathbf{I}(\mathbf{L})$ holds, a new version of the lot, $\mathbf{L}'$, is created, by performing the transformations described in $\mathbf{A}$ on $\mathbf{L}$. Otherwise, $\mathbf{L}$ remains unchanged:

$$\mathbf{L}' \leftarrow \begin{cases} \mathbf{A}(\mathbf{L}) & \text{if } \mathbf{I} \text{ holds,} \\ \mathbf{L} & \text{otherwise.} \end{cases} \tag{5.1}$$

We distinguish between two types of operators: *minimal* and *universal*. The **minimal operator** applies on a single pair of templates (x, y) in $\mathbf{L}$ for which $\mathbf{I}$ holds (normally, the pair for which x is minimal, and in case of ties, for which both x and y are minimal). The **universal operator** is applied on all pairs (x, y) for which $\mathbf{I}$ holds.

As an example of a transformation, the application of $\mathbf{A}$ can consist of inserting the FD f in the path j of the n^{th} template of $\mathbf{L}$:

$$\mathbf{L}[n].j = f, \tag{5.2}$$

Actual examples of operators are shown in the next section.

The idea behind operators is that we can present conceptual multi-document summarization in SUMMONS as a pipeline of operators applied on the input lot $\mathbf{L}$ and send the output of the pipeline, $\mathbf{L}'$ to the surface generator.

$$\mathbf{L}' = O_k(...(O_2(O_1(\mathbf{L})))) \tag{5.3}$$

5.3 Generic planning operator

We consider a simple example of an operator. Let $\mathbf{L}$ consist of two FDs, numbered 1 and 2. We want to compare the values of the {*template incident-day*} slots of the two FDs. If the second day is the next day after the first one, we want to force the use of *the next day* or some equivalent phrase in the second FD.

A sample operator that can be used to address the problem of generation of incident days is shown in Figure 5.4.

The input condition $\mathbf{I}$ will hold whenever the (prev-day {template incident-day} {template incident-day}) predicate holds, that is when the incident day of the second template is the day after the incident day of the first template.

When $\mathbf{I}$ holds, the action, $\mathbf{A}$ is executed. In this case, the action is to insert a value in the FD of the second template : {meta incident-day} becomes "on the next day". Since the generation grammar looks at the "meta" values before looking

```
(def-operator
  consecutive-days

  "two consecutive days"

  ((condition
    (prev-day
   {template incident-day}
   {template incident-day}))

   (action
    ("right"
     {meta incident-day}
     "on the next day"))

   (type
   "minimal")
   ))
```

Figure 5.4: Sample operator.

at the "template" values, it will generate "on the next day" instead of the day that originally appears in the template. A more elaborate example of a planning operator will be shown in the following section.

5.4 Taxonomy of planning operators

The main point of departure for SUMMONS from previous work in text generation is in the stage of identifying what information to include and how to group it together. In PLANDOC [McKeown *et al.*, 1995], input templates are very similar to the MUC templates used by SUMMONS. Hence one of the main problems for PLANDOC is to form a grouping that puts the most similar items together, allowing the use of conjunction and ellipsis to delete repetitive material. This task is called *aggregation* [Dalianis and Hovy, 1993, Shaw, 1998]. For summarizing multiple news articles, the task is almost the opposite — we need to find the differences from one article to the next, to identify how the reported facts have changed. Thus, one of our main problems was to identify summarization strategies which indicate how information is linked together to form a concise and cohesive summary. As was found in other work [Robin, 1994], what information is included is often dependent on the language available to make concise additions. Thus, using a corpus of summaries was critical to identifying the different possible summaries.

We have developed a set of planning operators derived from the corpora that we analyzed (Chapter 3). These operators determine what types of simple sentences constitute a summary, in what order they need to be listed, and the ways in which simple sentences should be combined into more complex ones. In addition, we have specified which summarization-specific phrases are to be included in different types of summaries.

We start with a list of templates **L** and subsequently apply different operators $O_1, O_2, ..., O_n$ on it until no more operators can be applied (see Section 5.5). At each step, a summary operator is selected based on existing similarities between articles in the database. This operator is then applied to the input templates, resulting in a new template which combines, or synthesizes, information from the old. Each operator is independent of the others and several can be applied in succession to the input templates. Each of the seven major operators is further subdivided to cover various modifications to its input. This procedure is similar to the sentence planner of [Wanner and Hovy, 1996].

A summary operator encodes a means for linking information in two different templates. Often it results in the synthesis of new information. For example, a generalization may be formed from two independent facts. Alternatively, since we are summarizing reports written over time, highlighting how knowledge of the event changed is important; and therefore, summaries sometimes must identify differences between reports. A description of the operators we implemented in SUMMONS follows, accompanied by an example of system output for each operator. More complex summaries can be produced by applying multiple operators on the same input, as shown in the example in Section 5.6. The rest of this section describes the classes of operators implemented in SUMMONS.

5.4.1 Change of perspective

When an initial report gets a fact wrong or has incomplete information, the change is usually included in the summary. In order for the "change of perspective" operator to apply, the SOURCE field for the two templates must be the same, while the value of some other field is different in the two templates. For example, if the number of victims changes, we know that the first report was *incorrect* if the number goes down, while the source had *incomplete information* (or additional people died) if the number goes up. The first two sentences from the following example were generated using the "change of perspective" operator. The initial estimate of *at least 10 people* killed in the incident becomes *at least 12 people*. Similarly, the change in the number of wounded people is also reported.

> **Example:** March 4th, Reuters reported that a bomb in Tel Aviv killed at least 10 people and wounded 30. *Later the same day*, Reuters reported that *exactly 12 people* were *actually* killed and *105* wounded.

The corresponding operator is shown in Figure 5.5. Note that the keyword "right" refers to the second chronologically of the two templates (similarly, "left" is used to refer to the earlier of the two).

5.4.2 Contradiction

When two sources report conflicting information about the same event (e.g., a different number of victims or a different perpetrator), a contradiction arises. In the absence of values indicating the reliability of the sources, a summary cannot report either of them as true, but can indicate that the facts are not clear. A summary might indicate that one of the sources determined that 20 people were killed, while the other source determined that only 5 were indeed killed. The difference between this example and the previous one on "change of perspective" is the source of the

```
(def-operator
  change-of-perspective-1

  "change of perspective 1"

  ((condition
    (and
      (prev-day
       {template sec_src-day}
       {template sec_src-day})
      (less
       {template hum_tgt-number}
       {template hum_tgt-number})))

   (action
    (and
     ("right"
      {meta incident-type classifier}
      "actually")
     ("right"
      {meta sec_src day}
      "on the next day")
     ("right"
      {meta hum_tgt-number classifier}
      "exactly")))

   (type
   "minimal")
   ))
```

Figure 5.5: Change of Perspective operator.

update. If the same source announces a change, then we know that it is reporting a change of perspective. Otherwise, an additional source presents information which is not necessarily more correct than the information presented by the earlier source and we can therefore conclude that we have a contradiction.

> **Example:** The afternoon of February 26, 1993, Reuters reported that a suspected bomb killed *at least six* people in the World Trade Center. *However*, Associated Press announced that *exactly five* people were killed in the blast.

5.4.3 Addition/Elaboration

When a subsequent report indicates that additional facts (e.g., the identity of a perpetrator or the number of victims) are known, this is reported in the summary using an elaboration sentence. Additional results of the event may occur after the initial report or additional information may become known. The operator determines this by the way the value of a template slot changes. Since the former template doesn't contain a value for the perpetrator slot and the latter contains information about claimed responsibility, we can apply the addition operator.

> **Example:** On Monday, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel radio. *Later the same day*, Reuters reported that *the radical Muslim group* Hamas had claimed responsibility for the act.

5.4.4 Refinement

Often, a more general piece of information may be refined in subsequent reports. Thus, if an event is originally reported to have occurred in New York City, the location might later be specified as a particular borough of the city. Similarly, if a terrorist group is identified as Palestinian, later the exact name of the terrorist group may be determined. Unlike the previous example, there was a value for the perpetrator slot in the first template, while the second one further elaborates on it, specifying the perpetrator more specifically.

> **Example:** On Monday, Reuters announced that *a suicide bomber* killed at least 10 people in Tel Aviv. *Later the same day*, Reuters reported that *Hamas* claimed responsibility for the bombing.

5.4.5 Agreement

If two sources have the same values for a specific slot, this will heighten the reader's confidence in their veracity and thus, agreement between sources is caught and reported by SUMMONS.

> **Example:** The morning of March 1st 1994, UPI reported that a man was kidnapped in the Bronx. *Later*, this was confirmed by Reuters.

5.4.6 Superset/Generalization

If the same event is reported from different sources and all of them have incomplete information, it is possible to combine information from them to produce a more complete summary. This operator is also used to aggregate multiple events as shown in the example. Generalizations are based on the domain model described in Chapter 4 and Appendix A.

> **Example:** Reuters reported that 18 people were killed in a Jerusalem bombing *Sunday. The next day*, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel radio. *A total of at least 28 people* were killed in the *two terrorist acts in Israel over the last two days.*

5.4.7 Other types of operators

While only the six classes of operators (with a total of 51 actual operators) identified above are implemented in SUMMONS, we should note that the declarative approach allows for additional operators to be added. We have thought of at least two additional operator types that could be added: "trend" and "no information".

There is a trend if two or more articles reflect similar patterns over time. Thus, we might notice that three consecutive bombings occurred at the same location and summarize them into a single sentence.

> **Example:** This is the third terrorist act committed by Hamas in four weeks.

Since we are interested in conveying information about the primary and secondary source of a certain piece of news, which are generally trusted sources of information, we ought to pay attention also to the lack of information from a certain source when such is expected to be present. For example, it might be the case that a certain news agency reports a terrorist act in a given country, but the authorities

Algorithm 1 Applying all planning operators.

sort the list of templates **L** in chronological order.
repeat
  scan through the database of operators until one is found that matches the current elements of **L**
  if a matching operator **O** is found **then**
    apply its action **A** to **L** to produce **L**′
  end if
until no more operators can be applied
send the current version of **L** to the linguistic component.

of that country don't give out any information. Since there is an infinite number of sources which might not confirm a given fact (or the system will not have access to the appropriate templates), we have included this operator only as an illustration of a concept which further highlights the domain-specificity of the system.

> **Example:** Two bombs exploded in Baghdad, Iraqi dissidents reported Friday. There was *no confirmation* of the incidents by the Iraqi National Congress.

5.5 Algorithm for applying operators in sequence

The previous section described the types of operators already implemented in SUMMONS. In Section 5.2 we mentioned that in order to produce a summary, a sequence of operators is applied on the input. We experimented with several algorithms [Radev and McKeown, 1998] for deciding the order in which they are applied until we finally settled on the greedy algorithm described in Algorithm 2.

5.6 Example

This section describes how the algorithm is applied to a set of 4 templates by tracing the computational process that transforms the raw source into a final natural language summary. Excerpts from the four input news articles are shown in Figure 5.6.

The four news articles are transformed into four templates which correspond to four separate accounts of two related events and will be included in the set of templates from which the template combiner will work. Only the relevant fields are shown.

Let's now consider the four templates in the order that they appear in the list of templates. These templates are shown in Figures 5.7 – 5.10. They are generated manually from the input newswire texts. Information about the primary and secondary sources of information is added. The differences in the two templates (which will trigger certain operators) are shown in **bold face**. The summary generated by the system is shown in Figure 5.11.

The first two sentences are generated from template one. The subsequent sentences are generated using different operators which are triggered according to changing values for certain attributes in the three remaining templates.

As previous templates didn't contain information about the perpetrator, SUMMONS applies the Refinement operator to generate the fourth sentence. Sentence three is generated using the Change of perspective operator, as the number of victims reported in messages two and three is different.

The description for Hamas (*radical Muslim group*) was added by the extraction generator (see Chapter 8). Typically, a description is included in the source

Article 1: JERUSALEM - A Muslim suicide bomber blew apart 18 people on a Jerusalem bus and wounded 10 in a mirror-image of an attack one week ago. The carnage by Hamas could rob Israel's Prime Minister Shimon Peres of the May 29 election victory he needs to pursue Middle East peacemaking. Peres declared all-out war on Hamas but his tough talk did little to impress stunned residents of Jerusalem who said the election would turn on the issue of personal security.

Article 2: JERUSALEM - A bomb at a busy Tel Aviv shopping mall killed at least 10 people and wounded 30, Israel radio said quoting police. Army radio said the blast was apparently caused by a suicide bomber. Police said there were many wounded.

Article 3: A bomb blast ripped through the commercial heart of Tel Aviv Monday, killing at least 13 people and wounding more than 100. Israeli police say an Islamic suicide bomber blew himself up outside a crowded shopping mall. It was the fourth deadly bombing in Israel in nine days. The Islamic fundamentalist group Hamas claimed responsibility for the attacks, which have killed at least 54 people. Hamas is intent on stopping the Middle East peace process. President Clinton joined the voices of international condemnation after the latest attack. He said the "forces of terror shall not triumph" over peacemaking efforts.

Article 4: TEL AVIV (Reuters) - A Muslim suicide bomber killed at least 12 people and wounded 105, including children, outside a crowded Tel Aviv shopping mall Monday, police said. Sunday, a Hamas suicide bomber killed 18 people on a Jerusalem bus. Hamas has now killed at least 56 people in four attacks in nine days. The windows of stores lining both sides of Dizengoff Street were shattered, the charred skeletons of cars lay in the street, the sidewalks were strewn with blood. The last attack on Dizengoff was in October 1994 when a Hamas suicide bomber killed 22 people on a bus.

Figure 5.6: Fragments of input articles 1–4.

MESSAGE: ID	TST-REU-0001
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 3, 1996 11:30**
PRIMSOURCE: SOURCE	
INCIDENT: DATE	**March 3, 1996**
INCIDENT: LOCATION	**Jerusalem**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: 18”**
	“wounded: 10”
PERP: ORGANIZATION ID	

Figure 5.7: Template for article one.

MESSAGE: ID	TST-REU-0002
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 4, 1996 07:20**
PRIMSOURCE: SOURCE	**Israel Radio**
INCIDENT: DATE	**March 4, 1996**
INCIDENT: LOCATION	**Tel Aviv**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: at least 10”**
	“wounded: 30”
PERP: ORGANIZATION ID	

Figure 5.8: Template for article two.

MESSAGE: ID	TST-REU-0003
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 4, 1996 14:20**
PRIMSOURCE: SOURCE	
INCIDENT: DATE	**March 4, 1996**
INCIDENT: LOCATION	**Tel Aviv**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**“killed: at least 13”**
	“wounded: more than 100”
PERP: ORGANIZATION ID	**“Hamas”**

Figure 5.9: Template for article three.

MESSAGE: ID	TST-REU-0004
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	**March 4, 1996 14:30**
PRIMSOURCE: SOURCE	
INCIDENT: DATE	**March 4, 1996**
INCIDENT: LOCATION	**Tel Aviv**
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	**"killed: at least 12"**
	"wounded: 105"
PERP: ORGANIZATION ID	**"Hamas"**

Figure 5.10: Template for article four.

Reuters reported that 18 people were killed in a Jerusalem bombing *Sunday*. *The next day*, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel radio. Reuters reported that *at least 12 people* were killed and *105* wounded. *Later the same day*, Reuters reported that *the radical Muslim group* Hamas had claimed responsibility for the act.

Figure 5.11: SUMMONS output based on the four articles.

text and should be extracted by the message understanding system. In the cases in which a description doesn't appear or is not extracted, SUMMONS generates a description from the database of extracted descriptions.

5.7 Web-based interface

The current interface of SUMMONS is accessible through the Web. A screen snapshot is shown in Figure 5.12. It should be noted that the input is a set of templates, not a set of articles; that is, there is no live connection to a MUC system.

The user interface consists of three panels. Panel 1 (the left panel) displays the SUMMONS logo and provides links to SUMMONS's on-line help. Panel 2 (the top panel) is used to select the host and port number of the SUMMONS server as well as the number of templates that will be processed to produce the summary.

Panel 3 (in the center) displays a range of blank templates. The user may either fill in these templates with MUC-style data or can choose such data to be extracted from an external server. Slots for which the values are known to be blank should be left blank. Other possible values for the slots are available. One of them is the special symbol *REPEAT* which indicates that the value for the slot in the current template is the same as the value in the previous template. Another special value can be the indication of *no victims* as prescribed in the MUC guidelines.

Panel 2 automatically includes current values for the *now* sub-template (such as the current date and time), however the user is free to change them. A number of user-modifiable options appear at the bottom of Panel 2:

- SUMMONS **host** - the machine where the SUMMONS server is running.

- SUMMONS **port** - the port number of the server.
- **summary length** - one of: short, medium, or long.
- **use operators** - if selected, SUMMONS applies discourse operators on the templates before generating text. Otherwise, text is generated from each template separately.
- **use descriptions** - if checked, SUMMONS uses contextually-relevant noun phrases to describe the entities in the generated text.
- **description sources** - the sources of descriptions that should be used if the "use descriptions" options is also selected.
- **add hyperlinks** - if checked, SUMMONS adds hyperlinks from all participating entities to the corresponding entries in PROFILE.

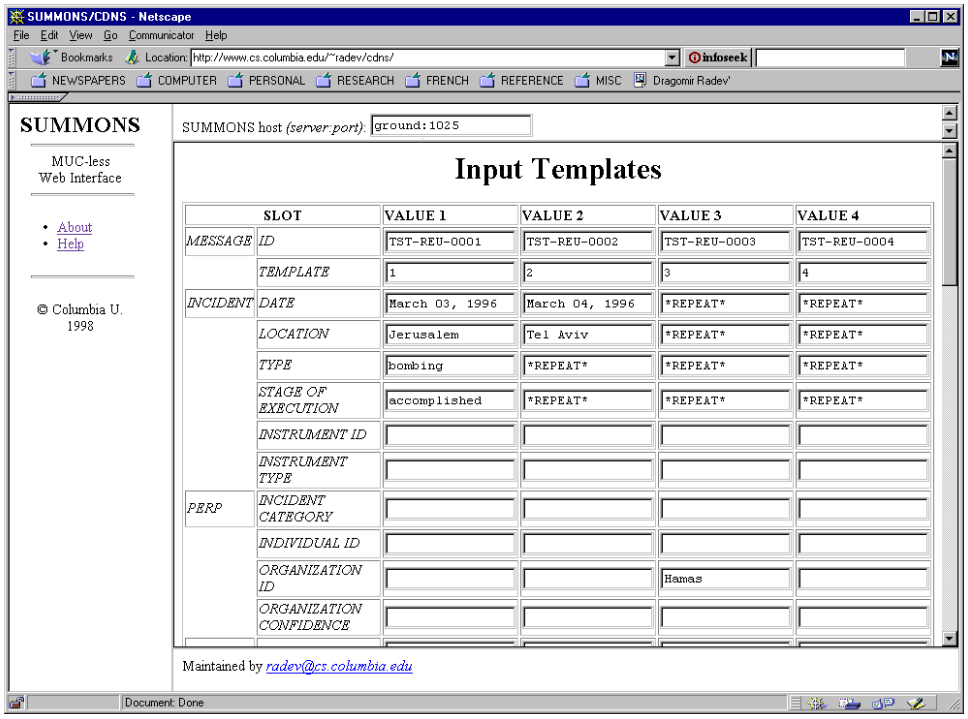

Figure 5.12: SUMMONS Web-based interface.

Chapter 6

Generation of single sentences

6.1 Introduction

While in Chapter 5 we explained how a paragraph is planned, here we describe how the individual sentences that compose a paragraph are generated using the FUF/SURGE package[Elhadad, 1993].

6.2 Sentence generation

To generate a sentence in the base summary, SUMMONS goes through three stages:

- it converts the template data into a semantic structure (in FD format) that can be processed in FUF.
- it unifies the result with its sentence-level grammar (placing appropriate constraints on adjuncts and embedded clauses among others) by mapping one set of FDs into another set. At this stage, lexical roles are filled in.

```
(((template
   ((perp-individual_id "terrorists")
    (incident-location "Guatemala City")
    (incident-type "kidnapping")
    (hum_tgt-name "Hector Oqueli Colindras")
    (sec_src-date 12)
    (sec_src-year 1990)
    (sec_src-month "January")
    (sec_src-report "report")
    (sec_src-source "ACAN-EFE")))
  ))
```

Figure 6.1: MUC template in FUF format (stage 1).

- it generates text using the SURGE grammar.

The rest of this section shows how a sample template is converted into an English sentence by going through the three stages described above.

Figure 6.1 shows the input template related to the 1990 kidnapping of Salvadoran politician Hector Oqueli Colindras in Guatemala. This template has been extracted from the MUC corpus and converted (automatically) into FUF FD format.

After structuring, the MUC template is converted into a hierarchical form — the eight semantic groups (corresponding to the eight categories of slots used in MUC, such as Incident, Perpetrator, Human Target, etc.) are thus created. The output of the structuring stage is shown in Figure 6.2.

Figure 6.3 shows the knowledge base entity corresponding to the template from Figure 6.2.

After unification with the SUMMONS grammar, the semantic roles are mapped into corresponding lexical roles. The output is shown in Figure 6.4 (FD format).

```
((sem
  ((incident ((location "Guatemala_City")
             (type "kidnapping")))
   (perp ((individual_id "terrorists")))
   (hum_tgt ((name "Hector Oqueli Colindras")))
   (sec_src ((source "ACAN-EFE")
            (report "report")
            (date 12)
            (month "January")
            (year 1990))))
  )))
```

Figure 6.2: MUC template in FUF format after structuring (stage 2).

$$\left[\begin{array}{ll} \text{sem} & \left[\begin{array}{ll} \text{incident} & \left[\begin{array}{ll} \text{location} & \text{“Guatemala City”} \\ \text{type} & \text{“kidnapping”} \end{array}\right] \\ \text{perp} & \left[\begin{array}{ll} \text{individual\_id} & \text{“terrorists”} \end{array}\right] \\ \text{hum\_tgt} & \left[\begin{array}{ll} \text{name} & \text{“Hector Oqueli Colindras”} \end{array}\right] \\ \text{sec\_src} & \left[\begin{array}{ll} \text{source} & \text{“ACAN-EFE”} \\ \text{report} & \text{“report”} \\ \text{date} & 12 \\ \text{month} & \text{“January”} \\ \text{year} & 1990 \end{array}\right] \end{array}\right] \end{array}\right]$$

Figure 6.3: KB representation of the template.

$$
\left[\begin{array}{ll}
\text{sem} & \left[\begin{array}{ll}
\text{incident} & \left[\begin{array}{ll} \text{location} & \text{[1] ``Guatemala City''} \\ \text{type} & \text{[2] ``kidnapping''} \end{array}\right] \\
\text{perp} & \left[\begin{array}{ll} \text{individual\_id} & \text{[3] ``terrorists''} \end{array}\right] \\
\text{hum\_tgt} & \left[\begin{array}{ll} \text{name} & \text{[4] ``Hector Oqueli Colindras''} \end{array}\right] \\
\text{sec\_src} & \left[\begin{array}{ll} \text{source} & \text{[5] ``ACAN-EFE''} \\ \text{report} & \text{``report''} \\ \text{date} & 12 \\ \text{month} & \text{``January''} \\ \text{year} & 1990 \end{array}\right]
\end{array}\right] \\
\text{cat} & \text{clause} \\
\text{obl} & \left[\begin{array}{ll}
\text{source} & \left[\begin{array}{ll} \text{report} & \text{``report''} \end{array}\right] \\
\text{stage} & \text{``accomplished''} \\
\text{srctime} & \left[\begin{array}{ll} \text{vprep} & \text{[6] ``on''} \\ \text{obldate} & \left[\begin{array}{ll} \text{cat} & \text{date} \end{array}\right] \end{array}\right] \\
\text{target} & \text{[7]} \left[\begin{array}{ll} \text{cat} & \text{[8]proper} \\ \text{lex} & \text{[4]} \end{array}\right] \\
\text{incident} & \left[\begin{array}{ll} \text{dlex-incident-type-in} & \text{[2]} \\ \text{vlex} & \text{[9] ``kidnap''} \end{array}\right] \\
\text{perpetrator} & \text{[10]} \left[\begin{array}{ll} \text{cat} & \text{[11]np} \\ \text{head} & \text{[12]} \left[\begin{array}{ll} \text{lex} & \text{[3]} \end{array}\right] \end{array}\right] \\
\text{target-name} & \text{[4]} \\
\text{src} & \left[\begin{array}{ll} \text{src} & \text{[5]} \end{array}\right]
\end{array}\right] \\
\text{process} & \left[\begin{array}{ll}
\text{lex} & \text{``report''} \\
\text{tense} & \text{past} \\
\text{type} & \text{verbal} \\
\text{object-clause} & \text{that}
\end{array}\right] \\
\text{partic} & \left[\begin{array}{ll}
\text{sayer} & \left[\begin{array}{ll} \text{cat} & \text{basic-proper} \\ \text{head} & \left[\begin{array}{ll} \text{lex} & \text{[5]} \end{array}\right] \end{array}\right] \\
\text{verbalization} & \left[\begin{array}{ll}
\text{process} & \left[\begin{array}{ll} \text{tense} & \text{past} \\ \text{voice} & \text{active} \\ \text{type} & \text{material} \\ \text{lex} & \text{[9]} \end{array}\right] \\
\text{partic} & \left[\begin{array}{ll} \text{affected} & \text{[7]} \\ \text{agent} & \text{[10]} \end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{circum} & \left[\begin{array}{ll}
\text{time} & \left[\begin{array}{ll} \text{prep} & \left[\begin{array}{ll} \text{lex} & \text{[6]} \end{array}\right] \\ \text{position} & \text{front} \\ \text{cat} & \text{pp} \\ \text{np} & \text{[13]} \end{array}\right] \\
\text{in-loc} & \left[\begin{array}{ll} \text{cat} & \text{common} \\ \text{head} & \left[\begin{array}{ll} \text{lex} & \text{[1]} \end{array}\right] \end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 6.4: Output of the application of the SUMMONS grammar on the template (KB format).

What gets added by the SUMMONS grammar is shown in Figure 6.4, more specifically in the *cat, obl, process, partic, and circum* sub-FDs (the first sub-FD, *sem* contains the input to SUMMONS). The numbers in the square brackets correspond to common paths in the FD. As an example, the number 3 corresponds to perp-individual_id in the semantic sub-FD, but it also gets mapped to the head of the perpetrator NP (number 10) which is also mapped to the agent of the verbalization (relative clause).

Similarly, the incident-location value ("Guatemala City"), indicated by the number 1, is mapped to the head of the circumstantial in-loc (on the last line of the FD).

6.3 Lexical and syntactic choice

The grammar used to generate sentences is in charge with selecting the proper phrasing and syntactic form. An excerpt of the sentence-level grammar is shown in Figure 6.5. Examples of linguistic processes that take place at this stage include the mapping between template types and sentence types, the choice of verbs for different report types (*report*, *deny*, *confirm*), the connection between the date of the event, the date of the report and the current date, the choice of relative clauses or passive voice.

The verb used to express the notion of a kidnapping ("kidnap", number 9 in the FD) is selected by SUMMONS's lexicon in relation with the incident type ("kidnapping"). Similarly, other incident types are expressed using appropriate lexical constructs.

The final stage of single-sentence generation involves sending the output of

```
(alt
 ((({sem sec_src source} given)
   ({real msg-type} none)
   (process ((type verbal)
             (object-clause that)))
   (partic ((sayer ((cat basic-proper)
                    (head ((lex {obl src src})))))
            (verbalization ((cat clause)
                            (process ((tense past)
                                      (alt
                                       ((({sem perp individual_id} given)
                                         (voice active))
                                        (({sem perp individual_id} none)
                                         (voice passive))))))
                            (alt
                             ((({obl stage} "accomplished")
                               (partic ((affected {obl target})
                                        (agent {obl perpetrator})))
                               (process ((type material)
                                         (lex {obl incident vlex}))))
                              ((partic ((affected ((cat clause)
                                                   (mood infinitive)
                                                   (proc ((type material)
                                                          (lex {obl incident vlex})))
                                                   (partic ((affected {obl target})))))))
                               (partic ((agent {obl perpetrator})))
                               (process ((type material)
                                         (alt
                                          ((({obl stage} "attempted")
                                            (lex "attempt"))
                                           (({obl stage} "threatened")
                                            (lex "threaten"))))))))))))))
  (({sem sec_src source} none)
   ({real msg-type} none)
   (process ((tense past)
             (alt
              ((({sem perp individual_id} given)
                (voice active))
               (({sem perp individual_id} none)
                (voice passive))))))))
```

Figure 6.5: Excerpts of the SUMMONS sentence-level grammar.

```
On January 12th 1990, ACAN-EFE reported that terrorists kidnapped
Hector Oqueli Colindras in Guatemala City.
```

Figure 6.6: Output of SUMMONS.

the previous stage to SURGE for surface generation. At this stage, all issues of syntactic and lexical choices have been resolved and SURGE can produce the actual sentence in English (Figure 6.6).

Part II

Language Reuse and Regeneration

This part of the dissertation includes three chapters related to our notion of language reuse and regeneration (LRR).

Chapter 7 introduces these notions and presents the motivation behind them as well as some examples (both from previous work and from the current thesis) to support them. The three stages of language reuse and regeneration (extraction, learning of constraints, and generation) are also introduced. Later, the chapter discusses how descriptions of entities are automatically extracted from different sources.

Chapter 8 describes the method used by SUMMONS to learn contextual constraints on the use of these descriptions which are then used by SUMMONS in the generation of summaries.

The last section of Chapter 8 describes how the information extracted in Chapters 7 and 8 is reused in the generation of summaries.

Chapter 7

Language reuse and regeneration

7.1 Motivation

Summaries generated by SUMMONS include references to a multitude of named entities (people, places, or organizations). These are sometimes quite familiar to the user (e.g., *Bill Clinton* or *Moscow*), however in many instances the generated summary must also include a detailed description to put the entities in context (e.g., *Guantanamo, the U.S. base in Cuba*). This chapter and the following two describe how such descriptions are automatically extracted by SUMMONS and how one out of many alternative descriptions is chosen as the actual description in the generated text.

The amount of text available through the World-Wide Web and in other on-line text corpora is enormous. If it is possible to create methods to reuse some of this text, instead of having to regenerate it from an internal representation (which, of course, would require analysis as well), the savings might be enormous. Two core problems present themselves:

1. how to identify which segments to reuse, and

2. how to ensure that the segments flow together naturally

The first problem requires some analysis of purpose. Since most text is written by humans for a certain purpose, we decided to figure out ways in which a natural language system can automatically decide to what purpose a certain text or text fragment was written and represent it in a way so that generation systems can use it. If the system has a similar purpose, it can simply extract the appropriate piece of human text and add it to the text that it generates.

We call the whole process **language reuse and regeneration**.

Definition 3 *:* **Language reuse** *refers to the process in which a lexical unit, a phrase, a clause, or an entire sentence is automatically extracted from a corpus, annotated with constraints on its use, and reused* **literally** *in automatically generated text.*

Definition 4 *:* **Language regeneration** *refers to the process of altering reused portions of already assembled text to insure that they read smoothly, fluently, and grammatically in context.*

The general idea of language reuse is related to that of "phrasal lexicons" [Kukich, 1983a, Jacobs, 1985]. Figure 7.1 shows one of the most basic forms of reusable text — **factual sentences**. Later in this chapter, we will present a full taxonomy of reusable text.

What makes our approach novel is the concept of dynamically extended phrasal lexicons. Whereas many of the sentences shown in Figure 7.1 can be as-

```
Water consists of reduced oxygen and oxidized hydrogen.

Deng Xiaoping was born on August 22, 1904 in Paifang Village in
Xiexing township, Guang'an County, in the province of Sichuan.

Benzene causes cancer in laboratory rats.
```

Figure 7.1: Sample reusable factual sentences.

sumed (under certain conditions) to be valid over extended periods of time, many cannot.

We will show how language reuse can be used to facilitate dynamic text generation (and summarization) in the cases in which the on-line sources of information are dynamically updated.

Traditionally, generation of dynamically changing information is handled using a combination of information extraction and text generation. First, relevant pieces of information are extracted from the source text (e.g., victim, date, and type of incident). Then, new text is produced by composing the extracted information using a generation grammar.

Our LRR-based approach improves on the IE+NLG approach in several ways:

- **Timeliness** – since the time of the report is known, the system can find the most recent wording, automatically obviating the need for a knowledge engineer to update the lexicon.

- **Expressibility** – since the text is written by a human, there are no technical restrictions in expressibility caused by a fixed grammar.

- **Appropriateness** – it is assumed that the human writer has judged the most appropriate facts to convey as well as the most appropriate form in which to convey them. Thus, LRR constructs capture and convey deliberate pragmatic decisions made by human writers.

- **Speed** – text generation being slower, the ability to reuse an existing piece of text speeds up the interaction with the user by bypassing both parsing and generation.

An important factor to consider is that not all factual sentences are can be reused in machine-generated text. We define the concept of **reusability** which can be used to decide whether to reuse the construct or not.

Definition 5 : **Reusability** *is a property of a language construct (sentence, phrase) which measures to what extent it can be reused in another context.*

We define two related concepts, **contextual attachment** and **lifetime**. The former describes how a construct is related to its context while the latter indicates the longevity of such a construct. Sentences that contain anaphora, for example, exhibit higher values of contextual attachment. On the other hand, noun phrase descriptions of entities exhibit high reusability and are therefore suitable for the kind of application that SUMMONS represents.

Definition 6 : **Contextual attachment** *is a measure of the extent to which a piece of text attached to its context.*

Value	Example
Indefinite	Water consists of reduced oxygen and oxidized hydrogen.
Long-term	Francfort is the capital of Kentucky.
Medium-term	Tony Blair is the British prime minister.
Short-term	A pair of major explosions near U.S. embassies rocked two African capitals early Friday.

Table 7.1: Examples of the four possible values for phrase lifetime.

Not all cases of high contextual attachment are difficult to handle, however. We devote the next two chapters to discuss a particularly auspicious case of contextual attachment, the fact that the choice of a description of an entity depends on the context in which it appears. We have developed an algorithm for SUMMONS to use this relationship and produce better-sounding summaries.

Definition 7 *: The* **lifetime of a textual construct** *is a measure that indicates for how long its factuality will hold.*

Table 7.1 shows possible values for the lifetime parameter of text.

It is difficult to analyze the lifetime of sentences automatically. For example, the sentences shown in Figure 7.1 have a very short lifetime — in some cases, only a few minutes. Descriptions of entities have, at least in principle, a lifetime that is long enough to make them reusable.

In some cases, the literal reuse of a phrase is not sufficient. In that case, we need to define a concept related to language reuse, namely **language regeneration**. In the cases when reuse is not sufficient, the system has to transform the source text to achieve its communicative goal. Some examples of transformations include **sentence simplification** and **nominalization** [Robin, 1994].

```
Ten people were killed overnight by suspected Islamic extremists in
the Ain-Defla region some 160 kilometers (100 miles) west of the
Algerian capital, security services announced Sunday.


Shortly before 11:30 a.m. the Dow Jones industrial average was up
85.47 points, or 1 percent, 7,935.31.


The stock of Microsoft (MSFT) rose 1 - 1/32 to 103.


Share prices on the London Stock Exchange were lower at midday
Wednesday.
```

Figure 7.2: Sample reusable factual sentences.

7.2 Discussion

In Figure 7.3 we compare a text generation system that uses information extraction (on the left-hand side) with a system that uses language reuse and regeneration instead (on the right-hand side).

The problem that we use for the comparison is illustrated when a system must answer a user's question such as "How did the Dow Jones index change today", when the system has access to a news story that already includes the sentence shown at the top of the figure. (Note that LR is not limited to whole sentences.)

With IE+NLG, the input text has to be parsed and converted to a template form. The question can be decomposed to a logical structure that is then converted to a query into the template shown on the left of the figure. Finally, a text generation grammar must be used to convert data from the template into natural

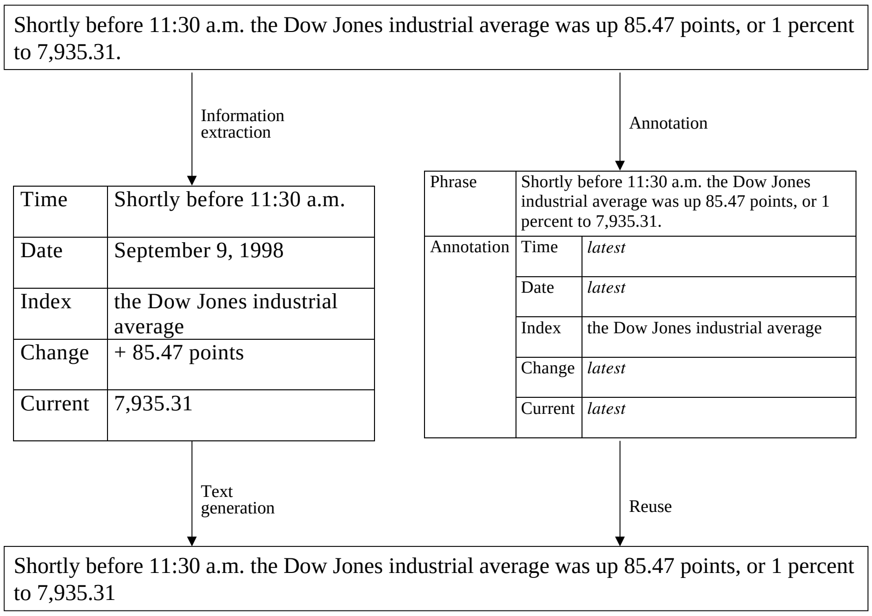

Figure 7.3: Comparison between IE+NLG and LRR.

language text.

In LRR, however, the annotation component needs only to find out that the sentence contains information about the change of the Dow Jones index and to use some external information (such as the date of the article) to determine that the information is the most current available. When the query comes to the system, the system has to realize that the sentence shown at the top is a potential answer to the user's query; it can then simply present the sentence to the user without involving a text generation component.

Neither IE+NLG alone, nor LRR alone is sufficient for building a robust system, hence the need for SUMMONS to incorporate both.

7.3 Extraction and reuse of descriptions

Some components of SUMMONS can be thought of as a testbed for ideas related to LRR. Several components of the system apply techniques described earlier in this chapter. We focus on the dynamically updated descriptions of named entities as a central example.

We choose descriptions of entities as the central application of LRR for several reasons. We deemed it to be a feasible task thanks to the simple structure of the information extracted and the ease in which it can be processed with a small-scale grammar.

The description component (PROFILE) described in Chapters 7 and 8) literally reuses descriptions of named entities. It uses a machine learning algorithm to associate the choice of one of many descriptions with the context in which it is used.

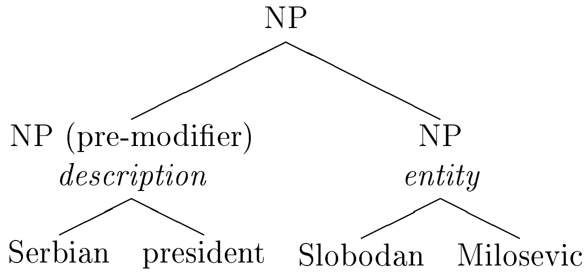

Figure 7.4: Pre-modifier relationship between a description and an entity.

We use the generation through LRR of noun phrase descriptions of named entities as a case study. This section describes how such descriptions are extracted automatically, while the following sections show how they are labeled according to the context in which they are used and then reused in text generation.

When a summary refers to a named entity (person, place, or organization), it can make use of descriptions extracted by the MUC systems. Problems arise when information needed for the summary is either missing from the input article(s) or not extracted by the information extraction system. In such cases, the information may be readily available in other current news stories, in past news, or in online databases. If the summarization system can find the needed information in other online sources, then it can produce an improved summary by merging information extracted from the input articles with information from the other sources [Radev and McKeown, 1997].

Both a description and an entity are noun phrases (see Figures 7.4 and 7.5).

Definition 8 *: The relation* $DescriptionOf(E)$ *relates a named entity* E *and a noun phrase,* D*, describing the named entity.*

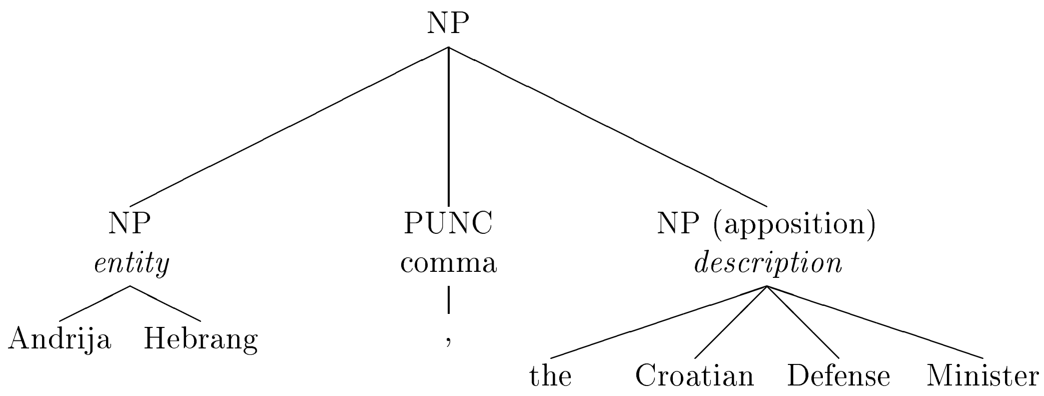

Figure 7.5: Apposition relationship between a description and an entity.

Table 7.7 shows several entity-description pairs (in comparison, Table 7.6 includes the entities only):

$DescriptionOf$ ("Tareq Aziz") = "Iraq's Deputy Prime Minister"

$DescriptionOf$ ("Richard Butler") = "Chief U.N. arms inspector"

In the news domain, a summary needs to refer to people, places, and organizations, and provide descriptions that clearly identify the entity for the reader. Such descriptions may not be present in the original text that is being summarized. For example, the American pilot Scott O'Grady, downed in Bosnia in June 1995, was unheard of by the American public prior to the incident. If a reader tuned into news on this event days later, descriptions from the initial articles may be more useful.

In this chapter, we describe an enhancement to the base summarization system, called the **profile manager**, which tracks prior references to a given entity by extracting descriptions for later use in summarization. The component includes the "Entity Extractor" and "Description Extractor" modules shown in Figure 2.4

Richard Butler met **Tareq Aziz** Monday after rejecting Iraqi attempts to set deadlines for finishing his work.

Yitzhak Mordechai will meet **Mahmoud Abbas** at 7 p.m. (1600 GMT) in Tel Aviv after a 16-month-long impasse in peacemaking.

Sinn Fein deferred a vote on Northern Ireland's peace deal Sunday.

Hundreds of troops patrolled **Dili** on Friday during the anniversary of Indonesia's 1976 annexation of the territory.

Figure 7.6: Sample sentences containing entities, but no descriptions.

Chief U.N. arms inspector **Richard Butler** met *Iraq's Deputy Prime Minister* **Tareq Aziz** Monday after rejecting Iraqi attempts to set deadlines for finishing his work.

Israel's Defense Minister **Yitzhak Mordechai** will meet senior Palestinian negotiator **Mahmoud Abbas** at 7 p.m. (1600 GMT) in Tel Aviv after a 16-month-long impasse in peacemaking.

Sinn Fein, the political wing of the Irish Republican Army, deferred a vote on Northern Ireland's peace deal Sunday.

Hundreds of troops patrolled **Dili**, the Timorese capital, on Friday during the anniversary of Indonesia's 1976 annexation of the territory.

Figure 7.7: Sample sentences containing both entities and descriptions.

and has the following features:

- It builds a database of profiles for entities by storing descriptions from a collected corpus of past news.
- It operates in real time, allowing for connections with the latest breaking, online news to extract information about the most recently mentioned individuals and organizations.
- It collects and merges information from various sources, thereby building a more complete record and reuse of information.
- As it parses and identifies descriptions, it builds a lexicalized, syntactic representation of the description in a form suitable for input to the FUF/SURGE language generation system.

As a result, SUMMONS is able to combine descriptions from articles appearing only a few minutes before the ones being summarized with descriptions from past news in a permanent storage for future use.

Since the profile manager constructs a lexicalized, syntactic FD from the extracted description, the generator can reuse the description in new contexts, merging it with other descriptions, into a new grammatical sentence. This would not be possible if only canned strings were used, with no information about their internal structure. Thus, in addition to collecting a knowledge source which provides identifying features of individuals, the profile manager also provides a lexicon of domain appropriate phrases that can be integrated with individual words from a generator's lexicon to produce summary wording in a flexible fashion. How SUMMONS actually uses the descriptions in generation will be shown in Chapter 8.

Description
A Serb
A member of Bosnia's multi-ethnic collective presidency
A senior Karadzic ally
The Serb member of Bosnia's multi-ethnic presidency
Serb hardliner
Serb presidency member
A Karadzic ally
Karadzic top aide
A member of the Bosnia-Herzegovina presidency
Top Karadzic aide
A top aide
Bosnian Serb hard-line leader

Table 7.2: Descriptions used for Momcilo Krajisnik extracted by PROFILE.

The rest of this chapter discusses the stages involved in the collection and reuse of descriptions.

7.4 Creation of a database of profiles

We call the list of all descriptions of a certain entity a **profile** for that entity. We name the entire component that manages profiles of entities PROFILE . For example, Table 7.2 shows the profile associated with a particular entity, Momcilo Krajisnik.

In this section, we describe the description management module of SUMMONS shown in Figure 2.4. We explain how entity names and descriptions for them are extracted from old newswire and how these descriptions are converted to FDs for surface generation.

7.4.1 Extraction of entity names from old newswire

To seed the database with an initial set of descriptions, we used a 1.7 MB corpus containing Reuters newswire from February to June 1995. All entity descriptions contained in it are added to the database. In addition, the Web sites shown in Figure 7.11 are currently indexed by the system.

At this stage, search is limited to the database of retrieved descriptions only, thus reducing search time, as no connections will be made to external news sources at the time of the query. Only when a suitable stored description cannot be found will the system initiate analysis of additional text.

- **Extraction of candidates for proper nouns**. After tagging the corpus using the PARTS part-of-speech tagger [Church, 1988], we used a CREP [Duford, 1993] grammar, shown in Figure 7.8 to first extract all possible candidates for entities. These candidates consist of all sequences of words that were tagged as proper nouns (NP) by PARTS. Our manual analysis showed that out of a total of 2150 entities recovered in this way, 1139 (52.9%) are not names of entities. Among these are bigrams such as "Prime Minister" or "Egyptian President" which were tagged as NP by PARTS. Table 7.3 shows how many entities we retrieve at this stage, and of them, how many pass the semantic filtering test described in the following paragraph.

- **Weeding out false candidates**. Our system analyzed all candidates for entity names using WORDNET [Miller *et al.*, 1990] and removed from consideration those that contain words appearing in WORDNET's dictionary. This resulted in a list of 421 unique entity names that we used for the automatic

```
SPACE           [ ]+
DOT             .

THE             [tT]he@AT                             ;; The Church Tagger
A_AN            [Aa](n)*@AT                           ;; (PARTS) is used.

ARTICLE         ({THE}|{A_AN})

T_NOUN          @(NP|NN|NPNP|NNS)
T_ADJ           @JJ
T_POSSESSIVE    @$
T_COMMA         @,

OF              [Oo]f@IN

POSSESSIVE      's{T_POSSESSIVE}
COMMA           ,{T_COMMA}

PROPER          [A-Z][-a-z][-a-zA-Z]+@(NP|NN|NPNP|NNS)
NUMBER          [A-Z][a-z][a-zA-Z]+@(CD)
WORD            [a-zA-Z]'*[a-z]+(-[a-zA-Z][a-z]+)*
NOUN            {WORD}{T_NOUN}
ADJ             {WORD}{T_ADJ}

PROPER2         {PROPER}{SPACE}{PROPER}
PROPER3         {PROPER}{SPACE}{PROPER}{SPACE}{PROPER}
PROPER4         {PROPER}{SPACE}{PROPER}{SPACE}{PROPER}{SPACE}{PROPER}
PROPER5         {PROPER}{SPACE}{PROPER}{SPACE}{PROPER}
                  {SPACE}{PROPER}{SPACE}{PROPER}

DESC_WORD       ({ARTICLE}{SPACE})*({NOUN}|{ADJ}|{POSSESSIVE}|{NUMBER})

NOUN_PHRASE     {DESC_WORD}(({SPACE}{OF})*{SPACE}{DESC_WORD})*

SEARCH_STRING   (({NOUN_PHRASE}{SPACE})+{SEARCH_0})|({SEARCH_0}{SPACE}
                  {COMMA}{SPACE}{NOUN_PHRASE})
SEARCH_0        [Yy]asser{T_NOUN}{SPACE}[Aa]rafat{T_NOUN}
```

Figure 7.8: Excerpts from CREP grammar used in the extraction of descriptions.

Stage	Two-word descriptions		Three-word descriptions	
	Entities	**Unique Entities**	**Entities**	**Unique Entities**
POS tagging only	9079	1546	2617	604
After WORDNET checkup	1509	395	81	26

Table 7.3: Two-word and three-word sequences retrieved by the system.

description extraction stage. All 421 entity names retrieved by the system are indeed proper nouns. We estimate recall to be in the 50% range, however we haven't performed a formal recall analysis.

7.4.2 Extraction of descriptions

There are two occasions in which we extract descriptions using finite-state techniques. The first case is when the entity that we want to describe was already extracted automatically (see Section 7.4.1) and exists in the database of descriptions. The second case is when we want a description to be retrieved in real time based on a request from the generation component.

In the first stage of either case, the profile manager generates finite-state representations of the entities that need to be described. These full expressions are used as input to the description extraction module which uses them to find candidate sentences in the corpus for extracting descriptions. Since the need for a description may arise at a later time than when the entity was found and may require searching new text, the description finder must first locate these expressions in the text.

These representations are fed to CREP, which extracts noun phrases on either side of the entity (either pre-modifiers or appositions) from the news corpus. The

Example	Trigger term	Semantic Category
Islamic Resistance Movement Hamas	movement	organization
radical Muslim group Hamas	group	organization
Addis Ababa, *the Ethiopian capital*	capital	location
South Africa's main black opposition leader, Mangosuthu Buthelezi	leader	occupation
Boerge Ousland, *33*	33	age
maverick French ex-soccer boss Bernard Tapie	boss	occupation
Italy's former prime minister, Silvio Berlusconi	minister	occupation
Sinn Fein, *the political arm of the Irish Republican Army*	arm	organization

Table 7.4: Examples of descriptions retrieved by CREP.

finite-state grammar for noun phrases that we use represents a variety of different syntactic structures for both pre-modifiers and appositions. Thus, they may range from simple nouns (e.g., "*president* Bill Clinton") to much longer expressions (e.g., "Gilberto Rodriguez Orejuela, *the head of the Cali cocaine cartel*"). Other forms of descriptions, such as those appearing in relative clauses, are not implemented.

Table 7.4 shows some of the different descriptions retrieved by CREP. For example, when the profile manager has retrieved the description "the political arm of the Irish Republican Army" for Sinn Fein, it looks at the head noun in the description NP ("arm") which we manually added to the list of trigger words to be categorized as an organization (see next section). It is important to notice that even though WORDNET typically presents problems with disambiguation of words retrieved from arbitrary text, we don't have any trouble disambiguating "arm" in this case due to the constraints on the context in which it appears (as an apposition describing an entity).

7.4.3 Categorization of descriptions

We consider the semantics (based on WORDNET) of each word in the description separately. This way, WORDNET helps us group extracted descriptions into categories. For the head noun of the description NP, we try to find a WORDNET hypernym

Entity	Description
Chandrababu Naidu	the computer-enthusiast chief minister of the Indian state of Andhra Pradesh
Alan Levitt	director of the National Youth Anti-Drug Media Campaign
Dharmalingan Sidharthan	leader of the People's Liberation Organization of Tamil Eelam
Gholamhossein Karbaschi	a powerful supporter of moderate president Mohammad Khatami
Murugasu Sivaithamparam	leader of the moderate Tamil United Liberation Front
Tim Bajarin	president of the Creative Strategies industry research firm
Valentin Moiseyev	a deputy chief of the foreign ministry's first Asian department
Mahmuti Bardhyl	a Swiss-based spokesman of the popular movement of Kosovo
Mitchel McLaughlin	Chairman of the Irish Republican Army's Sinn Fein political wing
Reid Detchon	executive director of the Interactive Travel Services Association
Rodrigo Infante	general manager of the Association of Chilean salmon farmers
Stephen Brobeck	executive director of Consumer Federation of America
Wolfgang Lieb	Chairman of the conference of state education ministers
Willy Voet	the personal masseur of last year's runner-up Richard Virenque of France

Table 7.5: Long descriptions retrieved by CREP.

that categorizes the description according to the type of entity it describes (e.g., "profession", "nationality", and "organization". Each of these concepts is triggered by one or more words (which we call "trigger terms") in the description. Table 7.4 shows some examples of descriptions and the concepts under which they are classified based on the WORDNET hypernyms for some "trigger" words. For example, all of the following triggers in the list ("minister", "head", "administrator", and "commissioner") can be traced up to "leader" in the WORDNET hierarchy. We have currently a list of 75 such trigger words that we have compiled manually. If no trigger word is found (in less than 2% of the cases), no category is assigned to the description. Table 7.6 shows more examples of categorized descriptions.

7.4.4 Organization of descriptions in a database of profiles

For each retrieved entity we create a new profile in a database of profiles. We keep information about the surface string that is used to describe the entity in newswire (e.g., "Addis Ababa"), the source of the description and the date that the entry has been made in the database (e.g., "reuters95_06_25") or the URL from which it was

Description	Categories
His Excellency	address
chief executive	leadership
	business
Oracle's chairman	company
	leadership
Cambodian foreign minister	country
	political post
Italian virtuoso singer	country
	singing
	expert
Opera star	singing
	fame
Protestant leader	leadership
	religious affiliation
Late Chinese leader	country
	dead
	leadership
Netanyahu's media adviser	REL2
billionnaire	wealth

Table 7.6: Sample Descriptions.

```
KEY: john major
SOURCE: reuters95_03-06_.nws
DESCRIPTION: british prime minister
FREQUENCY: 75
DESCRIPTION: prime minister
FREQUENCY: 58
DESCRIPTION: a defiant british prime minister
FREQUENCY: 2
DESCRIPTION: his british counterpart
FREQUENCY: 1
```

Figure 7.9: Profile for John Major.

extracted. In addition to these pieces of meta-information, all retrieved descriptions and their frequencies are also stored.

Currently, our system doesn't have the capability of matching references to the same entity that use different wordings. As a result, we keep separate profiles for each of the following: "Robert Dole", "Dole", and "Bob Dole". We use each of these strings as the key in the database of descriptions. There exist techniques, however, which address this problem [Wacholder *et al.*, 1997].

Figure 7.9 shows the profile associated with the key "John Major". It can be seen that four different descriptions have been used in the parsed corpus to describe John Major. Two of the four are common and are used in SUMMONS, whereas the other two result from errors (such as incorrect part of speech assignment) made by PARTS and/or CREP.

Since the database of descriptions is stored in a relational DBMS, queries are performed using simple SQL statements. An example of an SQL command is shown in Figure 7.10.

The database of profiles is updated every time a query retrieves new descrip-

```
select
  description.description,entity.entity
from
  entity, description
where
  entity.entity_id = description.entity_id
order by
  entity.entity
```

Figure 7.10: SQL code for searching the description database.

```
YAHOO
http://www.yahoo.com/headlines

WASHINGTON POST-AP
http://www.washingtonpost.com/wp-srv/digest/digest.htm

USA TODAY
http://www.usatoday.com/news/digest

PRODIGY-AP
http://headlines.prodigy.com/APnews/src/nm20indx.htm
```

Figure 7.11: Newswire sites used to extract descriptions.

tions matching a certain key. The following table includes information about the seed URLs that were used to extract descriptions of entities:

```
Italy@NPNP 's@$ former@JJ prime@JJ
minister@NN Silvio@NPNP Berlusconi@NPNP
```

Figure 7.12: Retrieved sentence containing a description for Silvio Berlusconi.

$$\begin{bmatrix} \text{cat} & \text{np} \\ \text{complex} & \text{apposition} \\ \text{restrictive} & \text{no} \\ \text{distinct} & \begin{bmatrix} \text{car} & \begin{bmatrix} \text{cat} & \text{common} \\ \text{possessor} & \begin{bmatrix}\text{cat} & \text{common}\\ \text{lex} & \text{“Italy”}\end{bmatrix} \\ \text{classifier} & \begin{bmatrix} \text{cat} & \text{noun-compound} \\ \text{classifier} & \begin{bmatrix}\text{lex} & \text{“former”}\end{bmatrix} \\ \text{head} & \begin{bmatrix}\text{lex} & \text{“prime”}\end{bmatrix}\end{bmatrix} \\ \text{head} & \begin{bmatrix}\text{lex} & \text{“minister”}\end{bmatrix}\end{bmatrix} \\ \text{cdr} & \begin{bmatrix}\text{car} & \begin{bmatrix}\text{cat} & \text{person-name}\\ \text{first-name} & \begin{bmatrix}\text{lex} & \text{“Silvio”}\end{bmatrix} \\ \text{last-name} & \begin{bmatrix}\text{lex} & \text{“Berlusconi”}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Figure 7.13: Generated FD for Silvio Berlusconi (KB format).

7.5 Representing descriptions in a form suitable for text generation

7.5.1 Transformation of descriptions into Functional Descriptions

In order to reuse the extracted descriptions in the generation of summaries, we have developed a module that converts finite-state descriptions retrieved by the description extractor into functional descriptions that we can use directly in generation. A description retrieved by the system is shown in Figure 7.12. The corresponding FD is shown in Figure 7.13.

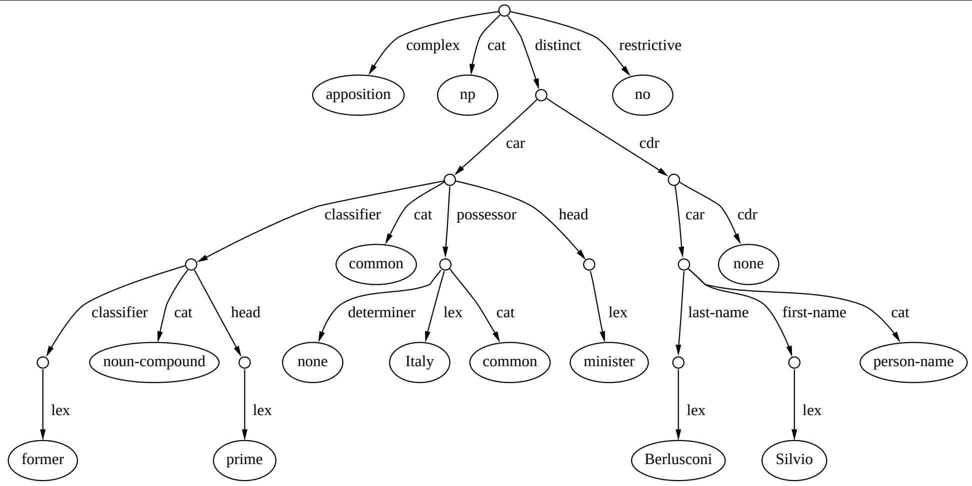

Figure 7.14: Generated FD for Silvio Berlusconi (Graph format).

7.5.2 Regenerating descriptions

Keeping the surface form of the descriptions is helpful for LRR-based inclusion in output (which is how SUMMONS actually generates the descriptions). However, we have also considered some potential uses of the parsed representations. While we do not currently perform them, by converting the descriptions into FDs, we facilitate future work on them. Since in the generation component 8 we generate no more than 1 description per entity, we avoid generating repetitive descriptions, e.g., "The Russian President, Boris Yeltsin, President of Russia".

- **Transformations**. The deeper representation allows for grammatical transformations, such as aggregation: e.g., "president Yeltsin" + "president Clinton" can be generated as "presidents Yeltsin and Clinton".

- **Unification with existing ontologies**. E.g., if an ontology contains information about the word "president" as being a realization of the concept "head of state", then under certain conditions, the description can be replaced by one referring to "head of state".

- **Generation of referring expressions**. In the previous example, if "president Bill Clinton" is used in a sentence, then "head of state" can be used as a referring expression in a subsequent sentence.

- **Modification/Update of descriptions**. If we have retrieved "prime minister" as a description for Silvio Berlusconi, and later we learn that someone else has become Italy's primer minister, then we can generate "former prime minister" using a transformation of the old FD.

- **Lexical choice**. When different descriptions are automatically marked for semantics, the profile manager can prefer to generate one over another based on semantic features. This is useful if a summary discusses events related to one description associated with the entity more than the others. For example, when an article concerns Bill Clinton on the campaign trail, then the description "democratic presidential candidate" is more appropriate. On the other hand, when an article concerns an international summit of world leaders, then the description "U.S. President" is more appropriate. An implementation of this idea is shown in the next chapter.

- **Merging lexicons**. The lexicon generated automatically by the system can be merged with a manually compiled domain lexicon.

7.6 Evaluation of performance

7.6.1 Recall

We should note that the relatively low recall of PROFILE in extracting all entity-description pairs is not really a problem : one can easily increase the total number N of good descriptions retrieved about a particular entity, by, for a constant value of R (recall), taking a sufficiently large training corpus (C is the size of the corpus):

$$R = \frac{N}{C}$$

For $R =$ const, we need to process $D = \frac{N}{R}$ descriptions to obtain the desired number of good descriptions (N).

If, on average, the rate of appearance of correct descriptions in a corpus is r descriptions per word in the corpus, then in order to retrieve N descriptions, we will have to process a corpus consisting of W words:

$$W = \frac{N}{R * r}$$

7.6.2 Conversion

The FD generation component produces syntactically correct functional descriptions that can be used to generate English-language descriptions using FUF and SURGE, and can also be used in a general-purpose summarization system in the domain of current news.

All components of the system assume no prior domain knowledge and are therefore portable to many domains, including sports, entertainment, and business.

Category	Type of problem	Examples
EXTR	extraction error	Kerry Packer
ATT	attachment error	the envy of media baron
REL1	relative description	his long-time rival
REL2	relative description within phrase	Netanyahu's adviser
		a senior Karadzic ally
REL	reverse relative description	Zhuo Lin, widow of paramount leader
PLUR	plural description	commanders
		opera stars
ADJ	adjective	a stern-faced
		the acquisitive

Table 7.7: Problematic categories.

7.6.3 Error analysis

We categorized some problematic descriptions into one or more "error" categories. Such problems can be classified into the categories shown in Table 7.7. The different "error" categories are shown below:

- EXTR - a non-description being extracted as a description.
- ATT - entity is described by NP other than the main NP in the extracted "description".
- REL1 - description is related to text outside the description + entity pair.
- REL2 - description is related to text inside the description + entity pair. Example: "Benjamin Netanyahu's media adviser" — "media adviser" should not be extracted as a description for "Benjamin Netanyahu".
- REL - reverse relation (another entity is needed to complete the description).
- PLUR - scoping error (e.g., plural).
- ADJ - adjectival description.

Our grammar did a particularly poor job on articles from the entertainment and sports categories, which created a large number of attachment problems. Texts of these categories often contain enumerations of people participating in a movie or in a sports game. Examples: *The 25-year-old Testud, whose victims this season also include Monica Seles, Arantxa Sanchez Vicario and Lindsay Davenport ...* or *The movie "Sphere", based on Michael Crichton's novel, stars Dustin Hoffman, Sharon Stone and Samuel L. Jackson.* In both cases some of the names are improperly extracted as descriptions.

Clearly, a better extraction grammar would correctly fail to extract names of people as descriptions.

7.6.4 Web interface

Figures 7.15 and 7.16 show the Web interface to PROFILE. Users can select an entity (such as "John Major"), specify what semantic classes of descriptions they want to retrieve (e.g., age, position, nationality) as well as the maximal number of queries that they want. They can also specify which sources of news should be searched. Currently, the system has an interface to Reuters News [Reuters, 1998], The CNN Web site [CNN, 1998] and to all Usenet news delivered via NNTP to our local news domain.

The Web-based interface is accessible publicly at `http://www.cs.columbia.edu/~radev/profile`. All answers to queries are cached for a specific amount of time (currently, one hour) and are returned immediately, without access to PROFILE if needed by a subsequent query.

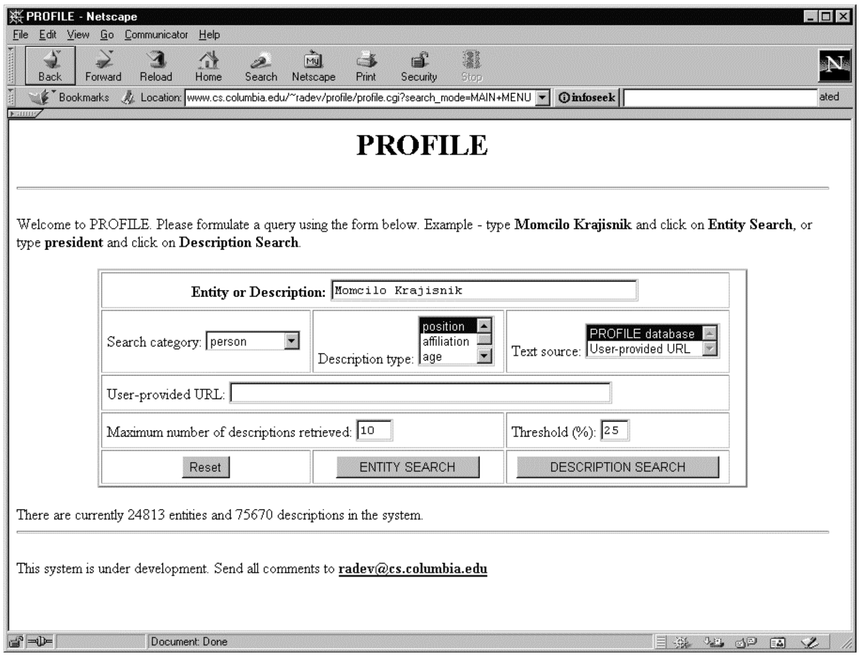

Figure 7.15: Web-based interface to PROFILE (input parameters).

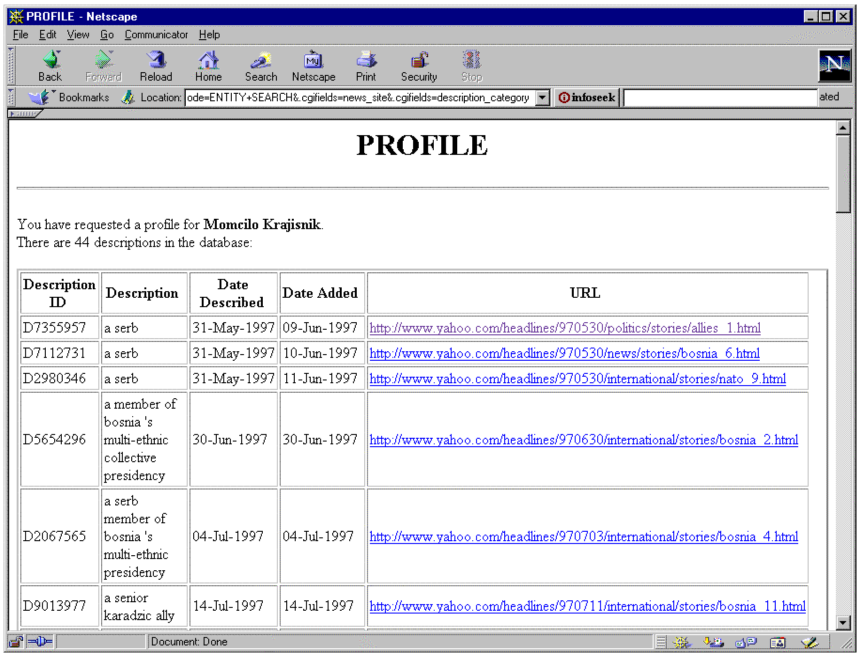

Figure 7.16: Web-based interface to PROFILE (output).

Chapter 8

Learning semantic and pragmatic constraints on descriptions

8.1 Introduction

Human writers typically make deliberate decisions about picking a particular way of expressing a certain concept. These decisions are made based on the topic of the text and the effect that the writer wants to achieve. Such contextual and pragmatic constraints are obvious to experienced writers who produce context-specific text without much effort and have been the focus of some research in natural language generation [Hovy, 1987]. However, in order for a computer to produce text in a similar way, these constraints must be known. Either they have to be added manually by an expert or a system must be able to acquire them in an automatic way.

An example related to the lexical choice of an appropriate nominal description of a person should make the above clear. Even though it seems intuitive that

Bill Clinton should always be described with the NP *U.S. president* or a variation thereof, it turns out that many other descriptions appear in on-line news stories that characterize him in light of the topic of the article. For example, an article from 1996 on elections uses *Bill Clinton, the democratic presidential candidate*, while a 1997 article on a false bomb alert in Little Rock, Ark. uses *Bill Clinton, an Arkansas native.*

This chapter presents the results of a study of the correlation between named entities (people, places, or organizations) and noun phrases used to describe them in a corpus.

Intuitively, the use of a description is based on a deliberate decision on the part of the author of a piece of text. A writer is likely to select a description that puts the entity in the context of the rest of the article.

It is known that the distribution of words in a document is related to its topic [Salton and McGill, 1983]. We have developed related techniques for approximating pragmatic constraints using words that appear in the immediate context of the entity.

We will show that context influences the choice of a description, as do several other linguistic indicators. Although each of the indicators by itself does not provide enough empirical data to distinguish among all descriptions that are related to an entity, a carefully selected combination of such indicators provides enough information in to pick an appropriate description with more than 80% accuracy.

Section 8.2 describes how we can automatically obtain enough constraints on the usage of descriptions. In Section 8.3, we show how such constructions are related to language reuse. In Section 8.4 we describe our experimental setup and the algorithms that we have designed. Section 8.5 includes a description of our results while Section 8.6 describes how the descriptions are used in producing enhanced summaries. In Section 8.7 we discuss some possible extensions to our study and we provide some thoughts about possible uses of our framework.

8.2 Problem description

Each entity appearing in a text can have multiple descriptions (we have identified up to several dozen for a single referent) associated with it. It turns out that there is a large variety in the size of the profile (number of distinct descriptions) for different entities. Table 8.1 shows a subset of the profile for Ung Huot, the former foreign minister of Cambodia, who was elected prime minister at some point of time during the run of our experiment. A few sample semantic features of the descriptions in Table 8.1 are shown as separate columns. We are not actually determining the semantic categories shown as column headers in Table 8.1. We are instead using the *synset offsets* of the primary senses of all words to represent the descriptions in a *lexico-semantic matrix* (Table 8.2). The *synset offset* of a word is a unique number which is used to represent the word and all of its synonyms (hence, *syn*-set) in WORDNET. The synset offset of a word and the synset offset of the parent node of the word are trivial to extract from WORDNET (given our assumption of only considering the first sense of each word).

Consider the profile related to Bill Clinton, shown in Table 8.3. If a user is interested in all possible descriptions of Bill Clinton, then just showing him the list of descriptions will suffice. However, the summary generator (or any text generation system) needs to be able to pick a description that is most appropriate for the text

Description	**Semantic categories** addressing	country	male	new	political post	seniority
A senior member						X
Cambodia's		X				
Cambodian foreign minister		X			X	
Co-premier					X	
First prime minister					X	
Foreign minister					X	
His Excellency	X					
Mr.			X			
New co-premier				X	X	
New first prime minister				X	X	
Newly-appointed first prime minister				X	X	
Premier					X	
Prime minister					X	

Table 8.1: Profile of Ung Huot.

	Word synsets			**Parent synsets**	
Description	07147929 premier	07009772 Kampuchean	...	07412658 minister	07087841 associate
A senior member			...		X
Cambodia's		X	...		
Cambodian foreign minister		X	...	X	
Co-premier	X		...	X	
First prime minister	X		...	X	
Foreign minister			...	X	
His Excellency			...		
Mr.			...		
New co-premier	X		...	X	
New first prime minister	X		...	X	
Newly-appointed first prime minister	X		...	X	
Premier	X		...	X	
Prime minister	X		...	X	

Table 8.2: Lexico-semantic matrix associated with the profile of Ung Huot.

Description
U.S. President
President
An Arkansas native
Democratic presidential candidate

Table 8.3: Descriptions used for Bill Clinton.

Description	Topic of the article
U.S. President	foreign relations
President	national affairs
An Arkansas native	false bomb alert in AR
Democratic presidential candidate	elections

Table 8.4: Descriptions used for Bill Clinton along with context.

being generated.

After analysis of the use of descriptions in news stories, we concluded that there is a correlation between the purpose of the article and the choice of the description. Table 8.4 shows some examples of such correlations.

Our hypothesis is that it should be possible to predict the use of a particular description (or, at least, to define some semantic constraints on its choice) given the context in which it appears. Some semantic categories used are shown in Figure 8.1. We are essentially trying to approximate the choice of the actual description with the semantic categories of the words that comprise it. We also approximate the topic of the article (or text to be generated) using words appearing near the entity that needs to be described.

We have processed 178 MB[1] of newswire and analyzed the use of descriptions related to 11,504 entities. Even though PROFILE extracts other entities in addition

[1] The corpus contains 19,473 news stories that cover the period October 1, 1997 – January 9, 1998 that were available through PROFILE.

place of birth
former occupation
current occupation
political position
non-political job
nationality
political affiliation
age
ethnicity
gender
religious affiliation

Figure 8.1: Semantic categories used for description categorization.

to people (e.g., places and organizations), we restricted this analysis to names of people only. We did so because only a very small number of the descriptions were associated with places and organizations. We believe, however, that our findings relate to the other types of entities as well.

We have investigated 35,206 tuples, consisting of an entity, a description, an article ID, and the position (sentence number) in the article in which the entity-description pair occurs. Since there are 11,504 distinct entities, we had on average 3.06 distinct descriptions per entity ($DDPE$). Table 8.5 shows the distribution of $DDPE$ values across the corpus. Notice that a large number of entities (9,053 out of the 11,504) have a single description. These are not as interesting for our analysis as the remaining 2,451 entities that have $DDPE$ values between 2 and 24.

8.3 Language reuse in text generation

Descriptions of entities are a particular instance of a surface structure that can be reused relatively easily. *Syntactic* constraints related to the use of descriptions are modest, since descriptions are always noun phrases that appear as either pre-

DDPE	count	DDPE	count	DDPE	count
1	9,053	8	27	15	4
2	1,481	9	26	16	2
3	472	10	12	17	2
4	182	11	10	18	1
5	112	12	8	19	1
6	74	13	2	24	1
7	31	14	3		

Table 8.5: Number of distinct descriptions per entity ($DDPE$).

modifiers or appositions[2], they are quite flexibly usable in any generated text in which an entity can be modified with an appropriate description. We will show in the rest of the paper how the requisite *semantic* (i.e., *"what is the meaning of the description to pick?"*) and *pragmatic* constraints (i.e., *"what purpose does using the description achieve?"*) can be extracted automatically.

Given a profile like the one shown in Table 8.1, and an appropriate set of semantic constraints (columns 2–7 of the table), the generation component needs to perform a profile lookup and select a row (description) that satisfies most or all semantic constraints. For example, if the semantic constraints specify that the description has to include the country and the political position of Ung Huot, the most appropriate description is *Cambodian foreign minister*.

We have identified two categories of descriptions: relational and non-relational. This section describes why it is important to make this distinction and how it influences the process of description selection in generation.

Definition 9 *: A* **relational description** *contains an anaphor to a referent outside the entity + description pair. Example: "Zhuo Lin, widow of paramount leader Deng Xiaoping".*

[2]We haven't included relative clauses in our study.

DIRECTOR:	{07063762} director, manager, managing director
	⇒ {07063507} administrator, decision maker
	⇒ {07311393} head, chief, top dog
	⇒ {06950891} leader
	⇒ {00004123} person, individual, someone, somebody, mortal, human, soul
	⇒ {00002086} life form, organism, being, living thing
	⇒ {00001740} entity, something

Figure 8.2: Hypernym chain of "director" in WORDNET , showing synset offsets.

A relational description cannot be reused without the proper referent. We have thus concentrated our efforts to non-relational descriptions. A more detailed taxonomy of descriptions was presented in the previous chapter.

Definition 10 *: A* **non-relational description** *does not contain an anaphor to a referent outside the entity + description pair. Example: "Deng Xiaoping, the paramount Chinese leader".*

8.4 Experimental setup

In our experiments, we have used two widely available tools, WORDNET and RIPPER (see Appendix C).

We use chains of hypernyms when we need to approximate the usage of a particular word in a description using its ancestor and sibling nodes in WORDNET. Particularly useful for our application are the synset offsets of the words in a description. Figure 8.2 shows that the synset offset for the concept *"administrator, decision maker"* is *"{07063507}"*, while its hypernym, *"head, chief, top dog"* has a synset offset of *"{07311393}"*. These numbers are used as classification features.

RIPPER [Cohen, 1995, Cohen, 1996] is an algorithm that learns rules from example tuples in a relation. Attributes in the tuples can be integers (e.g., length of an article, in words), sets (e.g., semantic features), or bags (e.g., words that appear in a sentence or document). We use RIPPER to learn rules that correlate context and other linguistic indicators with the semantics of the description being extracted and subsequently reused.

Some adjustments needed to be made in the learning process. RIPPER is designed to learn rules that classify data into atomic classes (e.g., *"good"*, *"average"*, and *"bad"*). We had to modify its algorithm in order to classify data into *sets of atoms*. For example, a rule can have the form *"if CONDITION then [{07063762} {02864326} {00017954}]"*[3]. This rule states that if a certain "CONDITION" (which is a function of the indicators related to the description) is met, then the description is likely to contain words that are semantically related to the three listed WORDNET nodes.

The stages of our experiments are described in detail in the remainder of this section.

8.4.1 Semantic tagging of descriptions

The PROFILE component of SUMMONS processes WWW-accessible newswire on a round-the-clock basis and extracts entities (people, places, and organizations) along with related descriptions. The extraction grammar, developed in CREP, covers a variety of pre-modifier and appositional noun phrases.

For each word w_i in a description, we use a version of WORDNET to extract the synset offset of the immediate parent of w_i.

[3]These offsets correspond to the WORDNET nodes *"manager"*, *"internet"*, and *"group"*

8.4.2 Finding linguistic cues

Initially, we were interested in discovering rules manually and then validating them using the learning algorithm. However, the task proved (nearly) impossible considering the sheer size of the corpus. One rule that we hypothesized and wanted to verify empirically at this stage was *parallelism*. This linguistically-motivated rule states that in a sentence with a parallel structure (for instance, the sentence fragment *"... Alija Izetbegovic, a Muslim, Kresimir Zubak, a Croat, and Momcilo Krajisnik, a Serb..."*) all entities involved have similar descriptions. However, rules stated at such a detailed syntactic level take too long to process on a 180 MB corpus and, further, no more than a handful of such rules can be discovered manually. As a result, we made a decision to extract all indicators automatically.

8.4.3 Extracting linguistic cues automatically

The list of indicators that we use in our system are the following:

- **Context:** (using a window of size 4, excluding the actual description used, but not the entity itself) - e.g., *"['clinton' 'clinton' 'counsel' 'counsel' 'decision' 'decision' 'gore' 'gore' 'ind' 'ind' 'index' 'news' 'november' 'wednesday']"* is a bag of words found near a description of Bill Clinton in the training corpus.

- **Length of the article:** - an integer.

- **Name of the entity:** - e.g., "Bill Clinton".

- **Profile:** The entire profile related to a person (all descriptions of that person that are found in the training corpus).

Number	Context	Entity	Description	Length	Profile	Parent	Classes
1	Election, promised, said, carry, party ...	Kim Dae-Jung	Veteran opposition leader	949	Candidate, chief, policy, maker, Korean ...	person, leader, Asian, important person ...	{07136302} {07486519} {07311393} {06950891} {07486079}
2	Introduced, responsible, running, should, bringing ...	Kim Dae-Jung	South Korea's opposition candidate	629	Candidate, chief, policy, maker, Korean ...	person, leader, Asian, important person ...	{07136302} {07486519} {07311393} {06950891} {07486079}
3	Attend,, during, party, time, traditionally ...	Kim Dae-Jung	A front-runner	535	Candidate, chief, policy, maker, Korean ...	person, leader, Asian, important person ...	{07136302} {07486519} {07311393} {06950891} {07486079}
4	Discuss, making, party, statement, said ...	Kim Dae-Jung	A front-runner	1114	Candidate, chief, policy, maker, Korean ...	person, leader, Asian, important person ...	{07136302} {07486519} {07311393} {06950891} {07486079}
5	New, party, politics, in, it ...	Kim Dae-Jung	South Korea's president-elect	449	Candidate, chief, policy, maker, Korean ...	person, leader, Asian, important person ...	{07136302} {07486519} {07311393} {06950891} {07486079}

Table 8.6: Sample tuples from training corpus.

- **Synset Offsets:** - the WORDNET node numbers of all words (and their parents) that appear in the profile associated with the entity that we want to describe.

Each tuple is represented as a feature vector:

$$(\text{Context}, \text{Entity}, \text{Description}, \text{Length}, \text{Profile}, \text{Parent}) \Rightarrow \text{Class} \qquad (8.1)$$

A small sample of the training corpus represented as training tuples for RIPPER is shown in Table 8.6. The table, consisting of 377,604 separate[4] training and testing tuples was created automatically from the corpus.

8.4.4 Applying a machine learning method

To learn rules, we ran RIPPER on 90% (10,353) of the entities in the entire corpus. We kept the remaining 10% (or 1,151 entities) for evaluation. In one of the

[4]test tuples were not seen during training.

Class	Rule
07136302	IF PROFILE ˜ P07136302 LENGTH ≤ 603 LENGTH ≥ 361 .
07136302	IF PROFILE ˜ P07136302 CONTEXT ˜ presidential LENGTH ≤ 412 .
07136302	IF PROFILE ˜ P07136302 CONTEXT ˜ nominee CONTEXT during .
07136302	IF PROFILE ˜ P07136302 CONTEXT ˜ case .
07136302	IF PROFILE ˜ P07136302 LENGTH ≤ 603 LENGTH ≥ 390 LENGTH ≤ 412 .
07136302	IF PROFILE ˜ P07136302 CONTEXT ˜ nominee CONTEXT

Table 8.7: Sample rules discovered by the system.

experiments, RIPPER extracted 4,085 rules from a training corpus with 100,000 tuples.

Sample rules discovered by the system are shown in Table 8.7. We should note that, as typically observed with decision tree based algorithms, most of the rules are not intuitively understandable by humans. The first rule, for example, indicates that if the synset offset *"{07136302}"* appears in the context and the length of the document is between 361 and 603 words, then regardless of the other synset offsets, a word that belongs to synset offset *"{07136302}"* should be used in the description. In a similar way, the other rules in the table indicate in which other cases the same offset should be picked.

8.5 Results and evaluation

We have performed a standard evaluation of the precision and recall that our system achieves in selecting a description. Table 8.9 shows our results under two sets of parameters.

Precision and recall are based on how well the system predicts a set of semantic constraints. Precision (or P) is defined to be the number of matches divided by the number of elements in the predicted set. Recall (or R) is the number of matches divided by the number of elements in the correct set. We should note

Model	System	Precision	Recall
[B] [D]	[A] [B] [C]	33.3 %	50.0 %
[A] [B] [C]	[A] [B] [D]	66.7 %	66.7 %

Table 8.8: Evaluation example.

that this measure is perhaps too tough on the system[5]. If, for example, the system predicts *[A] [B] [C]*, but the set of constraints on the actual description is *[B] [D]*, we would compute that $P = 33.3\%$ and $R = 50.0\%$. Table 8.9 reports the average values of P and R for all training examples[6]. The results[7] are shown in Table 8.8.

As an example, let's consider two rules. The first one assigns a tuple to class A if X is true. The other one assigns a tuple to class B if the Y is true. In our case, this can be interpreted as follows: if there is evidence of A, label the tuple with X; if there is evidence of B, label the tuple with Y. If there is evidence of both A and B then the two rules are not considered to be contradictory. Instead, we assume that the tuple should be labeled with both X and Y. This way, we can make use of rules that indicate that if an entity appears in a context in which two rules match. Then both rules will contribute semantically to the choice of the description for the entity in that particular context. In other words, we are learning a set of semantic constraints, which is not in W (the number of synsets in WordNet), but rather in 2^W (the set of all subsets of W).

[5] We find it reasonable, though, as we are not measuring the success of the system at picking a particular word, but rather a word of a given synset, thus allowing for synonyms to be scored as system successes.

[6] We run ripper in a so-called "noise-free mode", which causes the condition parts of the rules it discovers to be mutually exclusive and therefore, the values of P and R on the training data are both 100%.

[7] This method of evaluation was suggested by Vasileios Hatzivassiloglou.

	Word nodes only		**Word and parent nodes**	
Training set size	**Precision**	**Recall**	**Precision**	**Recall**
500	64.29%	2.86%	78.57%	2.86%
1,000	71.43%	2.86%	85.71%	2.86%
2,000	42.86%	40.71%	67.86%	62.14%
5,000	59.33%	48.40%	64.67%	53.73%
10,000	69.72%	45.04%	74.44%	59.32%
15,000	76.24%	44.02%	73.39%	53.17%
20,000	76.25%	49.91%	79.08%	58.70%
25,000	83.37%	52.26%	82.39%	57.49%
30,000	80.14%	50.55%	82.77%	57.66%
50,000	83.13%	58.53%	88.87%	63.39%
100,000	85.42%	62.81%	89.70%	64.64%
150,000	87.07%	63.17%		
200,000	85.73%	62.86%		
250,000	87.15%	63.85%		

Table 8.9: Values for precision and recall using word nodes only (left) and both word and parent nodes (right). "Training set size" refers to the number of training tuples.

$$X \Rightarrow [A], \tag{8.2}$$

$$Y \Rightarrow [B], \tag{8.3}$$

$$X \wedge Y \Rightarrow [A][B]. \tag{8.4}$$

Selecting appropriate descriptions using our algorithm is feasible even though the values of precision and recall obtained may seem only moderately high. The reason is that the problem that we are trying to solve is underspecified. That is, in the same context, more than one description can be potentially used. Mutually interchangeable descriptions include synonyms and near synonyms (*"leader"* vs. *"chief*) or pairs of descriptions of different generality (*"U.S. president"* vs. *"president"*).

This type of evaluation requires the availability of human judges, and therefore we opted for the automated evaluation instead.

There are two parts to the evaluation: how well does the system perform in selecting semantic features (WORDNET nodes) and how well does it work in constraining the choice of a description. To select a description, our system does a lookup in the profile for a possible description that satisfies most semantic constraints (e.g., we select a row in Table 8.1 based on constraints on the columns).

Our system depends crucially on the multiple components that we use. For example, the shallow CREP grammar that is used in extracting entities and descriptions often fails to extract good descriptions, mostly due to incorrect PP attachment. We have also had problems from the part-of-speech tagger and, as a result, we occasionally incorrectly extract word sequences that do not represent descriptions.

8.6 Generation of descriptions

Since our priority in building the LRR component was to allow SUMMONS to produce enhanced summaries, we implemented the choice of descriptions as part of SUMMONS.

One of the user-interface options (see Chapter 5) allows for the choice between base and enhanced summaries. If "add-descriptions" is selected, SUMMONS applies planning operators to find the most appropriate description in the given context. One of these operators, shown in Figure 8.3, checks if the "add-descriptions" option is set and then passes the entire list of templates and the entity that needs to be described to a function get-profile which implements the algorithm described earlier in this chapter. The return value of the function is the description that SUMMONS includes in generating the summary.

```
(def-operator
  add-description

  "add victim description"

  ((condition
    (eq
   {meta use-description}
    "true"}))

   (action
    ("left"
     {shadow hum_tgt-description}
     (get-profile
      {shadow hum_tgt-name}
      lot)
     ))

   (type
    "minimal")
   ))
```

Figure 8.3: Description-inserting operator.

Algorithm 2 Generating description to include in the summary.

generate summary using the planning operators
repeat
  extract contextual information about next entity in summary
  if first mention of entity in text and add-descriptions is set **then**
    pick an appropriate description from PROFILE and include it in the summary.
  end if
until no more entities need to be described
output summary

Figure 2 describes the skeleton of the algorithm used to incorporate descriptions of entities in the summaries generated by SUMMONS.

8.7 Applications and future work

We use PROFILE to improve lexical choice in the summary generation component [Radev and McKeown, 1998]. There are two particularly appealing cases : (1) when the extraction component has failed to extract a description, and (2) when the user model (user's interests, knowledge of the entity and personal preferences for sources of information and for either conciseness or verbosity) dictates that a description should be used even when one doesn't appear in the texts being summarized.

A second potentially interesting application involves using the data and rules extracted by PROFILE for *language regeneration*. In [Radev and McKeown, 1998] we show how the conversion of extracted descriptions into components of a generation grammar allows for flexible (re)generation of new descriptions that don't appear in the source text. For example, a description can be replaced by a more general one, two descriptions can be combined to form a single one, or one long description can be deconstructed into its components, some of which can be reused as new descriptions.

We are also interested in investigating another idea, that of predicting the use of a description of an entity, even when the corresponding profile doesn't contain any description at all, or when it contains only descriptions that contain words that are not directly related to the words predicted by the rules of PROFILE. In this case, if the system predicts a semantic category that doesn't match any of the descriptions in a specific profile, two things can be done: (1) if there is a single description in the profile, pick that one, and (2) if there is more than one description, pick the one whose semantic representation (i.e., the corresponding row in the lexico-semantic matrix shown in Figure 8.2) is closest to the predicted semantic vector.

PROFILE can be also used for linking together alternative spellings or for identifying typographical errors in entity names. As unbelievable as it may seem, we were able to find almost a dozen spellings of the name of the Libyan president, including Moammar Gadhafi (UPI), Moamer Kadhafi (AFP), Moammar El Gheddafi, Moammar Gaddafi (Arabic News), Moammar Qadhafi (The Muslim Journal), Moammar Gadaffi (The Electronic Mail and Guardian, Johannesburg), Moammar Khaddafi (Global Intelligence Update), Moammar Khadafy, Moammar Kadafi, and Muammar Gaddafi! All of the above spellings appear in English-language news in a consistent way and were located by PROFILE as a result of the query "Libyan Leader". In our corpus we identified 251 groups of alternative spellings based on common descriptions (see Table 8.10). Some of them are shown in Table 8.11.

Finally, the profile extractor will be used as part of a large-scale, automat-

Entities	**Descriptions**
Larry Ellison	founder of Oracle Corp.
Lawrence Ellison	founder of Oracle Corp.

Table 8.10: Linking alternative spellings of the same entity.

Spelling 1	**Spelling 2**
Aimal Kasi	Aimal Kansi
Allen Ginsburg	Allen Ginsberg
Anatoly Solovyov	Anatoly Solvyev
Berge Ayvanzian	Berge Avyazin
David Oderkerken	David Odekerken
Eva Larue	Eva LaRue
Georgi Aniniev	Georgi Ananiev

Table 8.11: Alternative spellings and typos.

ically generated *Who's who* site which will be accessible both by users through a Web interface and by NLP systems through a client-server API.

Part III

Discussion, Related Work, Future Work, and Conclusion

This part concludes the thesis.

Chapter 9 discusses some issues related to evaluation and system status which were not treated separately in the body of the dissertation.

Chapter 10 presents a summary overview of related work.

In Chapter 11 we discuss potential application of the material presented in the thesis as well as some ideas for future work.

Finally, Chapter 12 serves as a conclusion to the dissertation by summarizing the contributions of this thesis to the area of Natural Language Processing.

Chapter 9

Evaluation and system status

9.1 Introduction

Throughout this thesis, we have interspersed a significant number of evaluation-related ideas and results. In this chapter, we will summarize these results. Given that there don't exist other systems that summarize multiple articles in the way that SUMMONS works, we found it very difficult to perform an adequate evaluation of the summaries generated by SUMMONS. We settled on some metrics that are discussed below and which could be used by rival systems to compare their performance to ours.

We have performed five separate evaluations of the different components of SUMMONS. Three of them (generation coverage, description extraction, and description reuse) were performed thoroughly, while two others (article clustering and new information finding) were done on relatively small data sets.

9.2 Coverage of the base summary generator

Currently, SUMMONS produces single-paragraph summaries consisting of 1–5 sentences. These summaries are limited to the MUC domain of terrorist news. The evaluation of the summary generator was based on the coverage of the CSTI corpus (see Chapter 3). During the development of SUMMONS, we kept the 1993 portion of the CSTI summaries in order to be able to use it as a benchmark. We manually built input template sets for all 62 summaries included in it. To quantify our coverage, we used as a metric the ability of SUMMONS to generate the summaries in CSTI. We defined four levels for this metric:

- A. exact coverage (same information in the output of SUMMONS as in CSTI and same wording),
- B. essentially exact coverage (same information, but different wording as well as occasional omission of side facts),
- C. partial coverage (ability to generate one or more of the sentences in the summary, but not all), and
- D. incorrect (inability to generate anything close in content or form to the summary in CSTI).

The results are summarized in Table 9.1. We believe that some error analysis would be helpful for the reader to understand the categories above.

Two summaries from CSTI which SUMMONS tried to produce and which we classified in the B and C category, respectively, are shown in Figure 9.1. The first summary produced by SUMMONS doesn't include the relative clause *as they slept*

CSTI: **29 December 1993, Algeria**. Terrorists murdered a Belgian husband and wife as they slept in their home in Bouira. The husband had his throat cut, and his wife was shot.
SUMMONS: Terrorists killed a Belgian husband and wife in their home in Bouira.
CSTI: **16 October 1993, Algeria**. Terrorists shot and killed two Russian military officers and wounded a third outside an apartment building near the Algerian military academy. The Russians were instructors at the academy.
SUMMONS: Terrorists killed two Russian military officers outside an apartment building near the Algerian military academy.

Figure 9.1: Two summaries from the CSTI corpus that SUMMONS could not reproduce fully.

Category	**Number**	**Percentage**
exact coverage	12	19.4%
essentially exact coverage	29	46.8%
partial coverage	14	22.5%
incorrect	7	11.3%

Table 9.1: Coverage of the CSTI 1993 corpus.

in their home in Bouira. In the second summary, an entire sentence, *The Russians were instructors at the academy*, is missing. Again SUMMONS has no way to produce it since that type of information is not present in the MUC templates.

The main reason for SUMMONS's relatively poor performance is the existence in CSTI of sentences that cannot be described using MUC templates. To be able to generate these sentences, other types of MUC templates need to be considered in addition to the current ones.

9.3 Extraction of descriptions

This section presents an evaluation of the description retrieval and reuse component.

At the current stage, the description generator extracts pre-modifiers and appositions as descriptions (relative clauses are not processed). In Section 7.4.1 we included the precision of the extraction of entity names. Similarly, we have computed the precision of retrieved 611 descriptions using randomly selected entities from the list retrieved in Section 7.4.1. Of the 611 descriptions, 551 (90.2%) were real descriptions. The others included a roughly equal number of cases of incorrect NP attachment and incorrect part-of-speech assignment.

We should add that our system currently doesn't handle entity cross-referencing. It will not realize that "Clinton" and "Bill Clinton" refer to the same person. Nor will it link a person's profile with the profile of the organization of which he is a member. We should note that extensive research in this field exists (e.g., [Wacholder *et al.*, 1997]). Using such technology will streamline the description retrieval process.

9.4 Description reuse

We presented a detailed evaluation of this component in Chapter 8. Here we include a summary of the results.

Our main evaluation metrics were the precision and recall associated with the ability of the system to predict the correct set of semantic constraints (Section 8.5) on the choice of a description. We used two methods: one in which we looked at the WORDNET nodes corresponding to each of the words, and another in which we looked at both the words and their parent nodes in WORDNET . The method that worked better (using knowledge of the parent nodes) achieved precision of 89.7% and recall of 64.6%. The alternative method achieved 87.2% precision and 63.9%

recall. Both methods were evaluated on unseen data only.

Chapter 10

Related work

This chapter discusses related work to three components of the thesis: summarization, information extraction for text generation, and language reuse and regeneration.

10.1 Text and data summarization

Previous work on automated text summarization falls into three main categories. In the first, full text is accepted as input and some percentage of the text is produced as output. This is often called "extraction". Typically, statistical approaches, augmented with key word or phrase matching, are used to identify which full sentences in the article can serve as a summary. Many schemes to rate sentences and methods for combining ratings exist [Paice, 1990, Kupiec *et al.*, 1995, Mani and Bloedorn, 1997, Lin, 1998]. Most of the work in this category produces a summary for a single article, although there are a few exceptions. The second two categories correspond to the two stages of processing that have to be carried out if sentence extraction is not used: analysis of the input document to process and re-represent information that should appear in a summary, and generation of a textual summary from a set of facts that are to be included. In this chapter, we first present work on sentence extraction, next turn to work on identifying information in an article that should appear in a summary, and conclude with work on generation of summaries from data.

There is a large body of work on the nature of abstracting from a library science point of view [Borko and Bernier, 1975]. This work distinguishes between different types of abstracts, most notably, *indicative* abstracts that tell what an article is about and *informative* abstracts that include major results from the article and can be read in place of it. SUMMONS generates summaries that are informative in nature. Research in psychology and education also focuses on how to teach people to write summaries (e.g., [Endres-Niggemeyer, 1993, Rothkegel, 1993]). This type of work can aid the development of summarization systems by providing insights into the human process of summarization.

10.1.1 Summarization through sentence extraction

To enable summarization in arbitrary domains, researchers have traditionally applied statistical techniques to identify and extract key sentences from an article using statistical techniques that locate important phrases [Luhn, 1958, Paice, 1990, Preston and Williams, 1994, Rau *et al.*, 1994]. This approach is often termed *extraction* rather than summarization.

This method has been successful in different domains [Preston and Williams, 1994] and is, in fact, the approach used in recent commercial summarizers (Apple [Boguraev and Kennedy, 1997], Microsoft, and inXight). In the newspaper article domain, Rau *et al.* (1994) report that extracts of individual news articles were rated lower by evaluators than summaries formed by simply using the lead sentence or two from the article. This follows the principle of the "inverted pyramid" in news writing, which puts the most salient information in the beginning of the article and leaves elaborations for later paragraphs, allowing editors to cut from the end of the text without compromising the readability of the remaining text.

Summaries that consist of sentences plucked from texts have been shown to be useful indicators of content, but they are also judged to be hard to read [Brandow *et al.*, 1990]. Paice [Paice, 1990] also notes that problems for this approach center around the fluency of the resulting summary. For example, extracted sentences may accidentally include pronouns which have no previous reference in the extracted text or, in the case of extracting several sentences, may result in incoherent text when the extracted sentences are not consecutive in the original text and do not naturally follow one another. Paice describes techniques for modifying the extracted text to replace unresolved references.

A more recent approach [Kupiec *et al.*, 1995] uses a corpus of articles with summaries to train a statistical summarization system. During training, the system identifies the features of text sentences that are typically also included in abstracts. In order to avoid problems noted by Paice, Kupiec's system produces an itemized list of sentences from the article, thus eliminating the implication that these sentences function together coherently as a full paragraph. As with the other statistical approaches, Kupiec's work is aimed at summarization of single articles.

Work presented at the 1997 ACL Workshop on Intelligent Scalable Text Summarization primarily focused on methods of sentence extraction. Alternatives to the use of frequency of key phrases included the identification and representation of lexical chains (sequences of semantically related words) [Halliday and Hasan, 1976] to find the major themes of an article followed by the extraction of one or two sentences per chain [Barzilay and Elhadad, 1997], training over the position of summary sentences in the full article (but also using word frequencies and other methods) [Hovy and Lin, 1997], and the construction of a graph of important topics to identify paragraphs that should be extracted [Mitra *et al.*, 1997].

10.1.2 Multi-document summarization

While most of the work in summarization focuses on summarization of single articles, early work is beginning to emerge on summarization across multiple documents. Researchers at Carnegie Mellon University [Carbonell and Goldstein, 1998] are developing statistical techniques to identify similar sentences and phrases across articles. Their aim is to identify sentences that are representative of more than one article.

Mani and Bloedorn [Mani and Bloedorn, 1997] link similar words and phrases from a pair of articles using WORDNET semantic relations. They show extracted sentences from the two articles side by side in the output.

While useful in general, sentence extraction approaches cannot handle the task that we address, aggregate summarization across *multiple documents*, since this requires reasoning about similarities and differences across documents to produce generalizations, or contradictions at a conceptual level. The best that such systems can do is to use word-vector similarity measures to identify related paragraphs and present them side by side.

10.1.3 Text generation for summarization

Summarization of numeric data using symbolic techniques has met with more success than summarization of text. Summary generation from database records is distinguished from the more traditional problem of text generation by the fact that summarization is concerned with conveying the maximal amount of information within minimal space. This goal is achieved through two distinct subprocesses, *conceptual* and *linguistic* summarization. Conceptual summarization is a form of content selection. It must determine which concepts out of a large number of concepts in the input should be included in the summary. Linguistic summarization is concerned with expressing that information in the most concise way possible.

Three systems developed at Columbia are related to SUMMONS. STREAK [Robin and McKeown, 1993, Robin, 1994, Robin and McKeown, 1995] generates summaries of basketball games, using a revision-based approach to summarization. It builds a first draft using fixed information that must appear in the summary (e.g., in basketball summaries, the score and who won and lost is always present). In a second pass, it uses revision rules to opportunistically add in information, as allowed by the form of the existing text. Using this approach, information that might otherwise appear as separate sentences gets added in as modifiers of the existing sentences or new words that can simultaneously convey both pieces of information are selected. PLANDOC [McKeown *et al.*, 1994, McKeown *et al.*, 1995, Shaw, 1995] generates summaries of the activities of telephone planning engineers, using linguistic summarization to both order its input messages and to combine them into single sentences. Focus has been on the combined use of conjunction, ellipsis and paraphrase to result in concise, yet fluent reports [Shaw, 1995, Shaw, 1998]. ZEDDOC [Passonneau *et al.*, 1997, Kukich *et al.*, 1997] generates Web traffic summaries for advertisement management software. It makes use of an ontology over the domain to combine information at the conceptual level.

All of these systems take tabular data as input. The research focus has been on linguistic summarization. SUMMONS, on the other hand, focuses on conceptual summarization of both structured and full-text data.

At least four previous systems developed elsewhere use natural language to summarize quantitative data, including ANA [Kukich, 1983b, Kukich, 1983a], SEMTEX [Rösner, 1987], FOG [Bourbeau *et al.*, 1990], and LFS [Iordanskaja *et al.*, 1994]. All of these use some forms of conceptual and linguistic summarization and the techniques can be adapted for our current work on summarization of multiple articles. In related work, Dalianis and Hovy [Dalianis and Hovy, 1993] have also looked at the problem of summarization, identifying eight aggregation operators (e.g., conjunction around noun phrases) that apply during generation to create more concise text.

Some of these systems take structured data as input and produce textual summaries as output while others do the opposite. In contrast, SUMMONS incorporates elements of both information extraction and text generation to produce summaries of multiple sources.

- ANA [Kukich, 1983a] produces textual descriptions of stock price changes.

- FOG [Bourbeau *et al.*, 1990] generates multilingual weather forecasts from

Input	Output	Example	Authors
textual	tabular	MUC systems	Lehnert *et al.* 96
tabular	textual	ANA , etc.	Kukich 83
		FOG	Bourbeau *et al.* 90
		STREAK	Robin 96
textual	textual	MUC +SUMMONS	Radev 98

Table 10.1: Comparison with related work.

language-independent descriptions.

- STREAK [Robin, 1994] deals with the generation of box office summaries of basketball games.

- CIRCUS [Fisher *et al.*, 1995] is one of the MUC systems that produce tabular summaries of news events in the form of templates.

Table 10.1 present a high-level comparison of ANA, FOG, STREAK, and CIRCUS.

The methodology developed for SUMMONS has as its ultimate goal the coupling of information extraction systems with text generation systems to build text-to-text summarizers. While SUMMONS is not currently hooked to CIRCUS as a live system, we have laid the groundwork for such complete text-to-text systems to be built in the near future.

10.2 Extraction of information for use in generation

Work in summarization using symbolic techniques has tended to focus more on identifying information in text that can serve as a summary [Young and Hayes, 1985, Rau, 1988, Hahn, 1990] as opposed to generating the summary, and often relies heavily on domain dependent scripts [DeJong, 1979, Tait, 1983]. The DARPA message understanding systems [MUC4, 1992], which process news articles in specific domains to extract specified types of information, also fall within this category. As output, work of this type produces templates that identify important pieces of information in the text, representing them as attribute-value pairs which could be part of a database entry. The message understanding systems, in particular, have been developed over a long period, have undergone repeated evaluation and development, including moves to new domains, and as a result, are relatively robust. In the domains for which templates have been designed, they achieve around 67% precision and recall on the task of extracting the values for the slots in the templates. They are impressive in their ability to handle large quantities of free-form text as input. As stand-alone systems, however, they do not address the task of summarization since they do not combine and rephrase extracted information as part of a textual summary.

A recent approach to symbolic summarization is being carried out at Cambridge University on identifying strategies for summarization [Sparck-Jones, 1993]. This work studies how various discourse processing techniques (e.g., rhetorical structure relations) can be used to both identify important information and form the actual summary. While promising, this work does not involve an implementation as of yet, but provides a framework and strategies for future work. [Marcu, 1997] uses a rhetorical parser to build rhetorical structure trees for arbitrary texts and produces a summary by extracting sentences that span the major rhetorical nodes of the tree.

In addition to domain specific information extraction systems, there has also been a large body of work on identifying people and organizations in text through proper noun extraction. These are domain independent techniques that can also be used to extract information for a summary. Techniques for proper noun extraction include the use of regular grammars to delimit and identify proper nouns [Mani *et al.*, 1993, Paik *et al.*, 1994], the use of extensive name lists, place names, titles and gazetteers in conjunction with partial grammars in order to recognize proper nouns as unknown words in close proximity to known words [Cowie *et al.*, 1992, Aberdeen *et al.*, 1992], statistical training to learn, for example, Spanish names, from online corpora [Ayuso *et al.*, 1992], and the use of concept-based pattern matchers that use semantic concepts as pattern categories as well as part-of-speech information [Weischedel *et al.*, 1993, Lehnert *et al.*, 1993]. In addition, some researchers have explored the use of both local context surrounding the hypothesized proper nouns [McDonald, 1993, Coates-Stephens, 1991b] and the larger discourse context [Mani *et al.*, 1993] to improve the accuracy of proper noun extraction when large known word lists are not available. In a way similar to this research, our work also aims at extracting proper nouns without the aid of large word lists. We use a regular grammar encoding part-of-speech categories to extract certain text patterns (descriptions) and we use WORDNET to provide semantic filtering.

Sam Coates-Stephens [Coates-Stephens, 1991b, Coates-Stephens, 1991a] was the first to analyze noun phrase descriptions of proper nouns. However, he stopped short of providing a detailed analysis of their automated extraction, their semantic classification, or use in generation.

10.3 Language reuse and regeneration

While the formal approach to LRR is our contribution, we should note that a large number of existing systems use techniques that we would classify as LRR.

Some classic work on phrasal lexicons [Kukich, 1983a] and [Jacobs, 1985] define phrasal templates to be reusable components which can be represented in their surface form in the generation lexicon.

Other examples of language reuse include collocation analysis [Smadja, 1991, Smadja and McKeown, 1991, Doerr, 1995] and summarization using sentence extraction [Paice, 1990, Kupiec *et al.*, 1995]. In the case of summarization through sentence extraction, the target text has the additional property of being a subtext of the source text. Other techniques that can be broadly categorized as language reuse are learning relations from on-line texts [Mitchell, 1997] and answering natural language questions using an on-line encyclopedia [Kupiec, 1993]. Kupiec's system, MURAX [Kupiec, 1993], is similar to ours from a different perspective. It extracts information from a text to serve directly in response to a user question. MURAX uses lexico-syntactic patterns, collocational analysis, along with information retrieval statistics, to find the encyclopedia entry that is most likely to serve as an answer to a user's wh-query. Ultimately, this approach could be used to extract information on items of interest in a user profile, where each question may represent a different point of interest. In our work, we also reuse strings (i.e., descriptions) as part of the summary, but the string that is extracted may be merged, or regenerated, as part of a larger textual summary.

Sato and Sato [Sato and Sato, 1998] use a system that is related to language generation: it uses syntactic transformations on user questions to find answers to user problems in a database of answers to frequently asked questions.

HealthDoc [DiMarco *et al.*, 1997, Hirst *et al.*, 1997] extracts sentences from a so-called "master document" and generates a concise "summary" of the master document's full document content. To do so, HealthDoc's sentence planner also needs to perform LRR in order to remove infelicities of phrasing and lexicalization due to segment combination. The sentence planner [Wanner and Hovy, 1996] contains a list of rules that operate on the internal representations, transforming them in order to ensure better output, in exactly the same way as the LRR operators do in Chapter 5.

Chapter 11

Applications and future work

This chapters contains three sections. The first two deal with suggested improvements to SUMMONS and to potential uses of the methodology embodied in it. A particularly interesting application, namely the generation of evolving summaries is already under way and deserves a section of its own.

11.1 Improvements to SUMMONS

SUMMONS is a prototype system. Even so, it serves as the springboard for research in a variety of directions. First and foremost, we realize the need for statistical techniques to increase the robustness and vocabulary of the system. Since we were looking for phrasings that mark summarization in a full article that includes other material as well, for a first pass we found it necessary to do a manual analysis in order to determine which phrases were used for summarization. In other words, we knew of no automatic way of identifying summary phrases. However, having an initial seed set of summary phrases might allow us to automate a second pass analysis of the corpus by looking for variant patterns of the ones we have found.

By using automated statistical techniques to find additional phrases, we could increase the size of the lexicon and use the additional phrases to identify new summarization strategies to add to our stock of operators.

One of the more important current goals is to increase the system's coverage by providing interfaces to a large number of on-line sources of news.

11.1.1 Multilingual extentions

Two applications seem particularly appealing:

- using descriptions extracted from multilingual corpora, organized around common entity names for machine translation of descriptions and
- summarization in one language of news written in another language using the template forms as an interlingual representation.

11.1.2 Trainability

The most urgent and potentially most useful addition to SUMMONS would be a module that links it with a trainable MUC system such as CRYSTAL (which is part of CIRCUS). Such a system must be trained to recognize the source of information in an article. An important problem that is likely to hinder short-term progress in connecting the two systems is the relatively low precision and recall values of MUC systems. Since a summary is built from multiple articles, precision and recall over the entire set will be significantly lower.

The planning operators in SUMMONS are currently applied following a heuristic ordering. There is, however, no evidence that one specific ordering is better than another. An interesting problem is to use machine learning techniques to learn the order in which the operators must be applied. The current declarative framework allows for the development of a SUMMONS-based API for development of customized multi-document summarizers.

11.1.3 Portability issues

An important issue is portability of SUMMONS to other domains and languages. Together with Efrat Levy, we ported a portion of the SUMMONS grammar to the domain of mergers and acquisitions. The process took a semester's work, mostly performed by an undergraduate student. We estimate that the following components need to be adapted to use SUMMONS in a new domain:

- the information extraction component (however, there already exist systems that can learn extraction rules for unrestricted domains[Lehnert *et al.*, 1993]; unfortunately, these systems have a very low accuracy),
- the lexicon and the generation grammar, and
- the planning operators.

On the other hand, the rest of the modules can be reused with only minor adaptation or no adaptation at all:

- the actual planning component (since the operators are declarative, when new operators are developed for a new domain, the planner can still be used),

- the description extractor,
- the language reuse module,
- the clustering module (our latest algorithm doesn't use the location of the event as a heuristic), and
- the new information finder.

We can envisage a general summarizer that contains modules tailored to individual domains of interest to users, such as baseball games, natural disasters, elections, and stock market changes. Building domain-specific components for such a system remains however outside the scope of this thesis.

11.2 Uses of SUMMONS

11.2.1 Testing MUC systems

Our summary generator could be used both for evaluation of message understanding systems by using the summaries to highlight differences between systems, and for identifying weaknesses in the current systems. We have already noted a number of drawbacks with the current output, which makes summarization more difficult, giving the generator less information to work with. For example, the output only sometimes indicates that a reference to a person, place, or event is identical to an earlier reference; there is no connection across articles; the source of the report is not included. Finally, the structure of the template representation is somewhat shallow, being closer to a database record than a knowledge representation. This means that the generator's knowledge of different features of the event and relations between them is somewhat shallow.

11.2.2 Language reuse and regeneration

In Chapter 7 we identified some characteristics of reusable text such as lifetime and contextual attachment. We did not propose any methods by which these can be computed. An interesting problem would be to identify similar metrics and to find a computational treatment of all these problems alike. The ultimate goal of LRR would be the creation, by fully automated means, of a large-scale database of reusable constructions to use in text generation, summarization, or question answering.

11.2.3 Digital newspaper

One of the projects at Columbia that is related to SUMMONS is the Columbia Digital News System (CDNS) [Aho *et al.*, 1998]. The goal of this project is to build a multi-media environment for intelligent news delivery and processing. The different components of CDNS are:

- image classification based on the caption surrounding the image,
- the recognition of the number of people in an image and their names based on the captions and on the entire text of the article that includes the image, and
- the generation of illustrated summaries using indexing and retrieval of related images.

11.3 Evolving summaries

To generate summaries of threads of articles, it is important to be able to do two things: identify which articles belong together (because they refer to the same event) and, assuming chronological order of articles within a cluster, decide which portion in more recent articles contains new information about the event that was not already conveyed in the earlier articles.

These two processes are called topic detection [Allan *et al.*, 1998] and new information detection [Radev, 1999], respectively. Our approach to topic detection is described in detail in [Radev *et al.*, 1999]. This section describes the concept of evolving multilingual summaries based on topic detection.

When summarizing clusters of articles, we noticed that often news writers repeat a large amount of information from one story to another. For example, Figures 11.1 and 11.2 show excerpts from two articles that were found to be in the same cluster by the module described in the previous chapter. The figures show the two paragraphs of the first story and the first five paragraphs of the second story (out of 18). (The full stories are shown in Appendix B.) One can notice that paragraphs 1 and 3 in the second story essentially convey the same information as paragraphs 1 and 2 in the first story, respectively. A multi-document summarizer which uses as input a set of documents with repeated information should try to remove the repetitions (per Grice's Maxim of Quantity [Grice, 1975]).

There are at least three reasons why this phenomenon occurs in news writing:

- when the earlier story served the purpose of breaking urgent news and the details are written in a follow-up story,

```
<DOCID> reute960109.0101 </DOCID>

...

<HEADER> reute 01-09 0057 </HEADER>

...

 German court convicts Vogel of extortion

    BERLIN, Jan 9 (Reuter) - A German court on Tuesday convicted
Wolfgang Vogel, the East Berlin lawyer famous for organising
Cold War spy swaps, on charges that he extorted money from
would-be East German emigrants.

    The Berlin court gave him a two-year suspended jail sentence
and a fine --- less than the 3 3/8 years prosecutors had sought.
```

Figure 11.1: Two paragraphs from the first story in the BERLIN cluster.

```
<DOCID> reute960109.0201 </DOCID>

...

<HEADER> reute 01-09 0582 </HEADER>

...

 East German spy-swap lawyer convicted of extortion

    BERLIN (Reuter) - The East Berlin lawyer who became famous
for engineering Cold War spy swaps, Wolfgang Vogel, was
convicted by a German court Tuesday of extorting money from East
German emigrants eager to flee to the West.

    Vogel, a close confidant of former East German leader Erich
Honecker and one of the Soviet bloc's rare millionaires, was
found guilty of perjury, four counts of blackmail and five
counts of falsifying documents.

    The Berlin court gave him the two-year suspended sentence
and a $63,500 fine. Prosecutors had pressed for a jail sentence
of 3 3/8 years and a $215,000 penalty.

    Vogel, 70, who got his start arranging the 1962 exchange of
U.S. pilot Gary Powers for Soviet spy Rudolf Abel, insisted his
only crime was trying to help unite people separated by the Cold
War division of Germany.

    ``The court said that I helped people --- what more can I
say?'' Vogel said after Judge Heinz Holzinger spent 90 minutes
reading the verdict to a packed courtroom.
```

Figure 11.2: The first five paragraphs from the second story in the BERLIN cluster.

- when the second story serves as a background to the first one,
- when the latter story adds new information to the story while keeping the user informed about earlier developments.

When news journalists know that *all* potential readers would have enough background on the event they do not repeat the background information. For example, because of the popularity of the Clinton/Lewinsky scandal, later stories rarely described how the entire scandal started. However, stories about developments on less talked about topics such as the Swissair Flight 111 crash and the bombings in Kenya and Tanzania typically included some information about the background of the story, such as the day and time when it occurred as well as the number of victims, the location of the crash or bombings, and the make of the airplanes and cars involved.

In generating summaries of clusters of articles on the same topic, one would obviously run across cases of repeated information. Again, if the summarizer keeps track of its interaction with a particular user, it doesn't need to include any information in the later summaries if that information has already been used in earlier summaries. We call this model an **evolving summary** and we will spend the rest of this chapter discussing some techniques that can be used to produce evolving summaries.

Definition 11 *An* **evolving summary** S_{k+1} *is the summary of a story, numbered* A_{k+1}*, when the stories numbered* A_1 *to* A_k *have already been processed and presented in a summarized form to the user. Summary* S_{k+1} *differs from its predecessor,* S_k*, because it contains new information and omits information from* S_k*.*

Note that we haven't implemented evolving summaries as of the writing of this thesis.

We believe that the ability to identify repeated information in clusters of stories can be helpful for both statistical and conceptual summarizers as the next two subsections attempt to show.

11.3.1 Statistical summarizers

Sentences that contain repeated information should be ignored or assigned low scores prior to sentence extraction. Our analysis shows that most of the repeated sentences appear in the first 2–3 paragraphs of a new story. Prior research has shown that these paragraphs are most likely to contain the sentences that would serve as the best summary of the article, when extracted. [Rau *et al.*, 1994] had suggested that these are the paragraphs that should be assigned the highest scores, it is obvious that the ability to weed out such sentences will help produce better evolving summaries. Other results (such as [Lin and Hovy, 1997]) indicate that in certain genres, sentences ordered after the first provide actually the best summaries. In that case, it is usually the first sentence of the second paragraph that is most relevant to the summary. Omitting sentences that contain repeated information can potentially boost the performance of summarizers such as the ones described in [Rau *et al.*, 1994] and [Lin and Hovy, 1997].

Similarly, being able to identify new vs. background information can help in producing better briefings (remember that briefings are defined to ignore background information). *New* and what is *background* information are defined informally in the following subsection.

11.3.2 Conceptual summarizers

The advantages of recognizing repeated information are not limited to sentence extraction. In the SUMMONS paradigm, one could run the MUC system only on text that has not been labeled as repeated.

11.3.3 Purpose of sentences

We have identified four classes of sentences (paragraphs) according to their purpose:

- **N**: New (breaking/current) information : e.g., the announcement of a plane crash right after the accident.
- **B**: Background information: e.g., a history of prior crashes by planes of the same company.
- **R**: Repeated information: e.g., a mention of the fact that the plane crashed appearing in subsequent stories which are primarily concerned with describing the development of the salvage operation.
- **O**: Other: in this class, we group anecdotal leads and quotes from participants in the investigation, as well as any other sentence not categorized in either the **N**, **B**, and **R** classes.

For the purpose of creating evolving summaries we decided that four problems are potentially worth investigating. We decided to concentrate on the most promising problem first.

- **N-type recognition**: highest priority — these sentences (or information extracted from them, in the case of conceptual summarization) should appear in the summary with the highest priority.

- **B-type recognition**: sentences of this class will be assigned low priority before summarizing the story that contains them and hence they will be ignored by the summarizer.

- **O-type recognition**: we consider these sentences the least important to summarization. The summarizer should ignore them.

- **R-type recognition**: these sentences should not be processed if the system knows that the user has already seen summaries produced based on the earlier instances of related sentences.

We decided to focus on the fourth of these problems, namely the binary classification of paragraphs in clusters into R-type and not-R-type paragraphs. For this purpose, we annotated manually a corpus of clusters of news stories and used a portion of it for developing a method for R-type labeling. We used the rest of the corpus (unseen during training) for evaluation.

11.3.4 Methodology

Our goal was to identify related paragraphs in each cluster of articles. Our initial thought was to focus primarily on linguistic and stylistic features (such as the presence of quotes and proper nouns in different paragraphs). However, after a few experiments, we discovered that a simple statistical method, similar to the one that we used in [Radev *et al.*, 1999] achieves the best results. The rest of this section describes our experiments.

11.3.5 Examples and discussion

For illustration of our approach, we will use the four stories in the cluster shown in Appendix B (we remind the reader that NIF1 was used for the actual clustering). The number of paragraphs in the four stories are 2, 18, 7, and 8, respectively.

For the rest of this section, we will refer to each group of related paragraphs within a cluster as a **group of related paragraphs**. The chronologically first paragraph in a group will be called the **original** while the remaining ones will be referred to as the **copies** of the original. Of course, these paragraphs are not identical copies of the original, they are simply highly similar to it. In the experiment, all documents came from the same source. In the case of multiple sources, it may be the case that one of the sources will be preferred in summarization due to its shorter length.

11.3.6 Algorithm used

We used a simple cosine measure to find related sentences or paragraphs. When we compare paragraphs from related articles, we consider the similarity between the paragraphs in the different documents. If the similarity SIM is above an experimentally set threshold, two paragraphs are considered to be related. Note that all similarities are computed on a "bag of words" basis only.

Original	Copies
1	3 21 28
2	5 26 32
4	25 31
6	27 35
10	23

Table 11.1: System output on the Berlin cluster.

		System	
		R-type	not-R-type
Human	R-type	9	2
	non-R-type	1	23

Table 11.2: Evaluation of R-type recall and precision in the Berlin cluster.

11.3.7 Results

When we ran our algorithm on the Berlin cluster, we obtained 24 groups of related paragraphs. Obviously, the first paragraph of each group (also 24 in total) is labeled as not-R-type, while the remaining 11 paragraphs are marked to be of R-type. The system and model comparison is displayed in Table 11.1. Table 11.2 shows the contingency table used to measure precision and recall for R-type classification in the Berlin example. The corresponding precision is $10/11 = 90.9\%$ and recall is $9/10 = 90.00\%$.

11.4 Other suggestions for future work

We would like to conclude the list of suggestions for future work with the following:

- aligning different accounts of the same event. These accounts can be either central to particular articles or just brief mentions of the event in an article on a related topic.

- hypertext data mining. In a way similar to the extraction of descriptions, one can design a method for learning relations other than the Entity - Description relation.

- evolving summaries. When a user has already seen a summary of an event up to a given point in time, and new information arrives through the newswire, the summaries generated no longed need to include information that was already presented to the user and can instead focus on summarizing the new information only.

Chapter 12

Conclusion

12.1 Introduction

This thesis presented a case study in the generation of summaries from multiple sources of information. The work continues previous research in information extraction, conceptual reasoning, and text generation.

Our prototype system, SUMMONS, demonstrates the feasibility of generating briefings of a series of domain-specific news articles on the same event, highlighting changes over time as well as similarities and differences among sources and including some historical information about the participants. The ability to automatically provide summaries of heterogeneous material will critically help in the effective use of the Internet in order to avoid overload with information. We show how planning operators can be used to synthesize summary content from a set of templates, each representing a single article. These planning operators are empirically based, coming from analysis of existing summaries, and allow for the generation of concise briefings. Our framework allows for experimentation with summaries of different lengths and for the combination of multiple, independent summary operators to produce complex summaries with added descriptions.

The major contribution of SUMMONS is to explore a logical extension to the MUC systems, namely summarization of events using templates that represent expected types of information.

Our theoretical and practical contributions to information extraction, conceptual reasoning, and text generation are highlighted in the next two sections.

12.2 Theoretical and methodological contributions

- **Detecting contradictions, agreement, and generalization among sources:** We use symbolic, declarative operators which can be extended to other domains.

- **Planning operators for multiple documents and multiple sources:** We combine information from multiple news sources on the same topic at the conceptual level into a coherent and self-contained briefing.

- **Language reuse and regeneration:** We define these dual concepts, build modules that implement them, and identify applications for them.

- **Correlations between context and pragmatics on the one hand and lexical choice on the other:** We define a model of the relationship between pragmatics, semantics, and the lexicon. We use this model to show that context and other linguistic indicators correlate with the choice of a particular noun phrase to describe an entity. Using machine learning techniques from a very large corpus, we automatically extract a large set of rules that predict the choice of a description out of an entity profile. We show that high-precision automatic prediction of an appropriate description in a specific context is possible.

12.3 Technical contributions

- **Generation of briefings:** We have built a system that generates briefings in natural language from a template-based input.

- **Machine learning algorithm for establishing constraints on lexical choice:** We have implemented a machine learning algorithm for selecting a description of an entity based on the context in which it appears.

- **Semantic categorization of descriptions:** We have developed techniques for semantic categorization of description noun phrases using WORDNET.

- **Automated building of knowledge sources for text generation:** We have developed a Web-based search engine for entity and description pairs which can be extended to extract additional relations from the Web.

Index

Bibliography

[Aberdeen *et al.*, 1992] John Aberdeen, John Burger, Dennis Connolly, Susan Roberts, and Marc Vilain. MITRE-Bedford: Description of the ALEMBIC system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 215–222, McLean, Virginia, June 1992.

[Agency, 1997] Central Intelligence Agency. The CIA World Factbook. URL: http://www.odci.gov/cia/publications/factbook/country.html, 1997.

[Aho *et al.*, 1998] Alfred Aho, Shih-Fu Chang, Kathleen McKeown, Dragomir Radev, John Smith, and Kazi Zaman. Columbia Digital News Project. *International Journal of Digital Libraries*, 1(4):377–385, 1998.

[Allan *et al.*, 1998] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study — Final Report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, February 1998.

[AltaVista, 1998] Altavista search engine. WWW site, URL: http://-altavista.digital.com, 1998.

[Ayuso *et al.*, 1992] Damaris Ayuso, Sean Boisen, Heidi Fox, Herb Gish, Robert Ingria, and Ralph Weischedel. BBN: Description of the plum system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 169–176, McLean, Virginia, June 1992.

[Bach, 1986] E. Bach. The algebra of events. *Linguistics and philosophy*, 9:5–16, 1986.

[Barzilay and Elhadad, 1997] Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 10–17, Madrid, Spain, August 1997. Association for Computational Linguistics.

[Bender *et al.*, 1996] W. Bender, P. Chesnais, S. Elo, A. Shaw, and M. Shaw. Enriching communities: Harbingers of news in the future. *IBM Systems Journal*, 35(3&4):369–380, 1996.

[Berners-Lee, 1992] Tim Berners-Lee. World-Wide Web: The information universe. *Electronic Networking*, 2(1):52–58, 1992.

[Boguraev and Kennedy, 1997] Branimir Boguraev and Christopher Kennedy. Salience-based content characterization of text documents. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 2–9, Madrid, Spain, July 1997. Association for Computational Linguistics.

[Borko and Bernier, 1975] Harold Borko and Charles Bernier. *Abstracting Concepts and Methods*. Academic Press, New York, N.Y., 1975.

[Bourbeau *et al.*, 1990] Laurent Bourbeau, Denis Carcagno, E. Goldberg, Richard Kittredge, and Alain Polguère. Bilingual generation of weather forecasts in an operations environment. In Hans Karlgren, editor, *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, volume 3, pages 318–320, Helsinki, Finland, 1990.

[Brandow *et al.*, 1990] Ronald Brandow, Karl Mitze, and Lisa F. Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 26:135–170, 1990.

[Carbonell and Goldstein, 1998] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings, 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, Melbourne, Australia, August 1998.

[Church, 1988] Kenneth W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing (ANLP-88)*, pages 136–143, Austin, Texas, February 1988. Association for Computational Linguistics.

[ClariNet, 1998] ClariNet. WWW site, URL: http://www.clari.net, 1998.

[CNN, 1998] CNN interactive. WWW site, URL: http://www.cnn.com, 1998.

[Coates-Stephens, 1991a] Sam Coates-Stephens. Automatic acquisition of proper noun meanings. In Zbigniew W. Ras and Maria Zemankova, editors, *Methodologies for Intelligent Systems*. Springer-Verlag, 1991.

[Coates-Stephens, 1991b] Sam Coates-Stephens. Automatic lexical acquisition using within-text descriptions of proper nouns. In *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*, pages 154–169, 1991.

[Cohen, 1995] William W. Cohen. Fast effective rule induction. In *Proc. 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[Cohen, 1996] William W. Cohen. Learning trees and rules with set-valued features. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 709–716, Menlo Park, August 1996. AAAI Press / MIT Press.

[Cowie *et al.*, 1992] Jim Cowie, Louise Guthrie, Yorick Wilks, James Pustejovsky, and Scott Waterman. CRL/NMSU and Brandeis: Description of the *mucbruce* system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 223–232, McLean, Virginia, June 1992.

[Cuts, 1994] Short Cuts. Science and technology section. *Economist*, 17:85–86, December 1994.

[Dale, 1992] Robert Dale. *Generating Referring Expressions.* ACL-MIT Press Series in Natural Language Processing, Cambridge, Massachusetts, 1992.

[Dalianis and Hovy, 1993] Hercules Dalianis and Eduard Hovy. Aggregation in natural language generation. *Proceedings of the 4th European Workshop on Natural Language Generation*, 1993.

[Danlos, 1987] Laurence Danlos. *The Linguistic Basis of Text Generation.* Cambridge University Press, Cambridge, England, 1987.

[DejaNews, 1998] Dejanews. WWW site, URL: http://www.dejanews.com, 1998.

[DeJong, 1979] Gerald F. DeJong. *Skimming stories in real time: an experiment in integrated understanding.* PhD thesis, Yale University Department of Computer Science, New Haven, Connecticut, 1979.

[DiMarco *et al.*, 1997] Chrysanne DiMarco, Graeme Hirst, and Eduard Hovy. Generation by selection and repair as a method for adapting text for the individual reader. In *Proceedings of the Workshop on Flexible Hypertext, 8th ACM International Hypertext Conference*, Southampton, England, 1997.

[Doerr, 1995] Rita McCardell Doerr. *A Lexical-semantic and Statistical Approach to Lexical Collocation Extraction for Natural Language Generation.* Ph.d. thesis, University of Maryland Baltimore County, Baltimore, MD, USA, 1995.

[Duford, 1993] Darrin Duford. CREP: a regular expression-matching textual corpus tool. Technical Report CUCS-005-93, Columbia University, 1993.

[Elhadad, 1991] Michael Elhadad. FUF: The universal unifier - user manual, version 5.0. Technical Report CUCS-038-91, Columbia University, 1991.

[Elhadad, 1993] Michael Elhadad. *Using argumentation to control lexical choice: a unification-based implementation.* PhD thesis, Computer Science Department, Columbia University, 1993.

[Endres-Niggemeyer, 1993] Brigitte Endres-Niggemeyer. An empirical process model of abstracting. In *Workshop on Summarizing Text for Intelligent Communication*, Dagstuhl, Germany, December 1993.

[Evans and Zhai, 1996] David A. Evans and Chengxiang Zhai. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Conference of the ACL*, Santa Cruz, California, 1996. Association for Computational Linguistics.

[Fisher *et al.*, 1995] David Fisher, Stephen Soderland, Joseph McCarthy, Fangfang Feng, and Wendy Lehnert. Description of the UMass system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 221–236, 1995.

[Grice, 1975] H.P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, New York, 1975.

[Grishman *et al.*, 1992] R. Grishman, C. Macleod, and J. Sterling. New York University: Description of the PROTEUS system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference*, June 1992.

[Hahn, 1990] Udo Hahn. Topic parsing: accounting for text macro structures in full-text analysis. *Information Processing and Management*, 26:135–170, 1990.

[Halliday and Hasan, 1976] Michael Halliday and Ruqaiya Hasan. *Cohesion in English*. English Language Series. Longman, London, 1976.

[Halliday, 1985] M.A.K. Halliday. *An Introduction to Functional Grammar*. Edward Arnold, London, 1985.

[Hand, 1997] Thérèse F. Hand. A proposal for task-based evaluation of text summarization systems. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 31–38, Madrid, Spain, July 1997. Association for Computational Linguistics.

[Hirst *et al.*, 1997] Graeme Hirst, Chrysanne DiMarco, Eduard Hovy, and K. Parsons. Authoring and generating health-education documents that are tailored to the needs of the individual patient. In A. Jameson, C. Paris, and C. Tasso, editors, *Proceedings of the Sixth International Conference on User Modeling (UM97)*, pages 107–118, Sardinia, Italy, 1997.

[Hovy and Lin, 1997] Eduard Hovy and Chin-Yew Lin. Automated text summarization in SUMMARIST. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 18–24, Madrid, Spain, July 1997. Association for Computational Linguistics.

[Hovy, 1987] Eduard Hovy. *Generating Natural Language under Pragmatic Constraints.* PhD thesis, Yale University Department of Computer Science, New Haven, Connecticut, 1987.

[Hovy, 1988] Eduard H. Hovy. Planning coherent multisentential text. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, N.Y., June 1988. Association for Computational Linguistics.

[Iordanskaja *et al.*, 1994] Lidija Iordanskaja, M. Kim, Richard Kittredge, Benoit Lavoie, and Alain Polguère. Generation of extended bilingual statistical reports. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan, 1994.

[Jacobs, 1985] Paul Jacobs. *A knowledge-based approach to language production.* PhD thesis, University of California, Berkeley, California, 1985.

[Jing *et al.*, 1998] Hongyan Jing, Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Summarization evaluation methods: Experiments and analysis. In *Symposium on Intelligent Text Summarization*, pages 60–68, Stanford, California, March 1998. American Association for Artificial Intelligence.

[Kaufman and Rousseeuw, 1990] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley Series in Probability and Mathematical Statistics. Wiley, New York, 1990.

[Knight and Chander, 1994] Kevin Knight and Ishwar Chander. Automated postediting of documents. In *Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1*, pages 779–784, Menlo Park, CA, USA, july 31 - august 4 1994. AAAI Press.

[Kukich *et al.*, 1997] Karen Kukich, Rebecca Passonneau, Kathleen McKeown, Dragomir Radev, Vasileios Hatzivassiloglou, and Hongyan Jing. Software reuse and evolution in text generation applications. In *ACL/EACL Workshop - From Research to Commercial Applications: Making NLP Technology Work in Practice*, Madrid, Spain, July 1997.

[Kukich, 1983a] Karen Kukich. *Knowledge-based report generation: a knowledge engineering approach to natural language report generation.* PhD thesis, University of Pittsburgh, Pittsburgh, Pennsylvania, 1983.

[Kukich, 1983b] Karen K. Kukich. Design of a knowledge-based report generator. In *Proceedings of the 21st Annual Meeting of the ACL*, pages 145–150, Cambridge, Massachusetts, June 1983. Association for Computational Linguistics.

[Kupiec *et al.*, 1995] Julian M. Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, Seattle, Washington, July 1995.

[Kupiec, 1993] Julian M. Kupiec. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings, 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993.

[Lawrence and Giles, 1998] Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, 280(5360):98, 1998.

[Lehnert *et al.*, 1993] Wendy Lehnert, Joe McCarthy, Stephen Soderland, Ellen Riloff, Claire Cardie, Jonathan Peterson, and Fangfang Feng. UMass/Hughes: Description of the CIRCUS system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 277–291, Baltimore, Md., August 1993.

[Lin and Hovy, 1997] Chin-Yew Lin and Eduard Hovy. Identifying topics by position. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing (ANLP-97)*, pages 283–290, Washington, D.C., April 1997.

[Lin, 1998] Chin-Yew Lin. Assembly of topic extraction modules in summarist. In *Symposium on Intelligent Text Summarization*, pages 53–59, Stanford, California, March 1998. American Association for Artificial Intelligence.

[Luhn, 1958] Hans P. Luhn. The automatic creation of literature abstracts. *IBM Journal*, pages 159–165, 1958.

[Lycos, 1998] Lycos, Inc. home page. WWW site, URL: http://www.lycos.com, 1998.

[Mani and Bloedorn, 1997] Inderjeet Mani and Eric Bloedorn. Multi-document summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 622–628, Providence, Rhode Island, 1997. American Association for Artificial Intelligence.

[Mani *et al.*, 1993] Inderjeet Mani, Richard T. Macmillan, Susann Luperfoy, Elaine Lusher, and Sharon Laskowski. Identifying unknown proper names in newswire text. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 44–54, Columbus, Ohio, June 1993. Special Interest Group on the Lexicon of the Association for Computational Linguistics.

[Marcu, 1997] Daniel Marcu. From discourse structures to text summaries. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain, July 1997. Association for Computational Linguistics.

[Marcu, 1998] Daniel Marcu. To build text summaries of high quality, nuclearity is not sufficient. In *Symposium on Intelligent Text Summarization*, pages 1–8, Stanford, California, March 1998. American Association for Artificial Intelligence.

[McDonald and Pustejovsky, 1986] David D. McDonald and James D. Pustejovsky. Description-directed natural language generation. In *Proceedings of the 9th IJCAI*, pages 799–805. IJCAI, 1986.

[McDonald, 1993] David D. McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 32–43, Columbus, Ohio, June 1993. Special Interest Group on the Lexicon of the Association for Computational Linguistics.

[McKeown and Radev, 1995] Kathleen R. McKeown and Dragomir R. Radev. Generating summaries of multiple news articles. In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82, Seattle, Washington, July 1995.

[McKeown *et al.*, 1994] Kathleen R. McKeown, Karen Kukich, and James Shaw. Practical issues in automatic documentation generation. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, October 1994. Association for Computational Linguistics.

[McKeown *et al.*, 1995] Kathleen R. McKeown, Jacques Robin, and Karen Kukich. Generating concise natural language summaries. *Journal of Information Processing and Management*, 31(5):703–733, 1995.

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

[Miller *et al.*, 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, 3(4):235–312, 1990.

[Mitchell, 1997] Tom M. Mitchell. Does machine learning really work? *AI Magazine*, 18(3), 1997.

[Mitra *et al.*, 1997] Mandar Mitra, Amit Singhal, and Chris Buckley. Automatic text summarization by paragraph extraction. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 39–46, Madrid, Spain, July 1997. Association for Computational Linguistics.

[Moore and Paris, 1989] Johanna Moore and Cécile Paris. Planning text for advisory dialogues. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 203–211, Vancouver, B.C., June 1989. Association for Computational Linguistics.

[MUC4, 1992] Proceedings of the fourth message understanding conference (MUC-4). DARPA Software and Intelligent Systems Technology Office, 1992.

[NetSumm, 1998] Netsumm home page. WWW site, URL: http://-www.labs.bt.com/innovate/informat/netsumm/index.htm, 1998.

[NYT, 1998] New York Times. WWW site, URL: http://www.nytimes.com, 1998.

[Paice and Jones, 1993] Chris D. Paice and Paul A. Jones. The identification of important concepts in highly structured technical papers. pages 69–78. ACM-SIGIR 1993, 1993.

[Paice, 1990] Chris Paice. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26:171–186, 1990.

[Paik *et al.*, 1994] Woojin Paik, Elizabeth D. Liddy, Edmund Yu, and Mary McKenna. Interpretation of proper nouns for information retrieval. In *Proceedings of the Human Language Technology Workshop*, pages 309–313, Plainsboro, New Jersey, March 1994. ARPA Software and Intelligent Systems Technology Office, Morgan Kaufmann.

[Passonneau *et al.*, 1997] Rebecca Passonneau, Karen Kukich, Kathleen McKeown, Dragomir Radev, and Hongyan Jing. Summarizing Web traffic: A portability exercise. Technical Report CUCS-009-97, Columbia University, Department of Computer Science, New York, NY, USA, March 1997.

[PGT97, 1997] Patterns of global terrorism. U.S. Dept. of State Publication, 1997.

[Preston and Williams, 1994] Keith Preston and Sandra Williams. Managing the information overload. *Physics in Business*, June 1994.

[Probert, 1998] Matthew Probert. The probert encyclopedia. WWW site, URL: http://www.servilesoftware.ndirect.co.uk/, 1998.

[Quirk *et al.*, 1985] Randolph Quirk, Sidney Greenbaum, Leech Geoffrey, and Jan Svartvik. *A Comprehensive Grammar of the English Language*. Longman, London and New York, 1985.

[Radev and McKeown, 1997] Dragomir R. Radev and Kathleen R. McKeown. Building a generation knowledge source using internet-accessible newswire. In

Proceedings of the 5th Conference on Applied Natural Language Processing, pages 221–228, Washington, DC, April 1997.

[Radev and McKeown, 1998] Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, September 1998.

[Radev *et al.*, 1999] Dragomir R. Radev, Vasileios Hatzivassiloglou, and Kathleen R. McKeown. A description of the CIDR system as used for TDT-2. In *DARPA Broadcast News Workshop*, Herndon, VA, March 1999.

[Radev, 1996] Dragomir R. Radev. An architecture for distributed natural language summarization. In *Proceedings of the 8th International Workshop on Natural Language Generation: Demonstrations and Posters*, pages 45–48, Herstmonceux, England, June 1996.

[Radev, 1999] Dragomir R. Radev. Topic shift detection - finding new information in threaded news. submitted to ACL'99, 1999.

[Rau *et al.*, 1989] Lisa F. Rau, Paul S. Jacobs, and Uri Zernick. Information extraction and text summarization using linguistic knowledge extraction. *Information Processing and Management*, 25(4):419–428, 1989.

[Rau *et al.*, 1992] Lisa Rau, George Krupka, Paul Jacobs, Ira Sider, and Lois Childs. GE NLToolset: MUC-4 test results and analysis. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 94–99, McLean, Virginia, June 1992.

[Rau *et al.*, 1994] Lisa F. Rau, Ron Brandow, and Karl Mitze. Domain-Independent summarization of news. In *Summarizing Text for Intelligent Communication*, pages 71–75, Dagstuhl, Germany, 1994.

[Rau, 1988] Lisa F. Rau. Conceptual information extraction and information retrieval from natural language input. In *Proceedings, RIAO-88: Conference on User-Oriented, Content-Based, Text and Image Handling*, pages 424–437, Cambridge, Massachusetts, 1988.

[Reuters, 1998] Reuters news. WWW site, URL: http://www.yahoo.com/-headlines, 1998.

[Riloff and Lehnert, 1994] Ellen Riloff and Wendy Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, July 1994.

[Robin and McKeown, 1993] Jacques Robin and Kathleen R. McKeown. Corpus analysis for revision-based generation of complex sentences. In *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington, D.C., July 1993.

[Robin and McKeown, 1995] Jacques Robin and Kathleen R. McKeown. Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence Journal*, 1995. In press.

[Robin, 1994] Jacques Robin. *Revision-Based Generation of Natural Language Summaries Providing Historical Background: Corpus-Based Analysis, Design,* *Implementation, and Evaluation.* PhD thesis, Department of Computer Science, Columbia University, New York, 1994.

[Rösner, 1987] Dietmar Rösner. SEMTEX: A text generator for German. In Gerard Kempen, editor, *Natural Language Generation: New Results in Artificial Intellligence, Psychology, and Linguistics.* Martinus Ninjhoff Publishers, 1987.

[Rothkegel, 1993] Annely Rothkegel. Abstracting from the perspective of text-production. In *Workshop on Summarizing Text for Intelligent Communication*, Dagstuhl, Germany, December 1993.

[Salton and McGill, 1983] Gerard Salton and Michael McGill. *Introduction to Modern Information Retrieval.* Computer Series. McGraw Hill, New York, 1983.

[Sato and Sato, 1998] Satoshi Sato and Madoka Sato. Rewriting saves extracted summaries. In *Symposium on Intelligent Text Summarization*, pages 85–92, Stanford, California, March 1998. American Association for Artificial Intelligence.

[Shaw, 1995] James Shaw. Conciseness through aggregation in text generation. In *Proceedings of the 33rd ACL (Student Session)*, pages 329–331, 1995.

[Shaw, 1998] James Shaw. Segregatory coordination and ellipsis in text generation. In *Proceedings of the Joint 17th International Conference on Computational Linguistics 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98)*, pages 1220–1226, Montreal, Canada, August 1998.

[Smadja and McKeown, 1991] Frank Smadja and Kathleen R. McKeown. Using collocations for language generation. *Computational Intelligence*, 7(4), December 1991.

[Smadja, 1991] Frank Smadja. *Retrieving Collocational Knowledge from Textual Corpora. An Application: Language Generation.* PhD thesis, Department of Computer Science, Columbia University, New York, NY, 1991.

[Soderland *et al.*, 1995] Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the the 14th IJCAI*, pages 1314–1321. IJCAI, 1995.

[Soderland, 1997] Stephen Soderland. Learning to extract text-based information from the World Wide Web. In David Heckerman, Heikki Mannila, Daryl Pregibon, and Ramasamy Uthurusamy, editors, *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, page 251. AAAI Press, 1997.

[Sparck-Jones, 1993] Karen Sparck-Jones. What might be in a summary? In *Proceedings of Information Retrieval 93: Von der Modellierung zur Anwendung*, pages 9–26, Universitatsverlag Konstanz, 1993.

[Sparck-Jones, 1998] Karen Sparck-Jones. Automatic summarising: factors and directions. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization.* MIT press, to appear, 1998.

[Sundheim, 1992] Beth M. Sundheim. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 3–21, McLean, Virginia, June 1992.

[Tait, 1983] John I. Tait. *Automatic summarising of English texts.* PhD thesis, University of Cambridge, Cambridge, England, 1983.

[Wacholder *et al.*, 1997] Nina Wacholder, Yael Ravin, and Misook Choi. Disambiguation of proper names in text. In *Proceedings of the Fifth Applied Natural Language Processing Conference*, Washington, D.C., 1997. Association for Computational Linguistics.

[Wanner and Hovy, 1996] Leo Wanner and Eduard Hovy. The HealthDoc sentence planner. In *Proceedings of the 8th International Workshop on Natural Language Generation*, pages 1–10, Herstmonceux, England, June 1996.

[Weischedel *et al.*, 1993] Ralph Weischedel, Damaris Ayuso, Sean Boisen, Heidi Fox, Robert Ingria, Tomoyoshi Matsukawa, Constantine Papageorgiou, Dawn MacLaughlin, Masaichiro Kitagawa, Tsutomu Sakai, June Abe, Hiroto Hosihi, Yoichi Miyamoto, and Scott Miller. BBN: Description of the PLUM system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 93–108, Baltimore, Md., August 1993.

[Young and Hayes, 1985] S.R. Young and P.J. Hayes. Automatic classification and summarization of banking telexes. In *Proceedings of the Second Conference on Artificial Intelligence Applications*, pages 402–408, 1985.

Appendix A

semhier.terrorist

A.1 Introduction

This appendix includes the raw representation of the terrorism domain ontology. It is taken from the CIRCUS system developed at the University of Massachusetts [Fisher *et al.*, 1995] and is reprinted here with permission.

A.2 Code

```
Root_Class              Root_Class
Entity                  Root_Class
Event                   Root_Class
### Times
Time-Period             Entity
ws_Date                 Time-Period
ws_Absolute_Date        ws_Date
```

```
ws_Relative_Date        ws_Date
ws_Duration             ws_Date
### Places
Location                Entity
### Activities
Attack                  Event
Bombing                 Attack
Murder                  Attack
Kidnapping              Attack
Robbery                 Attack
Injury                  Attack
### Media
Media                   Entity
### People
Human                   Entity
Proper-Name             Human
Human-Title             Human
Military-Title          Human
Terrorist               Human
Human-Target            Human
Civilian                Human-Target
Clergy                  Civilian
Diplomat                Human-Target
Govt-Official           Human-Target
```

```
Former-Govt-Official    Human-Target
Former-Active-Military  Human-Target
Legal-Or-Judicial       Human-Target
### Not a target via guidelines
Active-Military         Human
Politician              Human-Target
Law-Enforcement         Human-Target
Security-Guard          Human-Target
###
Political               Entity
Organization            Entity
Terrorist-Organization  Organization
Govt-Organization       Organization
Military-Organization   Organization
Legal-Organization      Organization
Law-Enforcement-Organization    Organization
Religious-Organization  Organization
### Targets
Phys-Target             Entity
Generic-Loc             Phys-Target
Terrorist-Phys-Target   Phys-Target
Military-Phys-Target    Phys-Target
Building                Phys-Target
Church                  Building
```

```
Civilian-Residence      Building
Commercial              Phys-Target
Communications          Phys-Target
Diplomat-Office-Or-Residence Building
Financial               Building
Govt-Office-Or-Residence Building
Law-Enforcement-Facility Building
Politician-Office-Or-Residence Building
Organization-Office     Building
School                  Building
Energy                  Phys-Target
Transport-Vehicle       Phys-Target
Transport-Facility      Phys-Target
Transport-Route         Phys-Target
Water                   Phys-Target
### Stuff
Money                   Entity
Property                Entity
### Instruments
Weapon                  Entity
Gun                     Weapon
Machine-Gun             Gun
Mortar                  Gun
Handgun                 Gun
```

Rifle	Gun
Explosive	Weapon
Bomb	Explosive
Vehicle-Bomb	Bomb
Dynamite	Bomb
Mine	Bomb
Grenade	Explosive
Molotov-Cocktail	Explosive
Projectile	Weapon
Missile	Projectile
Rocket	Projectile
Aerial-Bomb	Weapon
Cutting-Device	Weapon
Fire	Weapon
Stone	Weapon
Torture	Weapon

Created by the Natural Language Processing Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
under the direction of Professor Wendy G. Lehnert

Appendix B

Berlin stories

B.1 Introduction

The four stories shown here are extracted from the North-American News Corpus. They are included as an illustration to Chapter 7.

B.2 Story Number 02 (BERLIN/960109.0101)

```
<DOCID> reute960109.0101 </DOCID>
<STORYID cat=i pri=b> a0586 </STORYID>
<FORMAT> &D3; &D1; </FORMAT>
<KEYWORD> BC-GERMANY-VOGEL-VERDICT </KEYWORD>
<HEADER> reute 01-09 0057 </HEADER>
<SLUG> BC-GERMANY-VOGEL-VERDICT </SLUG>
<HEADLINE>
 German court convicts Vogel of extortion
```

</HEADLINE>

<TEXT>

<p>

BERLIN, Jan 9 (Reuter) - A German court on Tuesday convicted Wolfgang Vogel, the East Berlin lawyer famous for organising Cold War spy swaps, on charges that he extorted money from would-be East German emigrants.

<p>

The Berlin court gave him a two-year suspended jail sentence and a fine --- less than the 3 3/8 years prosecutors had sought.

MORE

</TEXT>

<TRAILER>

Reut07:11 01-09-96

</TRAILER>

</DOC>

<DOC>

B.3 Story Number 03 (BERLIN/960109.0201)

<DOCID> reute960109.0201 </DOCID>

<STORYID cat=i pri=u> a1163 </STORYID>

<FORMAT> &D3; &D1; </FORMAT>

<KEYWORD> BC-GERMANY-VOGEL </KEYWORD>

<HEADER> reute 01-09 0582 </HEADER>

<SLUG> BC-GERMANY-VOGEL (SCHEDULED, PICTURE) </SLUG>

<HEADLINE>

East German spy-swap lawyer convicted of extortion

</HEADLINE>

<BYLINE> By Deborah Cole </BYLINE>

<TEXT>

<p>

BERLIN (Reuter) - The East Berlin lawyer who became famous for engineering Cold War spy swaps, Wolfgang Vogel, was convicted by a German court Tuesday of extorting money from East German emigrants eager to flee to the West.

<p>

Vogel, a close confidant of former East German leader Erich Honecker and one of the Soviet bloc's rare millionaires, was found guilty of perjury, four counts of blackmail and five counts of falsifying documents.

<p>

The Berlin court gave him the two-year suspended sentence and a $63,500 fine. Prosecutors had pressed for a jail sentence of 3 3/8 years and a $215,000 penalty.

<p>

Vogel, 70, who got his start arranging the 1962 exchange of U.S. pilot Gary Powers for Soviet spy Rudolf Abel, insisted his only crime was trying to help unite people separated by the Cold

War division of Germany.

<p>

‘‘The court said that I helped people --- what more can I say?’’ Vogel said after Judge Heinz Holzinger spent 90 minutes reading the verdict to a packed courtroom.

<p>

‘‘An extortionist cannot help people,’’ Vogel said, trying to look upbeat despite the conviction. ‘‘If I had to do it over again, I would do exactly the same thing.’’

<p>

A flashy dresser whose golden Mercedes was a regular sight at Cold War spy swaps, he also spent six years negotiating the 1986 release of Soviet dissident Anatoly Shcharansky (now Israeli Natan Sharansky) in exchange for captured communist spies.

<p>

Wolfgang Ziegler, his chief defense attorney, criticized the ruling and said Vogel would decide within the next week whether to appeal.

<p>

Tuesday’s ruling was a rare victory for prosecutors trying to use unified Germany’s courts to bring former communist power-brokers to justice.

<p>

Vogel’s numerous friends in the west, mostly Bonn officials who worked with him during the Cold War, attacked the verdict as an example of ‘‘victor’s justice.’’

<p>

‘‘Vogel was certainly no Florence Nightingale of the Cold War and we should not make a hero out of him,’’ said Klaus Boelling, Bonn’s representative in East Berlin during the 1980s. ‘‘But he had this important function, without which the Cold War in Germany would have been even colder.’’

<p>

Before the collapse of Erich Honecker’s iron-fisted regime, Vogel served as his personal aide for ‘‘humanitarian issues’’ and was widely admired on both sides of the Iron Curtain as a master deal-maker.

<p>

But prosecutors charged Vogel with abusing his position to force East German emigrants to sell their houses and property at bargain prices to the Communist elite.

<p>

The attorney is best known for his 30-year career as an unofficial East-West go-between, securing the release of more than 100 spies and agents.

<p>

Yet Vogel’s delicate negotiating also helped shepherd nearly

34,000 political prisoners and 215,000 members of German families to freedom in the West.

<p>

Vogel, a suave attorney with a passion for clocks and Mercedes cars, delivered an estimated $2.3 billion in hard currency to the impoverished East from Bonn in exchange for the release of the political prisoners.

<p>

He also managed to accumulate a small fortune for himself, earning hefty fees in both West German and East German marks, triggering charges that he was a greedy opportunist.

<p>

Throughout the 14-month trial which ended last month, Vogel maintained that he was an ‘‘honest broker’’ who assisted people seeking freedom or reunifcation with their families, using means that reflected the moral ambiguities of the time.

REUTER

</TEXT>

<TRAILER>

Reut12:09 01-09-96

</TRAILER>

</DOC>

<DOC>

B.4 Story Number 04 (BERLIN/960110.0288)

<DOCID> reute960110.0288 </DOCID>

<STORYID cat=i pri=r> a1708 </STORYID>

<FORMAT> &D3; &D1; </FORMAT>

<KEYWORD> BC-GERMANY-VOGEL </KEYWORD>

<HEADER> reute 01-10 0215 </HEADER>

<SLUG> BC-GERMANY-VOGEL </SLUG>

<HEADLINE>

Convicted East German spy-swap lawyer to appeal

</HEADLINE>

<TEXT>

<p>

BERLIN, Germany (Reuter) - A lawyer for Wolfgang Vogel, the East Berlin lawyer who gained fame by engineering Cold War spy swaps, announced Wednesday his client would appeal against a conviction for perjury and blackmail.

<p>

A Berlin court convicted the go-between Tuesday of extorting property from East Germans trying to get permission from the Communist government to emigrate to the West.

<p>

But Vogel's lawyer Wolfgang Ziegler said the trial had not proved that Vogel blackmailed his clients into giving up their houses and land in return for permission to depart.

<p>

Ziegler said he would also argue to the Federal Court of Justice that the statute of limitations had passed on several of the cases.

<p>

Vogel, a close confidant of former East German leader Erich Honecker and one of the Soviet bloc's rare millionares, was found guilty of perjury, four counts of blackmail and five counts of falsifying documents.

<p>

The Berlin court gave him a two-year suspended sentence and a $63,500 fine. Prosecutors had pressed for a 3 3/8-year jail sentence and a $215,000 penalty.

<p>

Vogel, 70, who arranged the 1962 exchange of U.S. pilot Gary Powers for Soviet spy Rudolf Abel, insisted his only crime was trying to help unite people separated by the Cold War division of Germany.

</TEXT>

<TRAILER>

Reut16:26 01-10-96

</TRAILER>

</DOC>

<DOC>

B.5 Story Number 14 (BERLIN/960117.0297)

<DOCID> reute960117.0297 </DOCID>

<STORYID cat=i pri=r> a1744 </STORYID>

<FORMAT> &D3; &D1; </FORMAT>

<KEYWORD> BC-GERMANY-VOGEL </KEYWORD>

<HEADER> reute 01-17 0237 </HEADER>

<SLUG> BC-GERMANY-VOGEL </SLUG>

<HEADLINE>

Stiffer sentence sought for German spy-swap lawyer

</HEADLINE>

<TEXT>

<p>

BERLIN, Germany (Reuter) - Prosecutors announced Wednesday they were appealing against a suspended sentence for perjury and blackmail given last week to Wolfgang Vogel, the East Berlin lawyer who gained fame by engineering Cold War spy swaps.

<p>

Berlin justice ministry spokeswoman Uta Foelster said prosecutors would seek to have the Federal Court of Justice declare the sentence too lenient.

<p>

A Berlin court convicted Vogel last week of extorting property from East Germans trying to get permission from the Communist government to emigrate to the West.

<p>

Vogel, 70, a close confidant of former East German leader Erich Honecker and one of the Soviet bloc's rare millionares, was found guilty of perjury, four counts of blackmail and five counts of falsifying documents.

<p>

The Berlin court gave him a two-year suspended sentence and a $63,500 fine. Prosecutors had pressed for a 3 3/8-year jail sentence and a $215,000 penalty.

<p>

Vogel is also appealing against the conviction.

<p>

His lawyers say the statute of limitations had passed on several of the counts, and that it was not proved that he blackmailed clients into giving up houses and land in return for permission to depart.

<p>

Vogel, 70, who got his start arranging the 1962 exchange of U.S. pilot Gary Powers for Soviet spy Rudolf Abel, has insisted his only crime was trying to help unite people separated by the Cold War division of Germany.

</TEXT>

<TRAILER>

Reut15:58 01-17-96

</TRAILER>

</DOC>

<DOC>

Appendix C

Tools and resources used

C.1 Introduction

This dissertation would not have been possible were there not tons of public domain programs, program languages, utilities, and corpora that we used to conceptualize, design, and implement SUMMONS. This appendix presents a brief overview of the tools and programs widely used in the implementation of SUMMONS.

C.2 FUF

Brief Description: FUF is a functional unification-based programming language mainly used for text generation.

Author/Site: Michael Elhadad (Columbia University)

Where mentioned in thesis: knowledge representation (Chapter 4) and text generation (Chapter 6).

C.3 SURGE

Brief Description: SURGE is a large-scale reusable functional grammar of English.

Author/Site: Michael Elhadad and Jacques Robin (Columbia University).

Where mentioned in thesis: text generation (Chapters 6 and 5).

C.4 CREP

Brief Description: a lex-based information extraction utility.

Author/Site: Darrin Duford and Jacques Robin (Columbia University)

Where mentioned in thesis: To extract summarization phrases (Chapter 3) and to build entity and description lists (Chapter 7).

C.5 PARTS

Brief Description: a stochastic part of speech tagger.

Author/Site: Ken Church (AT&T Research)

Where mentioned in thesis: To facilitate the extraction of entity and description names (Chapter 7).

C.6 WORDNET

Brief Description: WORDNET [Miller *et al.*, 1990] is an on-line hierarchical lexical database which contains semantic information about English words.

Author/Site: George Miller (Princeton University) and many others.

Where mentioned in thesis: To filter entity names and to categorize descriptions (Chapter 7).

C.7 POACHER

Brief Description: a Web-based recursive document search utility (robot).

Author/Site: Neil Bowers (Canon RCE)

Where mentioned in thesis: To collect the articles used in LRR 7, clustering, and description extraction (Chapter 7).

C.8 RIPPER

Brief Description: software for rule induction for classification from examples.

Author/Site: William Cohen (AT&T)

Where mentioned in thesis: To learn contextual constraints on the choice of descriptions of entities (Chapter 8).

C.9 CRYSTAL

Brief Description: a trainable information extraction system.

Author/Site: Stephen Soderland, Wendy Lehnert, David Fisher (University of Massashusetts) and others.

Where mentioned in thesis: Chapter 10.

Appendix D

The LOT library

D.1 Introduction

The LOT library deals with lists of FD templates (called "lots"). We decided to include it as an appendix for two reasons: to give a general feeling of the code written for SUMMONS and to promote its reuse in other generation systems.

D.2 Code

```
(defvar *operators* (make-hash-table)
  "Hash table operator-name to operator / description.")


(defvar *ordered-operators* nil
  "The list of names of operators in the order they have
   been defined.")
```

```
(defmacro def-operator (name description input)
  `(multiple-value-bind (test found) (gethash ',name *operators*)
     (declare (ignore operator))
     (if found
       (format t "Redefining operator ~s~%" ',name)
       (push ',name *ordered-operators*))
     (setf (gethash ',name *operators*)
               ',(cons description input))))

(defun get-operator (operator)
  (cdr (gethash operator *operators*)))

(defun get-condition (operator)
  (top-gdp operator {condition}))

(defun get-action (operator)
  (top-gdp operator {action}))

(defun get-type (operator)
  (top-gdp operator {type}))

(defun clear-operators ()
  "Clear operators"
  (setf *ordered-operators* nil)
```

```
  (clrhash *operators*))

(defun add-numbers (inputl)
  (reset-gi)
  (mapcar #'(lambda (y) (next-gi) (add-numbers-helper y))
    (get-test inputl)))

(defun add-numbers-not (lot)
  (reset-gi)
  (mapcar #'(lambda (y) (next-gi) (add-numbers-helper y)) lot))

(defun which-no (fd)
  (top-gdp fd {admin msg_no}))

(defun add-numbers-helper (fd)
  (insert-fd (list (list 'msg_no *i*)) fd {admin}))

(defun get-pair (pair fd)
  (list
      (nth (- (first pair) 1) fd)
      (nth (- (second pair) 1) fd)))

(defun get-items (lin fd)
  (mapcar #'(lambda (y) (nth (- y 1) fd)) lin))
```

```
(defun dpowerset (s)
  (if (null s) (list nil)
    (append (dpowerset-helper (car s) (dpowerset (cdr s)))
            (dpowerset (cdr s)))))

(defun dpowerset-helper (e s)
  (if (null s) nil
    (cons (cons e (car s)) (dpowerset-helper e (cdr s)))))

(defun powerset-n (lot n)
  (mapcan #'(lambda (y) (and (eq (length y) n) (list y)))
     (dpowerset lot)))

(defun powerset-msgno (lot n)
  (mapcar #'(lambda (x)
      (mapcar #'which-no x)) (powerset-n lot n)))

(defun lot-msgno (lot)
  (mapcar #'(lambda (x)
      (mapcar #'which-no x)) lot))

(defun powerset-msgno-reverse (pow lot)
  (mapcar #'(lambda (y)
```

```
    (mapcar #'(lambda (x) (nth (- x 1) lot)) y)) pow))

(defun get-value (lot n path)
  (top-gdp (nth (- n 1) lot) path))

(defun true-p (pred a b)
  (eval (list pred a b)))

(defun compare-paths (pred lot a pa b pb)
  (true-p pred (get-value lot a pa) (get-value lot b pb)))

(defun check-pair (pred lot pair pa pb)
  (compare-paths pred lot (first pair) pa (second pair) pb))

(defun match-power-p (pred lot pa pb)
  (mapcar #'(lambda (y) (check-pair pred lot y pa pb))
     (all-pairs lot)))

(defun match-power (pred lot pa pb)
  (mapcan #'(lambda (y)
      (and (check-pair pred lot y pa pb) (list y)))
      (all-pairs lot)))

(defun all-pairs (lot)
```

```
  (lot-msgno (powerset-n lot 2)))

(defun match-operator (operator inputl)
  (let* ((op (get-operator operator))
         (condition (get-condition op))
         (pred (first condition))
         (pa (second condition))
         (pb (third condition))
         (lop (add-numbers-not inputl)))
    (apply 'match-power (list pred lop pa pb))))

(defun modify-fd-if (fd total path n)
  (if (= n (which-no total))
      (insert-fd fd total path)
    total))

(defun modify-lot (fd path n lot)
  (mapcar #'(lambda (y) (modify-fd-if fd y path n)) lot))

(defun apply-minimal-operator (operator lop0 pair)
  (let* ((op (get-operator operator))
         (action (get-action op))
         (where (first action))
         (path (second action))
```

```
        (val (third action))
        (lop (add-numbers-not lop0))
        (n (cond ((string-equal "left" where) (first pair))
                 ((string-equal "right" where) (second pair))
                 (t 0))))
    (apply 'modify-lot (list val path n lop))))

(defun apply-operator (operator lop0 pair)
  (let* ((op (get-operator operator))
        (action (get-action op))
        (val (third action)))
    (apply-minimal-operator operator lop0 pair)))

(defun match-op (operator inputl pair)
  (let* ((op (get-operator operator))
        (condition (get-condition op))
        (pred (first condition))
        (pa (second condition))
        (pb (third condition)))
    (check-pair pred inputl pair pa pb)))

(defun run-op (operator inputl pair)
  (if (match-op operator inputl pair)
      (apply-operator operator inputl pair)
```

```
    inputl))

(defun run-ops (ops-pairs l)
  (cond ((= (length ops-pairs) 0) (add-numbers l))
        (T (apply #'run-op (list (first (first ops-pairs))
             (run-ops (rest ops-pairs) l)
             (second (first ops-pairs)))))))
```

Appendix E

Sample project - building an encyclopedia from the Web

E.1 Introduction

This appendix describes a moderately difficult project related to some aspects of the dissertations. The amount of work required is more or less equivalent to a semester project in a Natural Language Processing course.

The goal is to build a lexical resource ("Who's who" or Encyclopedia) from Web-accessible text and use it in text generation.

Each entry (about a person or concept) in the encyclopedia should be annotated with factual sentences or phrases extracted from on-line text. For example, the entry about a famous writer would ideally contain information about his or her life, work, and influence on others.

E.2 Automated creation of an encyclopedia

An encyclopedia can be viewed as a collection of cross-referenced documents, each of which describes an entity (person, place, organization) or concept. Encyclopedias that focus on people are called biographical encyclopedias (or dictionaries). Another type of encyclopedias (geographical encyclopedias) describe places (cities, countries, regions, etc.)

Traditionally, encyclopedias are manually built and require a tremendous amount of time and the effort of hundreds of specialists and technical writers. This task involves a lot of research through literature as well as field work.

A hypothesis that can be verified through a project like this one is that a large amount of encyclopedia-type information can be found in text documents accessible through the Web. A news story, for example, may contain a reference to a city and include a description of it which indicates where it is located. For example, the sentence *Bill Clinton and Boris Yeltsin met in Moscow, the capital of Russia* includes a fact about Moscow (being the capital of Russia) which could potentially be included in the encyclopedia entry on Moscow.

The goal of this project is to built a small-scale encyclopedia using the concepts of language reuse. This involves identifying a feasible set of entities (or encyclopedia entries) and potential sources of textual information about them. The output should consist of a (potentially hyperlinked) encyclopedia describing the entities using grammatically correct sentences or phrases extracted from the original text sources.

E.3 System components

At least three components are necessary: a Web robot to search the Web for appropriate documents, a parser or information extraction tool to extract reusable sentences, and a user interface through which the system can be queried. In SUMMONS , the corresponding components are POACHER , CREP , and the CGI interface to PROFILE.

E.4 Potential evaluation

Some parameters on which to evaluate the system include: entities retrieved (precision and recall), correctness of the factual information about them (again in terms of precision and recall), the ability to filter out outdated and duplicate information, and the grammaticality of the extracted text.

E.5 Possible extensions

So far, we have only mentioned the extraction of phrases or sentences related to a particular entity. It would be interesting to use the semantic and syntactic information in the sentences to understand the relations between them and the entity to which they are related. For example, certain sentences describe the age of a person and his major accomplishments or positions held, while others relate a sports team with the city in which it is based or a newspaper with its political affiliation.