Generating Natural Language Summaries from Multiple On-Line Sources

Ph.D. Thesis Proposal

Dragomir R. Radev
`radev@cs.columbia.edu`

Technical Report CUCS-005-97

Department of Computer Science
Columbia University
New York, NY 10027

March 28, 1997

Contents

List of Figures

List of Tables

Abstract

We present a methodology for summarization of news on current events. Our approach is included in a system, called SUMMONS which presents news summaries to the user in a natural language form along with appropriate background (historical) information from both textual (newswire) and structured (database) knowledge sources. The system presents novel approaches to several problems: summarization of multiple sources, summarization of multiple articles, symbolic summarization through text understanding and generation, asynchronous summarization and generation of textual updates.

We pay specific attention to the generation of summaries that include descriptions of entities such as people and places. We show how certain classes of lexical resources can be automatically extracted from on-line corpora and used in the generation of textual summaries. We describe our approach to solving the interoperability problem of the various components by wrapping all system modules with facilitators which effect the communication between the components using a standardized language. We present a plan for completion of the research as well as a set of metrics that can be used in measuring the performance of the system.

Chapter 1

Introduction

One of the major problems with the Internet is the abundance of information and the difficulty for the average computer user to read everything existing on a specific topic. There exist now more than 100 operational sources of live newswire on the Internet, mostly accessible through the World-Wide Web [Berners-Lee, 1992]. Some of the most popular sites include Reuters News [Reuters, 1996], CNN's Web site [CNN, 1996], ClariNet's e.News on-line newspaper [ClariNet, 1996], as well as the New York Times online edition [NYT, 1996].

For the typical user, it is nearly impossible to go through megabytes of news every day to select articles he wishes to read. Even in the case 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 search and selection services, as well as for summarization facilities made available to the user.

There currently exist more than 40 search and selection services on the World-Wide Web, such as DEC's AltaVista [AltaVista, 1996], Lycos [Lycos, 1996], and DejaNews [DejaNews, 1997], all of which allow keyword searches for recent news. However, only recently have there been practical results in the area of summarization.

One currently existing Web-based summarization system, NetSumm, developed by the Language Group at British Telecom Laboratories [Preston and Williams, 1994, Cuts, 1994, NetSumm, 1996], uses a statistical, language-independent approach to selecting relevant sentences from a news article. It has an impressive user interface, and is practically domain-independent, but doesn't address two major issues: it only summarizes articles that the user has selected and it only summarizes a single article at a time. Other statistical systems [Kupiec et al., 1995], [Rau et al., 1994], while using different algorithms for sentence extraction, have similar disadvantages as NetSumm.

Another major unsolved problem involves conveying rapidly changing information to the end user in a sensible format. This information can come from a multitude of different sources which use different internal representations to store it. A summarizing program needs to be able to retrieve all this information in real time, process it and produce meaningful summaries in natural language.

We present a system, called SUMMONS[1] [McKeown and Radev, 1995, Radev and McKeown, 1997b], which introduces novel techniques in the following areas:

- It *combines information* from *multiple* news articles into a coherent summary using symbolic

[1]SUMMarizing Online NewS articles

techniques.

- It *augments* the resulting summaries using descriptions of entities obtained from on-line sources.

- It features an agent-based architecture [Radev, 1996] that provides access to heterogeneous lexical and conceptual resources.

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

1.1 Summarization of multiple articles

With a few exceptions (see Chapter 2), all existing statistical text summarizers provide summaries of single articles by extracting sentences from them. If such systems were to summarize a series of articles, they would simply process each of them on its own and output the resulting summaries. Such summaries will likely contain a significant amount of repeated information, as do the source articles themselves.

In our approach, the summarizer works conceptually on a large set of articles, treating them as a single input for summarization. This way, the system can trace the development of an event over time and report its different stages without repetition. It can also trace the updates in the news reports as they occasionally contradict each other.

1.2 Summarization from multiple sources

Given the omnipresence of on-line news services, one can expect that any interesting news event will be covered by several, if not most of them. If different sources present the same information, the user clearly needs only have access to one of them. Practically, this assumption doesn't hold, as different sources provide updates from a different perspective and at different times. An intelligent summarizer's task, therefore, is to attain 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 will report the contradiction and attribute the contradictory information to its sources, rather than select one of the contradictory pieces without the other.

1.3 Symbolic summarization through text understanding and generation

An inherent problem to sentence-extraction based summarizers is the lack of discourse-level fluency in the output. The extracted sentences fit together only in the case they are adjacent in the source document. Because SUMMONS uses language generation techniques to determine the content and wording of the summary based on information extracted from input articles, it has all necessary information to produce a fluent surface summary.

1.4 Automatic acquisition of lexical resources for generation

We show how the summary generated using symbolic techniques can be enhanced so that it includes descriptions of entities that participate in it. If a user tunes in to news on a given event several days after the first occurrence, references to and descriptions of the event, people and organizations involved may not be adequate. We collect such descriptions from on-line sources of past news, representing them using our generation formalism, and reuse them in later generation of summaries.

1.5 Interoperability

A large-scale summarization system will be able to monitor multiple sources of news as well as other types of input data (such as domain ontologies). Since all of these are owned and maintained by different organizations, it makes sense for the summarization system to be as modular and extensible as possible. This means that it should be able to handle the addition or disappearance of specific components without losing performance. An interesting research topic that we address is that of interoperability - more specifically, how to integrate smaller subsystems developed by different people and not all of which are directly related to summarization. We show how a knowledge interchange language can be used to coordinate the multiple participants in the system. This approach also allows for the separate components to be written in different programming languages and operate on different platforms.

1.6 Asynchronous summarization

Synchronous (demand-based) summarization requires that the user need to know when a new article relevant to his interests has appeared and *feed it* to the summarizer in order to get a summary back. Such an approach doesn't lead to any economy of time for the user, since he still has to spend time checking whether new articles have been posted and then send them to the summarizer.

It would be more efficient for the user to be notified automatically when a new article has been published [Radev, 1994], and even better, to be sent a summary of the article directly. Such asynchronous summaries can be based on the specific interests of the user, contained in his user profile or the pending subscription. These summaries can also be tailored to the user's prior knowledge of the subject or event. For example, if the announcement of a specific event has already been sent to the user, he need only get *updates* in the future. In our work on SUMMONS, we are adding facilities so that the summarizer can keep track of what information has already been presented to the user and avoid its repetition in future summaries.

1.7 Structure of the proposal

The next chapter positions our research in the context of prior work in the area. Chapter 3 describes the system architecture we have developed for the summarization task. The next two chapters describe in more detail how a base summary is generated from multiple source articles and how the base summary is enhanced using descriptions extracted from on-line sources. Chapter 6 describes the current status of our system as well as some directions for future work in symbolic summarization of heterogeneous sources. We also present a schedule for completion of the work for this thesis and a propose a methodology for evaluation of the different components of SUMMONS. We conclude this proposal in Chapter 7 where we also summarize the contributions of the research work.

Chapter 2

Related Work

Previous work related to this dissertation can be divided into five groups: text summarization, construction of knowledge sources for text generation, summarization of non-textual sources, extraction of lexical elements from raw text, and architectures and protocols for building intelligent information processing systems. This chapter will provide some background on the research performed in these five areas and will conclude with a brief overview of prior research on summarization at Columbia.

2.1 Text Summarization

All prior work on text summarization has been essentially done using statistical sentence extraction techniques. Paice [Paice, 1990] has given an overview of the different existing methods for summarization using sentences extracted from the original text. The original idea was presented in the 60es [Luhn, 1958] and has been used extensively over the years, especially since the Internet came into wide use. Existing algorithms and systems range from domain-dependent [DeJong, 1979, Tait, 1983, Jacobs and Rau, 1990] techniques which make use of specialized knowledge to topic-independent such as NetSumm [Preston and Williams, 1994], or Xerox's summarizer [Kupiec et al., 1995].

Xerox's system uses a training algorithm by means of which corpus analysis is used to determine the weights of different features used in sentence extraction, such as keyword frequencies, use of fixed phrases, thematic words, and lower-level features such as paragraph position, sentence length, and initial capitalization of words.

Summaries generated through the extraction of sentences typically don't address two major issues: fluency (with the notable exception of the work done by Paice who performs post-processing of the summary to resolve certain cases of anaphora), and performance only slightly better or even lower than the baseline (extracting the lead, or initial sentences only). Rau *et al.* [Rau et al., 1994] report that statistical summaries of individual news articles were rated lower by evaluators than simply using the lead sentence or two from the article. In more recent work, Kupiec's system is known for performing better than the baseline approach. Summaries that consist of sentences plucked from texts have been shown to be useful indicators of content, but they are also highly unreadable, often judged to readable at between 5% and 27% [Brandow et al., 1990].

The sentence-extraction approach has the problem that it uses sentences that were not intended to be used alone as a replacement for the text. Note, in any case, that these approaches cannot handle the task that we address, summarization of *multiple documents*, since this requires information about similarities and differences across documents.

In ongoing work at Carnegie Mellon University, Carbonell (1996, personal communication) is developing statistical techniques to identify similar sentences and phrases across articles and unlike other statistical approaches, this shows promise for summarization across 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.

Recently, text summarizers have been announced by both Microsoft and Apple Computer. The author hasn't been able to review these systems in time for this proposal.

2.2 Building Knowledge Sources for Generation

The construction of a database of phrases for reuse in generation is quite novel. Previous work on extraction of collocations for use in generation [Smadja and McKeown, 1991] is related to ours in that full phrases are extracted and syntactically typed so that they can be merged with individual words in a generation lexicon to produce a full sentence. However, extracted collocations were used only to determine *realization* of an input concept. In our work, stored phrases are intended to be used to provide content that can identify a person or place for a reader, in addition to providing the actual phrasing.

2.3 Summarization of Non-Textual Data

There exists a significant body of previous work related to summarization of non-textual data. Among the most well-known systems, ANA [Kukich, 1983], SEMTEX [Rösner, 1987], FOG [Bourbeau et al., 1990], and LFS [Iordanskaja et al., 1994] need to be mentioned. All of them are domain-specific and their domains range from weather forecasts (FOG) to stock-market reports (ANA). Another system, ZEDDOC, which summarizes Web access logs is described in 2.6.

2.4 Description Extraction

As our work also involves generation using extracted descriptions, we also provide a comparison with work on proper noun extraction, extraction of people descriptions in various information extraction systems developed for the message understanding conferences [MUC, 1992], and use of extracted information for question answering.

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, 1991] 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. Like 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 [Miller et al., 1990] to provide semantic filtering.

Our work on extracting descriptions is quite similar to the work carried out under the DARPA message understanding program for extracting descriptions. The purpose for and the scenario in which description extraction is done is quite different, but the techniques are very similar. It is based on the paradigm of representing patterns that express the kinds of descriptions we expect; unlike previous work we do not encode semantic categories in the patterns since we want to capture all descriptions regardless of domain. We represent the descriptions in a format that facilitates symbolic generation.

Another system, called MURAX [Kupiec, 1993], is similar to ours from a different perspective. MURAX also 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 string of words in a text that is most likely to serve as an answer to a user's wh-query. In our work, the string that is extracted may be merged, or regenerated, as part of a larger textual summary.

2.5 Architectures for Intelligent Information Processing Systems

Some ideas in the overall architectural design of our system are derived from the concepts set forth by Genesereth and Ketchpel [Genesereth and Ketchpel, 1994] and Etzioni and Weld [Etzioni and Weld, 1994] and the work on the standardization of protocols and languages for agent communication such as KQML and KIF [Finin et al., 1994, Genesereth and Fikes, 1992]. The current proposed KQML specification is described in [Labrou, 1996].

KQML (Knowledge Query and Manipulation Language) is a language and protocol for exchanging information and knowledge and is part of the ARPA Knowledge Sharing Effort [Patil et al., 1992]. KQML defines an extensible set of performatives which specify the actions that intelligent agents can perform or attempt to perform on themselves and on each other's knowledge bases. KQML has been used in concurrent engineering, intelligent planning, and scheduling.

2.6 Previous Work on Summary Generation at Columbia

Summarization has been also one of the main research areas in the natural language group at Columbia. Independently or in conjunction with Bellcore, the following systems have been developed over the course of the last few years.

- STREAK [Robin and McKeown, 1993, Robin, 1994, Robin and McKeown, 1995] generates summaries of basketball games using the concept of revisions.
- PLANDOC [McKeown et al., 1994b, McKeown et al., 1995, Shaw, 1995] generates summaries of the activities of telephone planning engineers. It uses of conjunction, ellipsis and paraphrase to result in concise, yet fluent reports.
- ZEDDOC [Passonneau et al., 1997] generates Web traffic summaries for advertisement management software.

All of these systems take data as input. The focus for these systems has been on linguistic summarization. SUMMONS, on the other hand, focuses on conceptual summarization.

Chapter 3

System overview

The overall architecture of our summarization system (Figure 3.1) draws on research in software agents [Genesereth and Ketchpel, 1994] to allow connections to a variety of different types of data sources. Facilities are used to provide a transparent interface to heterogeneous data sources which run on various machines and may be written in different programming languages. Currently, we have developed (but not fully integrated) facilities to connect to various live news streams, the CIA World Factbook, and past newspaper archives. The architecture allows for the incorporation of additional facilitators and data sources as our work progresses.

Data is extracted from these different sources and then combined into a conceptual representation of the summary. The summarization component, shown on the left side of the figure, consists of a *base summary generator*, which combines information from multiple input articles and organizes that information using a paragraph planner. The structured conceptual representation of the summary is passed to the content planner, shown at the bottom of the diagram. The content planner also receives input from the CIA World Factbook and possible descriptions of people or organizations to augment the base summary. The full content will be sent to a sentence generator, implemented using the FUF/SURGE language generation system [Elhadad, 1993, Robin, 1994].

The right side of the figure shows how proper nouns and their descriptions are extracted from past news. An entity extractor identifies proper nouns in the past newswire archives, along with descriptions. Descriptions are then categorized using the WordNet hierarchy, to identify the function of each description. Finally, an FD for the description is generated in order that it can be re-used in fluent ways in the final summary. The final (enhanced) summary contains the text from the base summary along with the text from the descriptions.

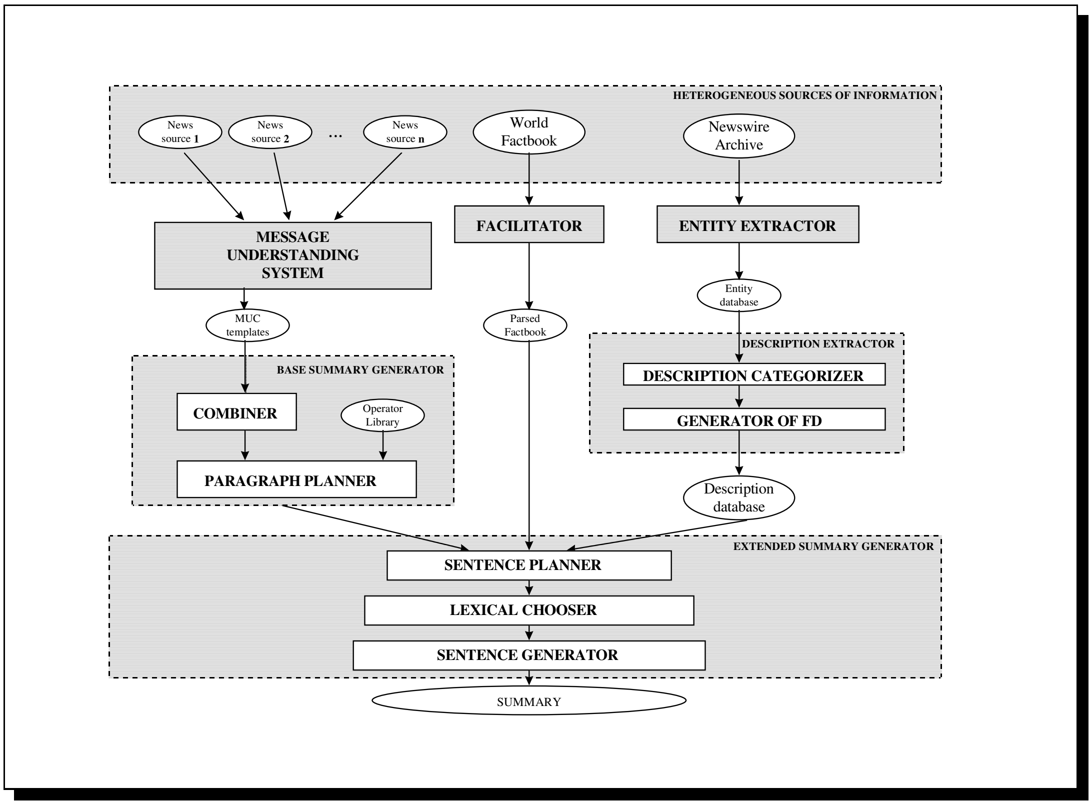

Figure 3.1: SUMMONS Architecture.

Chapter 4

Generating the Base Summary

The base summary is produced by SUMMONS in an attempt to generate fluent text from sets of templates that contain the salient facts reported in the input texts. To produce these templates, we rely upon the DARPA message understanding systems. These systems accept full text as input, extracting specific pieces of information from a given newspaper article. An example of a template produced by MUC systems and used in our system is shown in Figure 4.1. To test our system, we used the templates produced by systems participating in MUC-4 [MUC, 1992], available from the Linguistic Data Consortium (LDC), as input. MUC-4 systems operate on the terrorist domain and extract information by filling fields such as perpetrator, victim, and type of event, for a total number of 25 fields. In addition, we filled the same template forms by hand from current news articles for further testing[1]. Currently, work is under way in our group on the building of an information extraction module similar to the ones used in the MUC conferences which we will later use as an input to SUMMONS. The resulting system will not only be able to generate summaries from pre-parsed templates but will also produce summaries directly from raw text.

Our work provides a methodology for developing summarization systems, identifies planning operators for combining information in a concise summary, and uses empirically collected phrases to mark summarized material. We have collected a corpus of newswire summaries that we used as data for both developing the planning operators and gathering a large set of lexical constructions used in summarization. This corpus will eventually aid in a full system evaluation. Since news articles often summarize previous reports of the same event, we collected a corpus of articles which included short summaries of previous articles.

We used this corpus to develop both the content planner (i.e., the module which determines what information to include in the summary) and the linguistic component (i.e., the module which determines the words and surface syntactic form of the summary) of our system. We used the corpus to identify operators which are used to combine information; this includes techniques for linking information together in a related way (e.g., identifying changes, similarities, trends) as well as making generalizations. We also identified phrases that are used to mark summaries and used these to build the system lexicon. An example summary produced by the system is shown in Figure 4.3. This paragraph summarizes four articles about two separate terrorist acts that took place in Israel in March of 1996 using two different operators.

[1] Answer templates or system output from later MUC and TIPSTER conferences were not available to us.

```
MESSAGE: ID                      TST3-MUC4-0010
MESSAGE: TEMPLATE                2
INCIDENT: DATE                   01 NOV 89
INCIDENT: LOCATION               EL SALVADOR
INCIDENT: TYPE                   ATTACK
INCIDENT: STAGE OF EXECUTION     ACCOMPLISHED
INCIDENT: INSTRUMENT ID          -
INCIDENT: INSTRUMENT TYPE        -
PERP: INCIDENT CATEGORY          TERRORIST ACT
PERP: INDIVIDUAL ID              "TERRORIST"
PERP: ORGANIZATION ID            "THE FMLN"
PERP: ORG. CONFIDENCE            REPORTED: "THE FMLN"
PHYS TGT: ID                     -
PHYS TGT: TYPE                   -
PHYS TGT: NUMBER                 -
PHYS TGT: FOREIGN NATION         -
PHYS TGT: EFFECT OF INCIDENT     -
PHYS TGT: TOTAL NUMBER           -
HUM TGT: NAME                    -
HUM TGT: DESCRIPTION             "1 CIVILIAN"
HUM TGT: TYPE                    CIVILIAN: "1 CIVILIAN"
HUM TGT: NUMBER                  1:  "1 CIVILIAN"
HUM TGT: FOREIGN NATION          -
HUM TGT: EFFECT OF INCIDENT      DEATH: "1 CIVILIAN"
HUM TGT: TOTAL NUMBER            -
```

Figure 4.1: Sample MUC-4 Template.

```
(message
  (system (id "TST3-MUC4-0010")
          (template-no 2))
  (source (secondary "NCCOSC"))
  (incident (date "01 NOV 89")
            (location "El Salvador")
            (type attack)
            (stage accomplished))
  (perpetrator (category terr-act)
               (org-id "THE FMLN")
               (org-conf rep-fact))
  (victim (description civilian)
          (number 1)
)
```

Figure 4.2: Parsed MUC-4 Template.

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 in the second incident. *Later the same day*, Reuters reported that *the radical Muslim group* Hamas has claimed responsibility for the act.

Figure 4.3: Sample output from SUMMONS.

While the system we report on is fully implemented, our work is undergoing continuous development. Currently, the system includes 7 different planning operators, a testbed of over 100 input templates, and can produce fully lexicalized summaries for approximately half of the cases. Our work provides a methodology for increasing the vocabulary size and the robustness of the system using a collected corpus, and moreover, it shows how summarization can be used to evaluate the message understanding systems, identifying future research directions that would not be pursued under the current MUC evaluation cycle[2]. Due to inherent difficulties in the summarization task, our work is a substantial first step and provides the framework for a number of different research directions.

The rest of this chapter describes the summarizer, specifying the planning operators used for summarization as well as a detailed discussion of the summarization algorithm showing how summaries of different length are generated. We provide examples of the summarization markers we collected for the lexicon and show the demands that summarization creates for interpretation.

4.1 Overview of the Summarization Component

The summarization component of SUMMONS is based on the traditional language generation system architecture [McKeown, 1985, 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 perform its task. The lexicon contains the vocabulary for the system and encodes constraints about when each word can be used. As shown in Figure 3.1, SUMMONS' content planner 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 (Functional Unification Formalism) [Elhadad, 1991, Elhadad, 1993] sentence generator, which uses a large systemic grammar of English, called SURGE[3] to fill in syntactic constraints, build a syntactic tree, choose closed class words, and eventually linearize the tree as a sentence.

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 restricted the domain to articles on terrorism, since MUC systems including the MUC system that we obtained from NYU

[2]Participating systems in the DARPA message understanding program are evaluated on a regular basis. Participants are given a set of training text to tune their systems over a period of time and their systems are tested on unseen text at follow-up conferences.

[3]FUF is a sentence generator that follows the functional unification paradigm, whereas SURGE is a large-scale surface generation grammar of English built on top of FUF.

[Grishman et al., 1992] were also restricted to such topics only. However, we constructed by hand a set of 100 templates that include also recent terrorist events such as the World Trade Center bombing, the Hebron Mosque massacre and more recent incidents in Israel, and the disaster in Oklahoma City, which were not handled by the original message understanding systems. We also created by hand a set of templates unrelated to real newswire messages which we used for testing some techniques of our system. We enriched the templates for all these cases by adding four slots: the primary source, the secondary source and the times at which both sources made their reports[4]. We found having the source of the report immensely useful for summarization, because there are often conflicts between different reports of an event and these can indicate the level of confidence in the report, particularly as reports change over time. For example, if many sources all report the same incidents for a single event, 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.

Researchers in our department are currently working on an event tracking language [Aho et al., 1997]. It uses pattern-matching techniques to track changes to on-line news sources and provide a live feed of articles that relate to a changing event to the summarizer. The articles may be written at any point in time and may be written by the same or many sources.

The 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 extended summary. 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. From our interactions with the Columbia School of Journalism, we have learned that human writers usually lay out articles in the form of "inverted pyramids", i.e., the most information appearing in the lead sentence(s). Consequently, SUMMONS rates information in terms of importance, where information that appears in only one article is given a lower rating and information that is synthesized from multiple articles is rated more highly. When space allows, SUMMONS may choose to include the base facts from two separate articles as well as the conclusion that can be drawn from both, while given less space, only the summarizing fact would be included.

Development of SUMMONS was made easier because of the language generation tools and framework available at Columbia University. No changes in the FUF sentence generator were needed. In addition, the lexical chooser and content planner were based on the design used in the PLANDoc automated documentation system, developed jointly with Bellcore to summarize the activities of telephone planning engineers [McKeown et al., 1994a]. In particular, we used FUF to implement the lexical chooser, representing the lexicon as a grammar as we have done in many previous systems (e.g., Elhadad [Elhadad, 1993], Robin [Robin, 1994], McKeown *et al.* [McKeown et al., 1993], Feiner and McKeown [Feiner and McKeown, 1991]), and thus the main effort was in identifying the words and phrases needed for the domain. The content planner features several stages, as does the PLANDoc system. It first groups messages together, identifies commonalities between them, and notes how the discourse influences wording by setting realization flags. Before lexical choice, SUMMONS maps the templates into the FD [Elhadad, 1993] formalism expected as input to FUF and uses a domain ontology (derived from the ontologies represented in the message understanding systems) to enrich the input.

[4]primary source - usually a direct witness of the event, and secondary source - most often a press agency or journalist, reporting the event.

The main point of departure for SUMMONS is in the stage of identifying what information to include and how to group it together, as well as the use of a corpus to guide this and later processes. In PLANDoc, successive messages are very similar and the problem is to form a grouping that puts the most similar messages together, allowing the use of conjunction and ellipsis to delete repetitive material. For summarizing multiple news articles, the task is almost the opposite; we need to find the differences from one article to the next, identifying how the news has changed. Thus, the main problem was the identification of summarization strategies, which indicate how information is linked together to form a concise and cohesive summary. As we have 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 summary was critical to identifying the different summaries possible.

4.2 Methodology: collecting and using a summary corpus

In order to produce plausible and understandable summaries, we used available on-line corpora as models, including the Wall Street Journal and current newswire from Reuters and the Associated Press. Our corpora contain about 2 MB of news articles. We have manually grouped articles in threads related to single events or series of similar events.

From the corpora collected in that way, we extracted manually, and after careful investigation, several hundred language constructions which we found relevant to the types of summaries that we want to produce. In addition to the summary cue 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 essentially scarce in the newswire corpora and we needed to collect them from other sources (e.g., modifying templates that we acquired from the summary corpora to provide a wider coverage).

Since one of our goals has been conciseness, we have tried to assemble small paragraph summaries which in essence describe a single event and its change over time, or a series of related events with no more than a few sentences.

4.3 Summary operators for content planning

We have developed a set of heuristics derived from the corpora which decide what types of simple sentences constitute a summary, in what order they need to be listed, as well as the ways in which simple sentences are combined into more complex ones. In addition, we have specified which summarization-specific phrases are to be included in different types of summaries.

We attempt to identify a preeminent set of templates from the input to the system. This set needs to contain a large number of similar fields. If this holds, we can merge the set into a simpler structure, keeping the common features and marking the distinct features as Elhadad [Elhadad, 1993] and McKeown [McKeown et al., 1994a] suggest.

At each step, a summary operator is selected based on existing similarities between messages 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 from 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 of its input and output. Figure 4.4 shows part of the rules for the Contradiction operator. Given two templates, if INCIDENT.LOCATION is the same, the time of first report is before time of second report, the report sources are different, and at least one other slot differs in value (this rule not shown), apply the contradiction operator to combine the templates.

```
((#TEMPLATES == 2) &&
 (T[1].INCIDENT.LOCATION == T[2].INCIDENT.LOCATION) &&
 (T[1].INCIDENT.TIME < T[2].INCIDENT.TIME) && ...
 (T[1].SECSOURCE.SOURCE != T[2].SECSOURCE.SOURCE)) ==>
(apply (``contradiction'',``with-new-account'', T[1], T[2]))
```

Figure 4.4: Rules for the Contradiction operator.

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 thus, summaries sometimes must identify differences between reports. A description of the operators we identified in our corpus follows, accompanied by an example of system output for each operator. Each example primarily summarizes two input templates, as this is the result from applying a single operator once. More complex summaries can be produced by applying multiple operators on the same input, as shown in the introductory example.

4.3.1 Change of perspective

When an initial report gets a fact wrong or has incomplete information, the change is usually included in a summary. In order for this operator to apply, the source field must be the same, while the value of another field changes so that it is not compatible with the original value. For example, if the number of victims changes, we know that the first report was *wrong* 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.

> 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 *at least 12 people* were killed and *105* wounded.

4.3.2 Contradiction

When two sources report conflicting information about the same event, 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. The number of sources that contradict each other can indicate the level of confusion about the event. Note that the current output of the message understanding systems does not include sources. However, SUMMONS could use this feature to report disagreement between output by different systems. A summary might indicate that one of the MUC systems determined that 20 people were killed, while the other MUC system determined only 5 were killed. The difference between this example and the previous one on Change of Perspective is the source of the update. If the same source announces a change, then we know that it has realized a change in the facts. Otherwise, an additional source presents information which is not necessarily more correct than the information presented by the earlier source.

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

4.3.3 Addition

When a subsequent report indicates that additional facts became known, this is reported in a summary. 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 (up for numbers).

> 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 has claimed responsibility for the act.

4.3.4 Refinement

In subsequent reports a more general fact may be refined. Thus, if the location is originally reported to be New York City, it might later be noted as a particular borough of New York. Or, if a terrorist group is identified as Palestinian, later the exact name of the terrorist group may be determined. Since the update is assigned a higher value of "importance", it will be favored over the original message in a shorter summary.

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

4.3.5 Agreement

If two sources agree on the facts, this will heighten the reader's confidence in their veracity and thus, agreement between sources is usually reported.

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

4.3.6 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 generalize multiple events as shown in the 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.*

4.3.7 Trend

There is a trend if two or more messages 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.

4.3.8 No information

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 of that country don't give out any information.

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

4.4 Algorithm

The algorithm used in the system to sort, combine, and generalize the input messages can be described as follows:

4.4.1 Input

At this stage, the system receives a set of templates from the Message Understanding Conferences or a similar set of messages from a related domain. All templates are described as lists of attribute/value pairs (see Figure 4.6). These pairs (with the exception of the source information) are defined in the MUC-4 guidelines.

4.4.2 Preprocessing

This stage includes the following substages:

- The templates are sorted in chronological order.
- Messages that have obviously been incorrectly generated by a MUC system are identified and filtered out by hand.
- A database of all fields and messages is created. This database is used later as a basis for grouping and collapsing messages.
- All irrelevant fields or fields containing bad values are manually marked as such and don't participate in further analyses.

- Knowledge of the source of the information is marked as the specific Message Understanding System for the site submitting the template if it is not present in the input template. Note that since the current Message Understanding Systems do not extract the source, this is the most specific we can be for such cases.

4.4.3 Heuristic combination

The template database is scanned for interesting relationships between templates. Such patterns trigger reordering of the templates and modification of their individual "importance" values. As an example, if two templates are combined with the "Refinement" operator, the "importance" value of the combined message will be greater than the sum of the individual "importance" of the constituent messages. At the same time, the values of these 2 messages are lowered (still keeping a higher value on the later one, which is supposed to be the more correct of the two). All templates directly extracted from the MUC output are assigned an initial importance value of 100. Currently, with each application of an operator, we lower the value of a contributing individual template by 20 points and give any newly produced template that combines information from already existing contributing templates a value greater than the sum of the values of the contributing templates after those values have been updated. Furthermore, some operators reduce the importance values of existing templates even further (e.g., the refinement operator reduces the importance of chronologically earlier templates by additional increments of 20 points because they contain outdated information). These values were set empirically and future work will incorporate a more formal approach. Thus, the final summary is likely to contain only the combined message if there are restrictions on length. It can also contain all three of them if length restrictions are considerably lax. The value of the "importance" of the message corresponds also to the position in the summary paragraph, as more important messages will be generated first.

Each new template contains information indicating whether its constituent templates are obsolete and thus no longer needed. Also, at this stage the coverage vector (a data structure which keeps track of which templates have been already combined and which ones are still to be considered in applying operators) is updated to point to the messages which are still active and can be further combined. This way we make sure that all messages still have a chance of participating in the actual summary.

The resulting messages are combined into small "paragraphs" according to the event or series of events that they describe. Each paragraph can then be realized by the linguistic component. Each set of templates produces a single paragraph.

4.4.4 Discourse planning

Given the relative importance of the messages included in the database after the Heuristic Combination stage, the content planner is called to organize the presentation of information within a paragraph. It looks at consecutive messages in the database, marked as separate paragraphs from the previous stage, and assigns values to "realization switches" [McKeown et al., 1994a] which control local choices such as tense and voice. They also govern the presence or lack of certain constituents to avoid repetition of constituents and to satisfy anaphora constraints.

4.4.5 Format conversion

All messages included in the database and augmented through the content planner are converted to FUF Functional descriptions (FDs).

4.4.6 Ordering of templates and linguistic generation

In order to produce the final text, SUMMONS carries out the following steps:

- Templates are sorted according to the order of the value of the "importance" slot. Only the top templates are realized. Messages with higher importance values appear with priority in the summary if a restriction on length is specified.
- An intermediate module, the ontologizer, converts factual information from the message database into data structures compatible with the ontology of the MUC domain. This is used, for example, to make generalizations (e.g., that Medellin and Bogota are in Colombia).
- The lexical chooser component of SUMMONS is a functional (systemic) grammar which emphasizes the use of summarization phrases originating from the summary corpora.
- Surface generation from the augmented message FDs is performed using SURGE and FUF. We have written additional generation code to handle paragraph-level constructions (the summarization operators).

4.5 An example of system operation

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 4.5.

The four news articles result in four different 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.

Let's now consider the four templates in the order that they appear in the list of templates. These templates are shown in Figures 4.6 – 4.9. They are generated manually from the input newswire texts. Information about the primary and secondary sources of information is added SECSOURCE) The differences in the two templates (which will trigger certain operators) are shown in **bold face**. The summary generated by the system was shown earlier in Figure 4.3 and is repeated here in Figure 4.10.

The first two sentences are generated from templates one and two. 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 5). Typically, a description is included in the source text and should be extracted by the message understanding system. In the cases in which a description doesn't appear or is not extracted,

Message 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 the Islamic fundamentalist group 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.

Message 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.

Message 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.

Message 4: TEL AVIV (Reuter) - 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 4.5: Messages 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 4.6: Template for newswire message 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 4.7: Template for newswire message 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 4.8: Template for newswire message 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	

Figure 4.9: Template for newswire message 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 has claimed responsibility for the act.

Figure 4.10: SUMMONS output based on the four messages.

SUMMONS generates a description from the database of extracted descriptions. We are currently working on an algorithm which will select the best description based on such parameters as the user model (what information has already been presented to the user?), the attitude towards the entity (is it favorable?), or a historical model which describes the changes in the profile of a person over the period of time (what was the previous occupation of the person who is being described?).

4.6 Asynchronous notification

The user has the option of selecting that new summaries be generated whenever new information on a specific topic has arrived. For example, if the user has expressed an interest in getting summaries about El Salvador, the subscription will be handled via KQML messages. Currently, only a small KQML-compatible prototype has been implemented.

A subscription request (Figure 4.11) is sent from the planner to the MUC facilitator. Whenever a new message, such as the one shown in Figure 4.2 becomes available, the MUC facilitator will reply with an appropriate message (Figure 4.12).

We have already implemented the base KQML-base asynchronous subscription facilitator. We still need, however, to add content to the protocols used.

```
(subscribe
  :content
    (EQ
       (message
         (incident
           (location "El Salvador"))))
  :ontology geog-onto
  :language KQML
  :reply-with "loc-salvador-1"
  :sender "planner"
  :receiver "muc1"
)
```

Figure 4.11: KQML subscription message.

```
(reply
  :content (message
           (system (id "TST3-MUC4-0010")
             ...)
           )
  :ontology geog-onto
  :language KQML
  :in-reply-to "loc-salvador-1"
  :sender "muc1"
  :receiver "planner"
)
```

Figure 4.12: KQML reply message.

Chapter 5

Including Information from Non-Textual Sources

5.1 Introduction

The main source of information that will be included in the summary is the source article (or source articles). Most summarization systems use the message being summarized as the single source of information. A major part of the research for this dissertation, however, is to develop methods for information that might not have been included in the source articles to appear in the generated summaries.

Such information typically comes from additional newswire (e.g., older articles) or from on-line structured databases. We have currently developed a method for extracting one class of such information and its integration into the summaries generated.

5.2 Case Study: Generating Descriptions

When a summary refers to an 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 other sources [Radev and McKeown, 1997a].

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 of 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. A summarizer that has access to different descriptions will be able to select the description that best suits both the reader and the series of articles being summarized. Similarly, in the example in Chapter 4, if the user hasn't been informed about what Hamas is and either a description to Hamas is not found in the source text or is not extracted by the message understanding system, or if the summary generator decides to use a different description from the one used in the source articles, then older descriptions in FD format can be retrieved from the description database and be sent to SURGE for surface generation.

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 module 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 sources thus allowing for a more complete record 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 can combine descriptions from articles appearing only a few minutes before the ones being summarized with descriptions from past news in a permanent record for future use.

Since the profile manager constructs a lexicalized, syntactic functional description (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 flexibly produce summary wording.

We have extended the profile manager by semantically categorizing descriptions using WordNet, so that a generator can more easily determine which description is relevant in different contexts.

The profile manager can also be used in a real-time fashion to monitor entities and the changes of descriptions associated with them over the course of time.

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

5.3 Creation of a Database of Profiles

In this section, we describe the description management module of SUMMONS shown in Figure 3.1. We explain how entity names and descriptions for them are extracted from old newswire and how these descriptions are converted to FDs for symbolic generation.

5.3.1 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 of 1995. The purpose of such an initial set of descriptions is twofold. First, it allows us to test the other components of the system. Furthermore, at the time a description is needed it limits the amount of online full text, Web search that must be done. 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 search of additional text.

- **Extraction of candidates for proper nouns**. After tagging the corpus using the POS part-of-speech tagger [Church, 1988], we used a CREP [Duford, 1993] regular grammar to first extract all possible candidates for entities. These consist of all sequences of words that were tagged as proper nouns (NP) by POS. 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 POS. Table 5.1 shows how many entities we retrieve at this stage, and of them, how many pass the semantic filtering test.

- **Weeding out of 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 description extraction stage. All 421 entity names retrieved by the system are indeed proper nouns.

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

Table 5.1: Two-word and three-word sequences retrieved by the system.

5.3.2 Extraction of descriptions

There are two occasions on 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 5.3.1) 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 either a Web user or the generation system.

There exist many live sources of newswire on the Internet that can be used for this second case. Some that merit our attention are the ones that can be accessed remotely through small client programs that don't require any sophisticated protocols to access the newswire articles. Such sources include HTTP-accessible sites such as the Reuters site at www.yahoo.com and CNN Interactive at www.cnn.com, as well as others such as ClariNet which is propagated through the NNTP protocol. All these sources share a common characteristic in that they are all updated in real time and all contain information about current events. Hence, they are therefore likely to satisfy the criteria of pertinence to our task, such as the likelihood of the sudden appearance of new entities that couldn't possibly have been included a priori in the generation lexicon.

Our system generates finite-state representations of the entities that need to be described. An example of a finite-state description of the entity "Yasser Arafat" is shown in Figure 5.1. These full expressions are used as input to the description extraction module which uses them to find candidate sentences in the corpus for finding 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.

```
SEARCH_STRING   =   (({NOUN_PHRASE}{SPACE})+{SEARCH_0})|
                    ({SEARCH_0}{SPACE}{COMMA}{SPACE}{NOUN_PHRASE})
SEARCH_109      =   [Yy]asser{T_NOUN}{SPACE}[Aa]rafat{T_NOUN}
SEARCH_0        =   {SEARCH_1}|{SEARCH_2}|...|{SEARCH_109}|...
```

Figure 5.1: Finite-state representation of "Yasser Arafat" in the search pattern.

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 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 a simple noun (e.g., "president Bill Clinton") to a much longer expression (e.g., "Gilberto Rodriguez Orejuela, the head of the Cali cocaine cartel"). Other forms of descriptions, such as relative clauses, are the focus of ongoing implementation.

Table 5.2 shows some of the different patterns retrieved.

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 5.2: Examples of retrieved descriptions.

5.3.3 Categorization of descriptions

We use WordNet to group extracted descriptions into categories. For all words in the description, we try to find a WordNet hypernym that can restrict the semantics of the description. Currently, we identify concepts such as "profession", "nationality", and "organization". Each of these concepts is triggered by one or more words (which we call "trigger terms") in the description. Table 5.2 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.

5.3.4 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"). 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.

Figure 5.2 shows the profile associated with the key "John Major".

```
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 5.2: Profile for John Major.

The database of profiles is updated every time a query retrieves new descriptions matching a certain key.

5.4 Generation of Descriptions

The content planner of a language generation system that needs to present an entity to the user that he has not seen previously, might want to include some background information about it. However, in case the extracted information doesn't contain an appropriate description, the system can use some descriptions retrieved by the profile manager.

5.4.1 Transformation of descriptions into Functional Descriptions

Since our major goal in extracting descriptions from on-line corpora is to use them in generation, we have written a utility which 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 5.3. The corresponding FD is shown in Figure 5.4.

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

Figure 5.3: Retrieved description for Silvio Berlusconi.

5.4.2 Lexicon creation

We have identified several major advantages of using FDs produced by the system in generation compared to using canned phrases. We are investigating ways of using all of these ideas to improve the summaries generated by the system.

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

```
((cat np)
 (complex apposition)
 (restrictive no)
 (distinct ~(((cat common)
             (possessor ((cat common)
                         (determiner none)
                         (lex ``Italy'')))
             (classifier ((cat noun-compound)
                          (classifier ((lex ``former'')))
                          (head ((lex ``prime'')))))
             (head ((lex ``minister''))))
            ((cat person-name)
             (first-name ((lex ``Silvio'')))
             (last-name ((lex ``Berlusconi''))))))))
```

Figure 5.4: Generated FD for Silvio Berlusconi.

- **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.
- **Enhancement of descriptions**. If we have retrieved "prime minister" as a description for Silvio Berlusconi, and later we obtain knowledge 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.
- **Merging lexicons**. The lexicon generated automatically by the system can be merged with a domain lexicon generated manually.

These advantages look very promising and we will be exploring them in detail in our work on summarization in the near future.

Chapter 6

System Status, Planned Work, and Proposed Evaluation

6.1 System status

Currently, our system can handle simple summaries consisting of up to five sentences which are limited to the MUC domain and to a few additional types of events for which we have manually created MUC-like templates. Seven planning operators have been implemented and tested on a set of over 100 input templates.

The description generator has the following coverage:

- **Syntactic coverage**. Currently, the system includes an extensive finite-state grammar that can handle various pre-modifiers and appositions. The grammar matches arbitrary noun phrases in each of these two cases to the extent that the POS part-of-speech tagger provides a correct tagging.
- **Precision**. In Section 5.3.1 we showed 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 5.3.1. Of the 611 descriptions, 551 (90.2%) were correct. The others included a roughly equal number of cases of incorrect NP attachment and incorrect part-of-speech assignment.
- **Length of descriptions**. The longest description retrieved by the system was 9 lexical items long: "Maurizio Gucci, the former head of Italy's Gucci fashion dynasty". The shortest descriptions are 1 lexical item in length - e.g. "President Bill Clinton".
- **Protocol coverage**. We have implemented retrieval facilities to extract descriptions using the NNTP (Usenet News) and HTTP (World-Wide Web) protocols. These modules can be easily reused in other systems with similar architecture to ours.

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.

At this stage, the systems doesn't incorporate the descriptions in the summary nor does it have a robust algorithm for selecting which descriptions should be used. As a result, the generation modules are not yet integrated with the description extraction/selection modules.

Several components related to interoperability are also implemented in entirety (e.g. the subscription package in KQML and the query-response interface to the MUC and World Factbook facilitators). We have converted all ontologies related to the MUC and the Factbook into a format accessible to all agents involved.

6.2 Planned work and proposed evaluation

6.2.1 Generating updates from live news

An important goal of our research is the generation of evolving summaries that continuously update the user on a given topic of interest. Hence, the system will have a model containing all prior interaction with the user. We will investigate the following problem: given that k articles have already been summarized and presented to the user, what information from a new, $k+1st$ article should be extracted for the summary? We will develop a discourse model based on empirical study of summary corpora. Such a model will contain a representation of the content and wording of summaries that have already been presented to the user.

When generating an update, the summarizer must avoid repeating content and at the same time, must be able to generate references to entities and events that were described in previously generated summaries. We will investigate the lexical structure of historical information as used in summaries written by humans and compare it with the structure used to express new information.

6.2.2 Integrating textual and structured information

We will develop a set of planning operators for combination of information from different types of sources. Such operators will be based on empirical analysis of summary corpora and will be similar in structure to the operators that are already implemented.

After we collect a series of descriptions for each possible entity, we need to decide how to select among all of them. There are two scenarios. In the first one, we have to pick one single description from the database which best fits the summary that we are generating. In the second scenario, the evolving summary, we have to generate a *sequence* of descriptions, which might possibly view the entity from different perspectives. We will be investigating algorithms that will decide the order of generation of the different descriptions. Among the factors that will influence the selection and ordering of descriptions, we can note the user's interests, his knowledge of the entity, the focus of the summary (e.g., "democratic presidential candidate" for Bill Clinton, vs. "U.S. president"). We can also select one description over another based on how recent they have been included in the database, whether or not one of them has been used in a summary already, whether the summary is an update to an earlier summary, and whether another description from the same category has been used already. We have yet to decide under what circumstances a description needs to be generated at all.

We are interested in implementing existing algorithms or designing our own that will match different instances of the same entity appearing in different syntactic forms - e.g., to establish that "PLO" is an alias for the "Palestine Liberation Organization". We will investigate using co-occurrence information to match acronyms to full organization names and alternative spellings of the same name with each other.

As a final twist to the integration of non-textual sources, we will apply some ongoing work on image classification [Aho et al., 1997, Smith and Chang, 1996] to the problem of generating *illustrated summaries*. We will define a set of features that can be used to retrieve and classify images from the Internet in order to be used as an appropriate illustration of a summary.

6.2.3 Robustness/lexicon

One of the main goals of the remaining work is to increase the overall robustness of SUMMONS. We plan to work primarily in the following areas: automatic retrieval and analysis of summarization phrases from corpora, increase of the coverage of the understanding and planning modules (both at the sentence and discourse levels), and the paraphrasing power of the summary generation components.

A more detailed list of planned work is included at the end of this section.

6.2.4 Proposed evaluation

We will perform a formal evaluation of SUMMONS using the following criteria:

- Develop metrics for evaluation of the success of the system in preserving the important information in the source. A good summary will allow the user to avoid reading the full set of source documents. We will set up a task evaluation, where the user must monitor a set of online sources to find a set of proscribed information. We will monitor length of time to complete the task as well as number of summaries and source documents accessed or read. We will compare task completion time with users who do not receive the automatically generated summaries. In addition to task completion time, we will also use follow-up questionnaires to ask users to rate summary quality in terms of adequacy of content and phrasing.

- Develop metrics for evaluation of the robustness of the system. Also, evaluate the ease of adding a new service, knowledge source, or medium.

- Develop a coverage benchmark for different message types and operators.

- Evaluate generated summaries against real summaries from the newswire as well as summaries written by humans (as in [Robin and McKeown, 1995]). Such a corpus would allow, for example, scoring of recall and precision of the content of generated summaries against content of the human written summaries. It would also allow us to measure coverage of the generated phrasing against the corpus. We will investigate carrying out the evaluation by asking subjects to write summaries for us. In the less likely event we are able to locate a corpus of summaries serving the function of briefing, we will be able to carry out a more extensive evaluation of this type.

- Evaluate portability of the system (architecture, protocols, grammars, and knowledge sources) to other domains, e.g., elections, natural disasters, stock reports, and sport score summaries.

6.3 Schedule for completion

Work on this dissertation (including writing of the thesis) should be completed by June of 1998. Appendix A shows the schedule for completion of different milestones. The rest of this section describes in some detail the list of more specific tasks to be completed before the dissertation defense. Depending on how much time and effort each of the individual tasks takes, certain items on the list might not be completed by the time of defense. Items that are particularly significant to the overall success of the dissertation will be handled with priority. Such items are marked with asterisks in the list below.

6.3.1 Corpus Analysis

- (*) Develop a methodology for extracting summarization phrases from real summaries. We will use our existing collection of corpora which contain both long news articles and summaries. We will investigate the possibility of automatically inserting certain classes of cue phrases into the summary.
- Determine lexical structure of historical information. Collect empirical evidence of different linguistic structures used to present new information as compared to historical information.
- (*) Investigate algorithms for automatic grouping articles into threads related to the same event. Possibly, make use of such algorithms being developed by other research groups in the department.

6.3.2 Lexicon and Language Generation

- Assess the feasibility of using a large-scale generation lexicon under development by other members of our group [Jing et al., 1997].
- (*) Enhance the sentence-level generation grammar. The goal would be to cover 50% of all possible values for each of the different slots of the MUC templates.
- (*) Enhance the paragraph-level generation grammar. The goal is to generate paragraphs of arbitrary length within one and five sentences.
- (*) Increase the paraphrasing power of the generation system. After performing an empirical analysis of news corpora, establish a list of frequently used paraphrases and include them in the summary generator.
- Determine the message space of all messages that can be generated.

6.3.3 Summarization Operators

- (*) Develop a taxonomy of robust summarization operators. Ideally, such operators should cover all possible patterns of templates in a set of up to 4 templates.
- (*) Design planning operators for combining data from both messages and non-textual data. These operators should be implemented in a way similar to the one used for the existing operators.
- Investigate a formal model of summarization.

6.3.4 Message Ontology

- (*) Add fields to the templates to express changes in point of view, reliability of information. Use these fields to guide summarization.
- Find and extend an existing ontology of event types.
- Integrate domain ontologies from other sources. Such ontologies are available from the Message Understanding Conferences.

6.3.5 Integration of Sources

- (*) Investigate the use of revisions to add background information. Robin [Robin, 1994] has successfully used such revisions in other domains. Our goal is to figure out how feasible revisions are in the domain of news summarization.
- (*) Develop methods for establishing what information should not be included in the summary.
- Develop a model of contradiction among sources.
- Investigate the possibility of doing all summary planning in FUF.

6.3.6 Message Understanding

- (*) Acquire and improve existing message understanding systems.
- (*) Evaluate small-scale message understanding systems already developed at Columbia.
- Rewrite template preprocessing module.

6.3.7 Agents and KQML

- (*) Develop the subscription module. Create a model of news subscription based on real-time information sources available on the World-Wide Web.
- (*) Evaluate alternatives to using FUF as the content language in the KQML model.
- (*) Develop a set of KQML performatives (advertise, tell, broker, ask) for various modules of the system. Try to incorporate the system into the framework of other existing KQML-based information systems.

6.3.8 User Interface

- (*) Build a demoable Web-based user interface to the system. The interface should demonstrate key components of the system, such as the inclusion of non-textual data or the generation of updates.
- (*) Evaluate the possibility of generating a summary as a response to a user query.
- Facilitate access to full articles that were used in building the summary.

6.3.9 Interoperability

- (*) Determine the feasibility of using agent-based and KQML-based technologies in Natural Language Processing and Information Retrieval.
- Investigate the possibility of adding a new layer on top of KQML that will allow sharing resources among multiple systems based on a similar architecture.
- Develop a scenario of interaction among agent components of the system. Test scenario with human users of the system.

6.3.10 Possible extensions beyond the scope of the thesis work

- Investigate the possibility of generating in one language text written in another language.
- (*) Integrate this research with other ongoing research related to Digital Libraries. More specifically, apply techniques developed for this dissertation in automated illustration of generated text.
- Investigate approaches to developing a temporal model of relevance of information.
- Evaluate existing algorithms for message classification and use them for grouping articles that will be summarized together.
- Investigate the possibility of automatic generation of templates for various domains.
- Position our work among related work in summarization.

Chapter 7

Conclusion and Contributions

Our prototype system, SUMMONS, demonstrates the feasibility of generating summaries of a series of news articles on the same event, highlighting changes over time. The ability to automatically provide summaries of heterogeneous material will critically increase 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 individual 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 summaries. Our framework allows for experimentation with summaries of different length and for the combination of multiple, independent summary operators to produce enhanced summaries that contain information from structured databases.

7.1 Contributions

We have described a framework for constructing summaries from on-line heterogeneous data using novel techniques in statistical information extraction, intelligent agent protocols, natural language generation, and information access over the Internet. We have shown how a system can automatically summarize multiple articles that come from multiple sources using symbolic methods. We have proposed a methodology that makes use of text understanding and generation instead of statistical sentence selection. An important contribution is the automatic retrieval of certain classes of lexical information and their use in text generation. Finally, we have shown how the different components of a multi-agent information processing system can be built of small, interoperable components and be extended to cover additional sources and modes of information.

Chapter 8

Acknowledgments

This work was partially supported by NSF grant GER-90-2406 and by a grant from Columbia University's Strategic Initiative Fund sponsored by the Provost's Office.

The author is grateful to the following people for their invaluable comments during the writing of the proposal and at presentations of work related to its content: Kathy McKeown (thesis adviser), Alfred Aho, Shih-Fu Chang, Vasileios Hatzivassiloglou, Alejandro Jaimes, Judith Klavans, Julian Kupiec, David McDonald, John Smith, Kazi Zaman.

Appendix A

Appendix: Schedule for Completion

Year	Month	Implementation	Evaluation	Writing/Defense
1997	March - April			finish proposal **propose**
	May - June	develop a model for generation of updates	develop evaluation metrics based on corpus analysis	
	July - August	develop modules for non-textual sources of information		
	September - October	build demoable system with Web interface which indicates		submit paper to "Information Processing and Management" or "ACM Transactions on Information Systems"
	November - December	complete work on planning operators for both textual and non-textual data		Submit paper to SIGIR'98
1998	January - February	complete work on agents, interoperability, and KQML	perform evaluation	write thesis
	March - April	finish implementation		write thesis
	May - June			finish writing **defend**

Table A.1: Schedule for completion.

Bibliography

[Aberdeen et al., 1992] Aberdeen, J., Burger, J., Connolly, D., Roberts, S., and Vilain, M. (1992). 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.

[Aho et al., 1997] Aho, A., Chang, S.-F., McKeown, K., Radev, D., Smith, J., and Zaman, K. (1997). Columbia digital news system : An environment for briefing and search over multimedia information. In *Proceedings of ADL*, Washington, DC.

[AltaVista, 1996] AltaVista (1996). Altavista. WWW site, URL: http:// altavista.digital.com.

[Ayuso et al., 1992] Ayuso, D., Boisen, S., Fox, H., Gish, H., Ingria, R., and Weischedel, R. (1992). 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.

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

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

[Brandow et al., 1990] Brandow, R., Mitze, K., and Rau, L. F. (1990). Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 26:135–170.

[Church, 1988] Church, K. W. (1988). 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. Association for Computational Linguistics.

[ClariNet, 1996] ClariNet (1996). ClariNet. WWW site, URL: http:// www.clari.net.

[CNN, 1996] CNN (1996). CNN interactive. WWW site, URL: http:// www.cnn.com.

[Coates-Stephens, 1991] Coates-Stephens, S. (1991). 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.

[Cowie et al., 1992] Cowie, J., Guthrie, L., Wilks, Y., Pustejovsky, J., and Waterman, S. (1992). 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.

[Cuts, 1994] Cuts, S. (1994). Science and technology section. *Economist*, 17:85–86.

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

[DejaNews, 1997] DejaNews (1997). Dejanews. WWW site, URL: http:// www.dejanews.com.

[DeJong, 1979] DeJong, G. (1979). *Skimming stories in real time: an experiment in integrated understanding.* PhD thesis, Computer Science Department, Yale University.

[Duford, 1993] Duford, D. (1993). Crep: a regular expression-matching textual corpus tool. Technical Report CUCS-005-93, Columbia University.

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

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

[Etzioni and Weld, 1994] Etzioni, O. and Weld, D. (1994). A softbot-based interface to the internet. *Communications of the ACM*, 37(7):72–76.

[Feiner and McKeown, 1991] Feiner, S. and McKeown, K. (1991). Automating the generation of coordinated multimedia explanations. *IEEE Computer*, 24(10):33–41.

[Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML - a language and protocol for knowledge and information exchange. Technical Report CS-94-02, Computer Science Department, University of Maryland and Valley Forge Engineering Center, Unisys Corporation.

[Genesereth and Fikes, 1992] Genesereth, M. and Fikes, R. (1992). Knowledge interchange format, version 3.0 : Reference manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, California.

[Genesereth and Ketchpel, 1994] Genesereth, M. and Ketchpel, S. (1994). Software agents. *Communications of the ACM*, 37(7):48–53.

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

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

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

[Jacobs and Rau, 1990] Jacobs, P. and Rau, L. (1990). SCISOR: Extracting information from on-line news. *Communications of the ACM*, 33(11):88–97.

[Jing et al., 1997] Jing, H., Hatzivassiloglou, V., Passonneau, R., and McKeown, K. (1997). Investigating complementary methods for verb sense pruning. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington, DC.

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

[Kupiec, 1993] Kupiec, J. M. (1993). 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.*

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

[Labrou, 1996] Labrou, Y. (1996). *Semantics for an Agent Communication Language.* PhD thesis, University of Maryland, Baltimore County.

[Lehnert et al., 1993] Lehnert, W., McCarthy, J., Soderlan, S., Riloff, E., Cardie, C., Peterson, J., and Feng, F. (1993). 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.

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

[Lycos, 1996] Lycos (1996). Lycos, Inc. home page. WWW site, URL: http:// www.lycos.com.

[Mani et al., 1993] Mani, I., Macmillan, R. T., Luperfoy, S., Lusher, E., and Laskowski, S. (1993). Identifying unknown proper names in newswire text. In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 44–54, Columbus, Ohio. Special Interest Group on the Lexicon of the Association for Computational Linguistics.

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

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

[McKeown et al., 1994a] McKeown, K., Kukich, K., and Shaw, J. (1994a). Practical issues in automatic documentation generation. In *Proceedings of the ACL Applied Natural Language Conference*, Stuttgart, Germany.

[McKeown et al., 1993] McKeown, K., Robin, J., and Tanenblatt, M. (1993). Tailoring lexical choice to the user's vocabulary in multimedia explanation generation. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Oh.

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

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

[McKeown and Radev, 1995] McKeown, K. R. and Radev, D. R. (1995). 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.

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

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, **3**(4):235–312.

[MUC, 1992] MUC, M. U. C. (1992). *Proceedings of the Fourth Message Understanding Conference (MUC-4).* DARPA Software and Intelligent Systems Technology Office.

[NetSumm, 1996] NetSumm (1996). Netsumm home page. WWW site, URL: http:// www.labs.bt.com/ innovate/informat/netsumm/index.htm.

[NYT, 1996] NYT (1996). New York Times. WWW site, URL: http:// www.nytimes.com.

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

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

[Passonneau et al., 1997] Passonneau, R., Kukich, K., McKeown, K., Radev, D., and Jing, H. (1997). Summarizing web traffic: A portability exercise. Technical Report CUCS-009-97, Columbia University, Department of Computer Science, New York, NY, USA.

[Patil et al., 1992] Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T., and Neches, R. (1992). The DARPA knowledge sharing effort: Progress report. In Rich, C., Swartout, W., and Nebel, B., editors, *Knowledge Representation*, pages 777–788.

[Preston and Williams, 1994] Preston, K. and Williams, S. (1994). Managing the information overload. *Physics in Business.*

[Radev, 1994] Radev, D. R. (1994). Rendezvous: A WWW synchronization system. Poster Session, Second International WWW Conference.

[Radev, 1996] Radev, D. R. (1996). 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.

[Radev and McKeown, 1997a] Radev, D. R. and McKeown, K. R. (1997a). Building a generation knowledge source using internet-accessible newswire. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington, DC.

[Radev and McKeown, 1997b] Radev, D. R. and McKeown, K. R. (1997b). Generating natural language summaries from multiple on-line sources. *submitted to Computational Linguistics.*

[Rau et al., 1994] Rau, L., Brandow, R., and Mitze, K. (1994). Domain-Independent summarization of news. In *Summarizing Text for Intelligent Communication*, pages 71–75, Dagstuhl, Germany.

[Reuters, 1996] Reuters (1996). Reuters news. WWW site, URL: http:// www.yahoo.com/headlines/.

[Robin, 1994] Robin, J. (1994). *Revision-Based Generation of Natural Language Summaries Providing Historical Background.* PhD thesis, Computer Science Department, Columbia University.

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

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

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

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

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

[Smith and Chang, 1996] Smith, J. R. and Chang, S.-F. (1996). Searching for images and videos on the world-wide web. submitted to ieee multimedia magazine, also cu/ctr technical report 459-96-25.

[Tait, 1983] Tait, J. (1983). *Automatic summarising of English texts.* PhD thesis, University of Cambridge, Cambridge, England.

[Weischedel et al., 1993] Weischedel, R., Ayuso, D., Boisen, S., Fox, H., Ingria, R., Matsukawa, T., Papageorgiou, C., MacLaughlin, D., Kitagawa, M., Sakai, T., Abe, J., Hosihi, H., Miyamoto, Y., and Miller, S. (1993). 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.