

Generating Natural Language Summaries from Multiple On-Line Sources

Dragomir R. Radev*
Columbia University

Kathleen R. McKeown†
Columbia University

We present a methodology for summarization of news about current events in the form of briefings that include appropriate background (historical) information. The system that we developed, SUMMONS, uses the output of systems developed for the DARPA Message Understanding Conferences to generate summaries of multiple documents on the same event, presenting similarities and differences, contradictions and generalizations among sources of information. We describe the various components of the system, showing how information from multiple articles is combined, organized into a paragraph, and finally, realized as English sentences. A feature of our work is the extraction of descriptions of entities such as people and places for reuse to enhance a briefing.

1. Introduction

One of the major problems with the Internet is the abundance of information and the resulting difficulty for a typical computer user to read all existing documents on a specific topic. Even within the domain of current news, the user's task is infeasible. There exist now more than 100 sources of live newswire on the Internet, mostly accessible through the World-Wide Web (Berners-Lee, 1992). Some of the most popular sites include news agencies and television stations like Reuters News (Reuters, 1996), CNN's Web site (CNN, 1996), and ClariNet's e.News on-line newspaper (ClariNet, 1996), as well as on-line versions of print media such as the New York Times on the Web 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 cases when the user can actually select all news relevant to the topic of his interest, he will still be faced with the problem of selecting a small subset that he can actually read in a limited time from the immense corpus of news available. Hence, there is a need for search and selection services, as well as for summarization facilities.

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.

Summaries can be used to determine if any of the retrieved articles are relevant (thereby allowing the user to avoid reading those that are not) or can be read in place of the articles to learn about information of interest to the user. Existing summarization systems (e.g., (Preston and Williams, 1994; Cuts, 1994; NetSumm, 1996; Kupiec,

* Department of Computer Science, 450 Computer Science Building, Columbia University, New York, NY 10027, USA. E-mail: radev@cs.columbia.edu

† Department of Computer Science, 450 Computer Science Building, Columbia University, New York, NY 10027, USA. E-mail: kathy@cs.columbia.edu

Pedersen, and Chen, 1995; Rau, Brandow, and Mitze, 1994)) typically use statistical techniques to extract relevant sentences from a news article. This domain-independent approach produces a summary of a single article at a time which can indicate to the user what the article is about. In contrast, our work focuses on generation of a summary that *briefs* the user on information in which he has indicated interest. Such briefings pull together information of interest from multiple sources, aggregating information to provide generalizations, similarities, and differences across articles, and changes in perspective across time. Briefings do not necessarily fully summarize the articles retrieved, but they update the user on information he has specified is of interest.

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

- Briefings are used to keep a person up to date on a certain event. Thus, they need to convey information about the event using appropriate historical references and the context of prior news.
- Briefings focus on certain types of information that are present in the source text in which the reader has expressed interest. They deliberately ignore facts that are tangential to the user's interests, whether or not these facts are the focus of the article. In other words, briefings are more user-centered than general summaries; the latter convey information that the writer has considered important, whereas briefings are based on information that the user is looking for.

We present a system, called SUMMONS¹ (McKeown and Radev, 1995; Radev, 1996; Radev and McKeown, 1997), shown in Figure 1, which introduces novel techniques in the following areas:

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

As can be expected from a knowledge-based summarization system, SUMMONS works in a restricted domain. We have chosen the domain of news on terrorism for several reasons. First, there is already a large body of related research projects in information extraction, knowledge representation and text planning in the domain of terrorism. For example, earlier systems developed under the DARPA Message Understanding Conference (MUC) were in the terrorist domain, and thus, we can build on these systems without having to start from scratch. The domain is important to a variety of users, including casual news readers, journalists, and security analysts. Finally, SUMMONS is being developed as part of a general environment for illustrated briefing over live multimedia information (Aho et al., 1997). Of all MUC system domains, terrorism is more likely to have a variety of related images than other domains that were explored such as mergers and acquisitions or management succession.

In order to extract information of interest to the user, SUMMONS makes use of components from several MUC systems. The output of such modules is in the form of

¹ SUMMarizing Online NewS articles

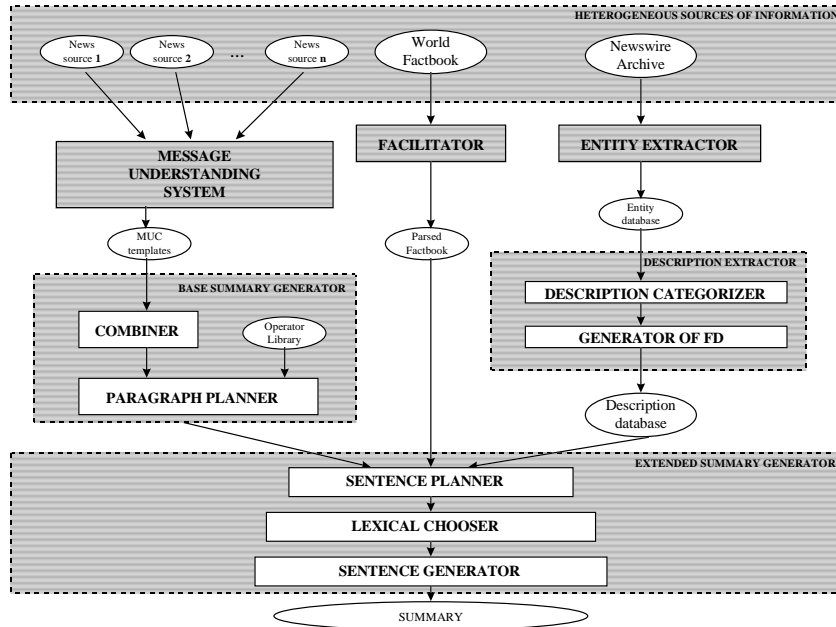


Figure 1
SUMMONS Architecture.

templates that represent certain pieces of information found in the source news articles, such as victims, perpetrators, or type of event. By relying on these systems, the task we have addressed to date is happily more restricted than direct summarization of full text. This has allowed us to focus on issues related to the combination of information in the templates and the generation of text to express them.

In order to port our system to other domains, we would need to develop new templates and the information extraction rules required for them. While this is a task we leave to those working in the information extraction field, we note that there do exist tools for semi-automatically acquiring such rules (Lehnert et al., 1993; Fisher et al., 1995). This helps to alleviate the otherwise knowledge-intensive nature of the task. We are working on the development of tools for domain-independent types of information extraction. For example, our work on extracting descriptions of individuals and organizations and representing them in a formalism that facilitates reuse of the descriptions in summaries can be used in any domain.

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

1.1 Summarization of multiple articles

With a few exceptions (Section 2), all existing summarizers provide summaries of single articles by extracting sentences from them. If such systems were applied to a series of articles, they might be able to extract sentences that have words in common with the other articles, but they would be unable to indicate how sentences that were extracted from different articles were similar. Moreover, they would certainly not be able to indicate significant differences between articles. In contrast, our work focuses on processing

of information from multiple sources to highlight agreements and contradictions as part of the summary.

1.2 Summarization from multiple sources

Given the omni-presence 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 only needs to 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 summarizers based on sentence extraction 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 (such as people, places, or organizations) it contains. If a user tunes in to news on a given event several days after the first report, 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 and represent them using our generation formalism for reuse in later generation of summaries.

1.5 Structure of the paper

The following section positions our research in the context of prior work in the area. Section 3 describes the system architecture that we have developed for the summarization task. The next two sections describe in more detail how a base summary is generated from multiple source articles and how the base summary is extended using descriptions extracted from on-line sources. Section 6 describes the current status of our system. We conclude this article in Sections 7 and 8 by describing some directions for future work in symbolic summarization of heterogeneous sources.

2. Related Work

Previous work related to summarization falls into three main categories. In the first, full text is accepted as input and some percentage of the text is produced as output. Typically, statistical approaches, augmented with key word or phrase matching, are used to lift full sentences from the article which can serve as a summary. Most of the work in this category produces a summary for a single article, although there are a few exceptions. The second two categories correspond to the two stages of processing that would have to be carried out if sentence extraction were not used: analysis of the input document to identify information that should appear in a summary and generation of a textual summary from a set of facts that are to be included. In this section, we first present work

on sentence extraction, next turn to work on identifying information in an article that should appear in a summary, and conclude with work on generation of summaries from data, showing how this task differs from the more general language generation task.

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

2.1 Summarization through Sentence Extraction

To allow summarization in arbitrary domains, researchers have traditionally applied statistical techniques (Luhn, 1958; Paice, 1990; Preston and Williams, 1994; Rau, Brandow, and Mitze, 1994). This approach can be better termed extraction rather than summarization, since it attempts to identify and extract key sentences from an article using statistical techniques that locate important phrases using various statistical measures. This has been successful in different domains (Preston and Williams, 1994) and is, in fact, the approach used in recent commercial summarizers (Apple (Boguraev and Kennedy, 1997), Microsoft, and inxight). Rau, Brandow, and Mitze (1994) report that statistical summaries of individual news articles were rated lower by evaluators than summaries formed by simply using the lead sentence or two from the article. This follows the principle of the “inverted pyramid” in news writing, which puts the most salient information in the beginning of the article and leaves elaborations for later paragraphs, allowing editors to cut from the end of the text without compromising the readability of the remaining text.

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

A more recent approach (Kupiec, Pedersen, and Chen, 1995) uses a corpus of articles with summaries to train a statistical summarization system. During training, the system uses abstracts of existing articles to identify the features of sentences that are typically included in abstracts. In order to avoid problems noted by Paice, Kupiec’s system produces an itemized list of sentences from the article thus eliminating the implication that these sentences function together coherently as a full paragraph. As with the other statistical approaches, Kupiec’s work is aimed at summarization of single articles.

Work presented at the 1997 ACL Workshop on Intelligent Scalable Text Summarization primarily focused on the use of sentence extraction. Alternatives to the use of frequency of key phrases included the identification and representation of lexical chains (Halliday and Hasan, 1976) to find the major themes of an article followed by the extraction of one or two sentences per chain (Barzilay and Elhadad, 1997), training over the position of summary sentences in the full article (Hovy and Lin, 1997), and the construc-

tion of a graph of important topics to identify paragraphs which should be extracted (Mitra, Singhal, and Buckley, 1997).

While most of the work in this category focuses on summarization of single articles, early work is beginning to emerge on summarization across multiple documents. In ongoing work at Carnegie Mellon, Carbonell (1996, personal communication) is developing statistical techniques to identify similar sentences and phrases across articles. The aim is to identify sentences that are representative of more than one article.

Mani and Bloedorn (1997) link similar words and phrases from a pair of articles using WordNet (Miller et al., 1990) semantic relations. They show extracted sentences from the two articles side by side in the output.

While useful in general, sentence extraction approaches cannot handle the task that we address, aggregate summarization across *multiple documents*, since this requires reasoning about similarities and differences across documents to produce generalizations, contradictions at a conceptual level.

2.2 Identifying Information in Input Articles

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

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

In addition to domain specific information extraction systems, there has also been a large body of work on identifying people and organizations in text through proper noun extraction. These are domain independent techniques that can also be used to extract information for a summary. Techniques for proper noun extraction include the use of regular grammars to delimit and identify proper nouns (Mani et al., 1993; Paik et al., 1994), the use of extensive name lists, place names, titles and "gazetteers" in conjunction with partial grammars in order to recognize proper nouns as unknown words in close proximity to known words (Cowie et al., 1992; Aberdeen et al., 1992), statistical training to learn, for example, Spanish names, from online corpora (Ayuso et al., 1992), and the use of concept-based pattern matchers that use semantic concepts as pattern categories as well as part-of-speech information (Weischedel et al., 1993; Lehnert et al., 1993). In addition, some researchers have explored the use of both local context surrounding the hypothesized proper nouns (McDonald, 1993; Coates-Stephens, 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. In a way similar to this research, our work also aims at extracting proper nouns without the aid of large word lists. We use a regular grammar encoding part-of-speech categories to extract certain text patterns (descriptions) and we use WordNet (Miller et al., 1990) to provide semantic filtering.

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. Ultimately, this approach could be used to extract information on items of interest in a user profile, where each question may represent a different point of interest. In our work, we also reuse strings (i.e., descriptions) as part of the summary, but the string that is extracted may be merged, or regenerated, as part of a larger textual summary.

2.3 Summary Generation

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

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

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

At least four previous systems developed elsewhere use natural language to summarize quantitative data, including ANA (Kukich, 1983), SEMTEX (Rösner, 1987), FOG (Bourbeau et al., 1990), and LFS (Iordanskaja et al., 1994). All of these use some forms of conceptual and linguistic summarization and the techniques can be adapted for our current work on summarization of multiple articles. In related work, Dalianis and Hovy (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.

3. System overview

The overall architecture of our summarization system (Figure 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 several machines and may be written in different programming languages. Currently, we have incorporated facilities 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.

The system extracts data from the different sources and then combines it 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 lexical chooser, shown at the bottom of the diagram. The lexical chooser also receives input from the World Factbook and possible descriptions of people or organizations to augment the base summary. The full content is then passed through a sentence generator, implemented using the FUF/SURGE language generation system (Elhadad, 1993; Robin, 1994). FUF is a functional unification formalism which uses a large systemic grammar of English, called SURGE² to fill in syntactic constraints, build a syntactic tree, choose closed class words, and eventually linearize the tree as a sentence.

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. Finally, an FD or functional description (Elhadad, 1993) for the description is generated so that it can be reused in fluent ways in the final summary. FDs mix functional, semantic, syntactic, and lexical information in a recursive attribute-value format which serves as the basic data structure for all information within FUF/SURGE.

4. Generating the Summary

SUMMONS produces a summary from sets of templates which contain the salient facts reported in the input articles and which are produced by the message understanding conferences. These systems extract specific pieces of information from a given news article. Example of a template produced by MUC systems and used in our system is shown in Figure 2. To test our system, we used the templates produced by systems participating in MUC-4 (MUC, 1992) 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 per template. In addition, we filled the same template forms by hand from current news articles for further testing. 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. We are basing our implementation on the tools developed at the University of Mas-

² 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.

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 2
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 3
Parsed MUC-4 Template.

sachusetts (Fisher et al., 1995). The resulting system will not only be able to generate summaries from pre-parsed templates but will also produce summaries directly from raw text by merging the message understanding component with the current version of SUMMONS.

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 developing the planning operators and for 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, our corpus also includes 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 planning 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. This paragraph summarizes four articles about two separate terrorist acts that

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

Sample output from SUMMONS.

took place in Israel in March of 1996 using two different planning operators.

While the system we report on is fully implemented, our work is undergoing continuous development. Currently, the system includes eight different planning operators, a testbed of 200 input templates grouped into sets on the same event, and can produce fully lexicalized summaries for approximately half of the cases (the rest of the templates were either not complete or the information extracted in them was irrelevant to the task). We haven't performed an evaluation beyond the testbed.

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 cycle³. 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 section 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 realize the conceptual representation into a sentence. The lexicon contains the vocabulary for the system and encodes constraints about when each word can be used. As shown in Figure 1, the content planner used by SUMMONS determines what information from the input MUC templates should be included in the summary using a set of planning operators that are specific to summarization and to some extent, the terrorist domain. Its linguistic component determines the phrases and surface syntactic form of the summary. The linguistic component consists of a lexical chooser, which determines the high level sentence structure of each sentence and the

³ 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.

words which realize each semantic role, and the FUF/SURGE (Elhadad, 1991; Elhadad, 1993) sentence generator.

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. However, we constructed by hand an additional set of templates that include also terrorist events that have taken place after the period of time covered in MUC-4, such as the World Trade Center bombing, the Hebron Mosque massacre and more recent incidents in Israel, as well as the disaster in Oklahoma City. These incidents were not handled by the original message understanding systems. We also created by hand a set of templates unrelated to real newswire articles which we used for testing some techniques of our system. 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 reports⁴. We found having the source of the report immensely useful for discovering and reporting contradictions and generalizations, because often different reports of an event are in conflict. Also, source information can indicate the level of confidence of the report, particularly when reported information changes over time. For example, if several secondary sources all report the same facts for a single event, citing multiple primary sources, it is more likely that this is the way the event really happened, while if there are many contradictions between reports, it is likely that the facts are not yet fully known.

Members of our research group are currently working on event tracking (Aho et al., 1997). Their prototype 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.

SUMMONS's summarization component generates a base summary, which contains facts extracted from the input set of articles. The base summary is later enhanced with additional facts from online structured databases with descriptions of individuals extracted from previous news to produce the 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. In the absence of a specific user model, the base summary is produced. Otherwise, the extended summary (base summary with added descriptions of entities) is generated instead. Similarly, the default is that the summary contains references to contradictory and updated information. However, if the user profile makes it explicit, only the latest and the most trusted (as per the user's preference of sources) facts are included.

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.

Development of the text generation component 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 2.3.

In particular, we used FUF to implement the lexical chooser, representing the lexicon as a grammar as we have done in many previous systems (Elhadad, 1993; Robin, 1994; McKeown, Robin, and Tanenblatt, 1993; Feiner and McKeown, 1991). The main effort in porting the approach to SUMMONS was in identifying the words and phrases needed for the domain. The content planner features several stages. It first groups news

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

articles together, identifies commonalities between them, and notes how the discourse influences wording by setting realization flags, which denote such discourse features as “similarity” and “contradiction”. Realization flags (McKeown, Kukich, and Shaw, 1994b) guide the choice of connectives in the generation stage.

Before lexical choice, SUMMONS maps the templates into FDs which are expected as input to FUF and uses a domain ontology (derived from the ontologies represented in the message understanding systems) to enrich the input. For example, grenades and bombs are both explosives, while diplomats and civilians are both considered to be human targets.

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. The corpus of summaries is 2.5 MB in size. We have manually grouped 300 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 the features of a briefing is conciseness, we have tried to assemble small paragraph summaries which in essence describe a single event and the change of perception of the event over time, or a series of related events with no more than a few sentences.

4.3 Summary operators for content planning

The main point of departure for SUMMONS from previous work 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 items to summarize are very similar and the problem is to form a grouping that puts the most similar items 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 reported facts have 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.

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.

The system identifies a preeminent set of templates from the input to the MUC 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 (1993) and McKeown *et al.* (1994b) suggest.

At each step, a summary operator is selected based on existing similarities between

```
( (#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 5

Rules for the Contradiction operator.

articles in the database. This operator is then applied to the input templates, resulting in a new template which combines, or synthesizes, information from the old. Each operator is independent 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 to its input. Figure 5 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, apply the contradiction operator to combine the templates.

A summary operator encodes a means for linking information in two different templates. Often it results in the synthesis of new information. For example, a generalization may be formed from two independent facts. Alternatively, since we are summarizing reports written over time, highlighting how knowledge of the event changed is important and therefore, summaries sometimes must identify differences between reports. A description of the operators we identified in our corpus follows, accompanied by an example of system output for each operator. Each example primarily summarizes two or three 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 example (Figures 6– 11).

4.3.1 Change of perspective. When an initial report gets a fact wrong or has incomplete information, the change is usually included in the summary. In order for the “change of perspective” operator to apply, the `SOURCE` field 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.

Example: March 4th, Reuters reported that a bomb in Tel Aviv killed at least 10 people and wounded 30. *Later the same day*, Reuters reported that *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` uses this feature to report disagreement between output by different systems. A summary might indicate that one of the sources determined that 20 people were killed, while the other source determined

that only 5 were indeed killed. The difference between this example and the previous one on “change of perspective” is the source of the update. If the same source announces a change, then we know that it is reporting 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 and we can therefore conclude that we have a contradiction.

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

4.3.3 Addition. When a subsequent report indicates that additional facts are 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. Since the former template doesn’t contain a value for the perpetrator slot and the latter contains information about claimed responsibility, we can apply the addition operator.

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

4.3.4 Refinement. In subsequent reports a more general piece of information may be refined. Thus, if an event is originally reported to have occurred in New York City, the location might later be specified as a particular borough of the city. Similarly, if a terrorist group is identified as Palestinian, later the exact name of the terrorist group may be determined. Since the update is assigned a higher value of “importance”, it will be favored over the original article in a shorter summary. Unlike the previous example, there was a value for the perpetrator slot in the first template, while the second one further elaborates on it, specifying the perpetrator more specifically.

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

4.3.5 Agreement. If two sources have the same values for a specific slot, this will heighten the reader’s confidence in their veracity and thus, agreement between sources is usually reported.

Example: 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 aggregate multiple events as shown in the example.

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.

It should be noted that in this example, the third sentence will not be generated if there is a restriction on the length of the summary.

4.3.7 Trend. There is a trend if two or more articles reflect similar patterns over time. Thus, we might notice that three consecutive bombings occurred at the same location and summarize them into a single sentence. The trend operator was only partially implemented at the time of the writing.

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

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

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

4.4 Algorithm

The algorithm used in the system to sort, combine, and generalize the input templates 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 templates from a related domain. All templates are described as lists of attribute/value pairs (as shown in Figure 7). 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.
- Templates that have obviously been incorrectly generated by a MUC system are identified and filtered manually. This includes templates left blank or mostly unfilled by the MUC system.
- A database of all fields and templates is created. This database is used later as a basis for grouping and collapsing templates.
- All irrelevant fields or fields containing bad values are manually marked as such and don't participate in further analyses.
- Templates related to the same event are manually grouped into sets for combination using SUMMONS.

- 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.

We are experimenting with some techniques to automate the preprocessing stage. Our preliminary impressions show that by restricting SUMMONS on templates in which at least five or six slots are filled, we can eliminate most of the irrelevant templates.

4.4.3 Heuristic combination. The template database is scanned for relationships between templates which will trigger certain operators. Since slots are matched among templates in chronological order, there is only one sequence in which they can be applied.

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 template will be greater than the sum of the individual “importance” of the constituent templates. At the same time, the values of these two templates are lowered (still keeping a higher value on the later one, which is assumed 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).

Thus, the final summary will contain only the combined template if there are restrictions on length. Otherwise, text corresponding to the constituent templates will be also generated.

The value of the “importance” of the template corresponds also to the position in the summary paragraph, as more important templates 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 templates which are still active and can be further combined. This way we make sure that all templates still have a chance of participating in the actual summary.

The resulting templates 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 templates included in the database after the Heuristic Combination stage, the content planner organizes the presentation of information within a paragraph.

It looks at consecutive templates in the database, marked as separate paragraphs from the previous stage, and assigns values to “realization switches” 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 Ordering of templates and linguistic generation. After all templates have been converted into FDs, SUMMONS carries out the following steps in order to produce the base summary:

- Templates are sorted according to the order of the value of the “importance” slot. Only the top templates are realized. Templates with higher importance values appear with priority in the summary if a restriction on length is specified.
- An intermediate module, the ontologizer (part of the Base Summary Generator shown in Figure 1), converts factual information from the template database into data structures compatible with the ontology of the MUC domain. This is used, for example, to make generalizations (e.g., that Medellín and Bogotá 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. For example, it can generate verbs or nominal constructs for nodes in the MUC hierarchy (e.g., “kidnapping” vs. “X kidnapped Y”).
- Surface generation from the augmented template FDs is performed using FUF and SURGE. We have written additional generation code to handle paragraph-level constructions related to the summarization operators.

4.5 An example of system operation

This subsection 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 6.

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

Let’s now consider the four templates in the order that they appear in the list of templates. These templates are shown in Figures 7 – 10. They are generated manually from the input newswire texts. Information about the primary and secondary sources of information is added (PRIMSOURCE and 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 and is repeated here in Figure 11.

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

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

The description for Hamas (“*radical Muslim group*”) was added by the extraction generator (see Section 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, SUMMONS generates a description from the database of extracted descriptions. We are currently working on an algorithm which

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

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

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

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

Figure 6
Fragments of input articles 1-4.

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

Figure 7
Template for article one.

MESSAGE: ID	TST-REU-0002
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 07:20
PRMSOURCE: 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 8
Template for article two.

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?).

MESSAGE: ID	TST-REU-0003
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 14:20
PRMSOURCE: 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 9
Template for article three.

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

Figure 10
Template for article four.

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

Figure 11
SUMMONS output based on the four articles.

5. 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 the other sources (Radev and McKeown, 1997).

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 Section 4, if the user hasn't been informed about what Hamas is and no description is available in the source template, older descriptions in the FD format can be retrieved and used.

In this section, we describe an enhancement to the base summarization system,

called the *profile manager*, which tracks prior references to a given entity by extracting descriptions for later use in summarization. The component includes the “Entity Extractor” and “Description Extractor” modules shown in Figure 1 and has the following features:

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

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

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

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 section discusses the stages involved in the collection and reuse of descriptions.

5.1 Creation of a Database of Profiles

In this subsection, we describe the description management module of SUMMONS shown in Figure 1. We explain how entity names and descriptions for them are extracted from old newswire and how these descriptions are converted to FDs for surface generation.

5.1.1 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. Later, we used a Web-based interface which allowed anyone on the Internet to type in an entity name and force a robot to search for documents containing mentions of the entity and extract the relevant descriptions. These descriptions are then also added to the database.

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

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

Table 1

Two-word and three-word sequences retrieved by the system.

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 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.

5.1.2 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 Subsection 5.1.1) and exists in the database of descriptions. The second case is when we want a description to be retrieved in real time based on a request from the generation component.

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

These representations are fed to CREP which extracts noun phrases on either side of the entity (either pre-modifiers or appositions) from the news corpus. The 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 2 shows some of the different patterns retrieved. For example, when the profile manager has retrieved the description “the political arm of the Irish Republican Army” for Sinn Fein, it looks at the head noun in the description NP (“arm”) which we manually added to the list of trigger words to be categorized as an organization (see next subsection). It is important to notice that even though WordNet typically presents problems with disambiguation of words retrieved from arbitrary text, we don’t have

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

Table 2
Examples of retrieved descriptions.

```

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

Figure 12
Profile for John Major.

any trouble disambiguating “arm” in this case due to the constraints on the context in which it appears (as an apposition describing an entity).

5.1.3 Categorization of descriptions. We use WordNet to group extracted descriptions into categories. For the head noun of the description NP, 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 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. We have currently a list of 75 such trigger words that we have compiled manually.

5.1.4 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 12 shows the profile associated with the key “John Major”. It can be seen that four different descriptions have been used in the parsed corpus to describe John Major. Two of the four are common and are used in SUMMONS, whereas the other two result from incorrect processing by POS and/or CREP.

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

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

Figure 13
Retrieved description for Silvio Berlusconi.

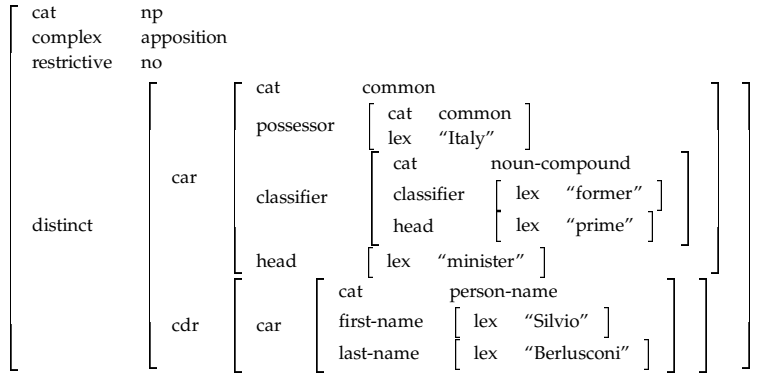


Figure 14
Generated FD for Silvio Berlusconi.

5.2 Generation of Descriptions

The content planner of a language generation system that presents an entity to the user that he has not seen previously may decide to include some background information about it.

When the extracted information doesn't contain an appropriate description, the system can use some descriptions retrieved by the profile manager.

5.2.1 Transformation of descriptions into Functional Descriptions. In order to reuse the extracted descriptions in the generation of summaries, we have developed a module 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 13. The corresponding FD is shown in Figure 14.

5.2.2 Regenerating Descriptions. We have completed tools to extract the descriptions and to represent them into FDs but we haven't yet implemented the module for including them in the output summary. We have focused so far on identifying when this kind of generation will be needed:

- **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".
- **Unification with existing ontologies.** E.g., if an ontology contains information about the word "president" as being a realization of the concept "head of state", then under certain conditions, the description can be replaced by one referring to "head of state".
- **Generation of referring expressions.** In the previous example, if "president Bill Clinton" is used in a sentence, then "head of state" can be used as a referring expression in a subsequent sentence.

- **Modification/Update of descriptions.** If we have retrieved “prime minister” as a description for Silvio Berlusconi, and later we 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. For example, when an article concerns Bill Clinton on the campaign trail, then the description “democratic presidential candidate” is more appropriate. On the other hand, when an article concerns an international summit of world leaders, then the description “U.S. President” is more appropriate.
- **Merging lexicons.** The lexicon generated automatically by the system can be merged with a manually compiled domain lexicon.

6. System Status

6.1 Summary Generation

Currently, our system can produce simple summaries consisting of 1-3 sentence paragraph which are limited to the MUC domain and to a few additional events for which we have manually created MUC-like templates. We have also implemented the modules to connect to the World Factbook. We have converted all ontologies related to the MUC and the Factbook into FDs. The user model which would allow users to specify preferred sources of information, frequency of briefings, etc. hasn’t been fully implemented yet.

A problem that we haven’t addressed is related to the clustering of articles according to their relevance to a specific event. This is an area of research where further work needs to be done. Another such area is the development of algorithms for grouping together articles that belong to the same topic.

Finally, one of our main topics for future work is the development of techniques that can generate summary updates. To do this, we must make use of a discourse model that represents 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 previously described.

6.2 The Description Generator

At the current stage, 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 Subsection 5.1.1 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 Subsection 5.1.1. 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.

6.2.1 Limitations. Our system currently doesn’t handle entity cross-referencing. It will not realize that “Clinton” and “Bill Clinton” refer to the same person. Nor will it link a person’s profile with the profile of the organization of which he is a member. We should note that extensive research in this field exists and we plan to make use of one of the proposed methods (Wacholder, Ravin, and Choi, 1997) to solve this problem.

6.3 Portability

An important issue is portability of SUMMONS to other domains. There are no a priori restrictions in our approach that would limit SUMMONS to template-based inputs (and hence, shallow knowledge-representation schemes without recursion). It would be interesting to determine the actual number of different representation schemes for news in general.

Since there exist systems that can learn extraction rules for unrestricted domains (Lehnert et al., 1993), the information extraction doesn’t seem to present any fundamental bottleneck either. Rather the questions are how many man-hours are required to convert to each new domain and how many of the rules from one domain are applicable to each new domain. There are no clear answers to these questions. The library of planning operators used in SUMMONS is certainly extensible and can be ported to other domains, although it is likely that new operators will be needed. In addition, new vocabulary will also be needed. The authors plan to perform a portability analysis and report on it in the future.

6.4 Suggested Evaluation

Given that no alternative approaches to conceptual summarization of multiple articles exist, we have found it very hard to perform an adequate evaluation of the summaries generated by SUMMONS. We consider several potential evaluations: qualitative (user satisfaction and readability) and task-based. In a task-based evaluation, one set of judges would have access to the full set of articles, while another set of evaluators would have the summaries generated by SUMMONS. The task would involve decision making (e.g., deciding whether the same organization has been involved in multiple incidents). The time for decision making will be plotted against the accuracy of the answers provided by the judges from the two sets. A third set of judges might have access to summaries generated by summarizers based on sentence-extraction from multiple documents. Similar evaluation techniques have been proposed for single-document summarizers (Jing et al., 1998).

7. Future Work

The prototype system that we have developed serves as the springboard for research in a variety of directions. First and foremost is the need to use statistical techniques to increase the robustness and vocabulary of the system. Since we were looking for

phrasings that mark summarization in a full article that includes other material as well, for a first pass we found it necessary to do a manual analysis in order to determine which phrases were used for summarization. In other words, we knew of no automatic way of identifying summary phrases. However, having an initial seed set of summary phrases might allow us to automate a second pass analysis of the corpus by looking for variant patterns of the ones we have found.

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

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

7.1 Generation of Descriptions

One of the more important current goals is to increase coverage of the system by providing interfaces to a large number of on-line sources of news. We would ideally want to build a comprehensive and shareable database of profiles that can be queried over the World-Wide Web. The database will have a defined interface which will allow for systems such as SUMMONS to connect to it.

Another goal of our research is the generation of evolving summaries that continuously update the user on a given topic of interest. In that case, the system will have a model containing all prior interaction with the user. To avoid repetitiveness, such a system will have to resort to using different descriptions (as well as referring expressions) to address a specific entity⁵. We will be investigating an algorithm that will select a proper ordering of multiple descriptions referring to the same person within the same discourse.

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 are 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, and 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 recently 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

⁵ Our corpus analysis supports this proposition – a large number of threads of summaries on the same topic from the Reuters and UPI newswire used up to 10 different referring expressions (mostly of the type of descriptions discussed in this paper, but also anaphoric references) to refer to the same entity.

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 as well as alternative spellings of the same name.

We will also look into connecting the current interface with news available on the Internet and with an existing search engine such as Lycos, AltaVista, or Yahoo. We can then use the existing indices of all Web documents mentioning a given entity as a news corpus on which to perform the extraction of descriptions.

8. Conclusion

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

Acknowledgments

This work was partially supported by NSF grants GER-90-24069, IRI-96-19124, and CDA-96-25374, as well as a grant from Columbia University's Strategic Initiative Fund sponsored by the Provost's Office. The authors are grateful to the following people for their invaluable comments during the writing of the paper and at presentations of work related to the content of the paper: Alfred Aho, Shih-Fu Chang, Eleazar Eskin, Vasileios Hatzivassiloglou, Alejandro Jaimes, Hongyan Jing, Judith Klavans, Min-Yen Kan, Carl Sable, Eric Siegel, John Smith, Nina Wacholder, Kazi Zaman as well as the anonymous reviewers and the editors of the Special Issue on Natural Language Generation.

References

- Aberdeen, John; Burger, John; Connolly, Dennis; Roberts, Susan; and Vilain, Marc (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, June 1992.
- Aho, Alfred; Chang, Shih-Fu; McKeown, Kathleen; Radev, Dragomir; Smith, John; and Zaman, Kazi (1997). "Columbia Digital News System : An Environment for Briefing and Search over Multimedia Information." In *Proceedings of ADL*, Washington, DC, April 1997.
- (1996). Altavista. WWW site, URL: <http://altavista.digital.com>.
- Ayuso, Damaris; Boisen, Sean; Fox, Heidi; Gish, Herb; Ingria, Robert; and Weischedel, Ralph (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, June 1992.
- Barzilay, Regina and Elhadad, Michael (1997). "Using Lexical Chains for Text Summarization." In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 10–17, Madrid, Spain, August 1997. Association for Computational Linguistics.
- Berners-Lee, Tim (1992). World-Wide Web: The Information Universe. *Electronic Networking*, 2(1):52–58.

- Boguraev, Branimir and Kennedy, Christopher (1997). "Salience-based Content Characterization of Text Documents." In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 2–9, Madrid, Spain, August 1997. Association for Computational Linguistics.
- Borko, H. (1975). *Abstracting Concepts and Methods*. Academic Press, New York, N.Y.
- Bourbeau, Laurent; Carcagno, Denis; Goldberg, E.; Kittredge, Richard; and Polguère, Alain (1990). "Bilingual Generation of Weather Forecasts in an Operations Environment." In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, edited by Hans Karlgren, volume 3, pages 318–320, Helsinki, Finland.
- Brandow, Ronald; Mitze, Karl; and Rau, Lisa F. (1990). Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 26:135–170.
- Church, Kenneth 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, February 1988. Association for Computational Linguistics.
- (1996). ClariNet. WWW site, URL: <http://www.clari.net>.
- (1996). CNN Interactive. WWW site, URL: <http://www.cnn.com>.
- Coates-Stephens, Sam (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, Jim; Guthrie, Louise; Wilks, Yorick; Pustejovsky, James; and Waterman, Scott (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, June 1992.
- Cuts, Short (1994). Science and Technology Section. *Economist*, 17:85–86, December 1994.
- Dalianis, Hercules and Hovy, Edward (1993). Aggregation in Natural Language Generation. *Proceedings of the 4th European Workshop on Natural Language Generation*.
- (1997). DejaNews. WWW site, URL: <http://www.dejanews.com>.
- DeJong, G.F. (1979). *Skimming stories in real time: an experiment in integrated understanding*. Doctoral dissertation, Computer Science Department, Yale University.
- Duford, Darrin (1993). CREP: a regular expression-matching textual corpus tool. Technical Report CUCS-005-93, Columbia University.
- Elhadad, Michael (1991). FUF: The universal unifier - user manual, version 5.0. Technical Report CUCS-038-91, Columbia University.
- Elhadad, Michael (1993). *Using argumentation to control lexical choice: a unification-based implementation*. Doctoral dissertation, Computer Science Department, Columbia University.
- Endres-Niggemeyer, Brigitte (1993). "An Empirical Process Model of Abstracting." In *Workshop on Summarizing Text for Intelligent Communication*, Dagstuhl, Germany, December 1993.
- Feiner, Steven and McKeown, Kathleen (1991). Automating the Generation of Coordinated Multimedia Explanations. *IEEE Computer*, 24(10):33–41, October 1991.
- Fisher, David; Soderland, Stephen; McCarthy, Joseph; Feng, Fangfang; and Lehnert, Wendy (1995). "Description of the UMass System As Used for MUC-6." In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 221–236.
- Genesereth, Michael and Ketchpel, Steven (1994). Software Agents. *Communications of the ACM*, 37(7):48–53, July 1994.
- Hahn, Udo (1990). Topic parsing: accounting for text macro structures in full-text analysis. *Information Processing and Management*, 26:135–170.
- Halliday, Michael and Hasan, Ruqaiya (1976). *Cohesion in English*. English Language Series. Longman, London.
- Hovy, Eduard and Lin, Chin Yew (1997). "Automated Text Summarization in SUMMARIST." In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 18–24, Madrid, Spain, August 1997. Association for Computational Linguistics.
- Hovy, Eduard H. (1988). "Planning Coherent Multisentential Text." In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, N.Y., June 1988. Association for Computational Linguistics.
- Iordanskaja, Lidija; Kim, M.; Kittredge, Richard; Lavoie, Benoit; and Polguère, Alain (1994). "Generation of Extended Bilingual Statistical Reports." In

- Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan.
- Jing, Hongyan; Barzilay, Regina; McKeown, Kathleen; and Elhadad, Michael (1998). "Summarization Evaluation Methods: Experiments and Analysis." In *Symposium on Intelligent Text Summarization*, Stanford, California, 3 1998. American Association for Artificial Intelligence.
- Kukich, Karen; Passonneau, Rebecca; McKeown, Kathleen; Radev, Dragomir; Hatzivassiloglou, Vasileios; and Jing, Hongyan (1997). "Software re-use and evolution in text generation applications." In *ACL/EACL Workshop - From Research to Commercial Applications: Making NLP Technology Work in Practice*, Madrid, Spain.
- Kukich, Karen K. (1983). "Design of a Knowledge-Based Report Generator." In *Proceedings of the 21st Annual Meeting of the ACL*, pages 145–150, Cambridge, Massachusetts, June 1983. Association for Computational Linguistics.
- Kupiec, Julian 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, Julian M.; Pedersen, Jan; and Chen, Francine (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, July 1995.
- Lehnert, Wendy; McCarthy, Joe; Soderland, Stephen; Riloff, Ellen; Cardie, Claire; Peterson, Jonathan; and Feng, Fangfang (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., August 1993.
- Luhn, Hans P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal*, pages 159–165.
- (1996). Lycos, Inc. Home Page. WWW site, URL: <http://www.lycos.com>.
- Mani, Inderjeet and Bloedorn, Eric (1997). "Multi-document Summarization by Graph Search and Matching." In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 622–628, Providence, Rhode Island. American Association for Artificial Intelligence.
- Mani, Inderjeet; Macmillan, Richard T.; Luperfoy, Susann; Lusher, Elaine; and Laskowski, Sharon (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, June 1993. Special Interest Group on the Lexicon of the Association for Computational Linguistics.
- Marcu, Daniel (1997). "From Discourse Structures to Text Summaries." In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain, August 1997. Association for Computational Linguistics.
- McDonald, David D. (1993). "Internal and External Evidence in the Identification and Semantic Categorization of Proper Names." In *Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text*, pages 32–43, Columbus, Ohio, June 1993. Special Interest Group on the Lexicon of the Association for Computational Linguistics.
- McDonald, David D. and Pustejovsky, James D. (1986). "Description-directed natural language generation." In *Proceedings of the 9th IJCAI*, pages 799–805. IJCAI.
- McKeown, Kathleen R. (1985). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Texts*. Cambridge University Press, Cambridge, England.
- McKeown, Kathleen R.; Kukich, Karen; and Shaw, James (1994a). "Practical Issues in Automatic Documentation Generation." In *Proceedings of the 4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, October 1994. Association for Computational Linguistics.
- McKeown, Kathleen R.; Kukich, Karen K.; and Shaw, James (1994b). "Practical Issues in Automatic Documentation Generation." In *Proceedings of the ACL Applied Natural Language Conference*, Stuttgart, Germany, October 1994.
- McKeown, Kathleen R. and Radev, Dragomir 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, July 1995.
- McKeown, Kathleen R.; Robin, Jacques; and Kukich, Karen (1995). Generating Concise Natural Language Summaries. *Journal of Information Processing and Management*, 31(5):703–733.

- McKeown, Kathleen R.; Robin, Jacques; and Tanenblatt, Michael (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., June 1993.
- Miller, George A.; Beckwith, Richard; Fellbaum, Christiane; Gross, Derek; and Miller, Katherine J. (1990). Introduction to WordNet: An On-Line Lexical Database. *International Journal of Lexicography (special issue)*, 3(4):235-312.
- Mitra, Mandar; Singhal, Amit; and Buckley, Chris (1997). "Automatic Text Summarization by Paragraph Extraction." In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pages 39-46, Madrid, Spain, August 1997. Association for Computational Linguistics.
- MUC, Message Understanding Conference (1992). *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. DARPA Software and Intelligent Systems Technology Office.
- (1996). NetSumm Home Page. WWW site, URL: <http://www.labs.bt.com/innovate/informat/netsumm/index.htm>.
- (1996). New York Times. WWW site, URL: <http://www.nytimes.com>.
- Paice, Chris (1990). Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing and Management*, 26:171-186.
- Paik, Woojin; Liddy, Elizabeth D.; Yu, Edmund; and McKenna, Mary (1994). "Interpretation of Proper Nouns for Information Retrieval." In *Proceedings of the Human Language Technology Workshop*, pages 309-313, Plainsboro, New Jersey, March 1994. ARPA Software and Intelligent Systems Technology Office, Morgan Kaufmann, San Francisco, California.
- Passonneau, Rebecca; Kukich, Karen; McKeown, Kathleen; Radev, Dragomir; and Jing, Hongyan (1997). Summarizing Web Traffic: A Portability Exercise. Technical Report CUCS-009-97, Columbia University, Department of Computer Science, New York, NY, USA, March 1997.
- Preston, Keith and Williams, Sandra (1994). Managing the Information Overload. *Physics in Business*, June 1994.
- Radev, Dragomir 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, June 1996.
- Radev, Dragomir R. and McKeown, Kathleen R. (1997). "Building a Generation Knowledge Source using Internet-Accessible Newswire." In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington, DC, April 1997.
- Rau, Lisa F. (1988). "Conceptual Information Extraction and Information Retrieval from Natural Language Input." In *Proceedings, RAIO-88: Conference on User-Oriented, Content-Based, Text and Image Handling*, pages 424-437, Cambridge, Massachusetts.
- Rau, Lisa F.; Brandow, Ron; and Mitze, Karl (1994). "Domain-Independent Summarization of News." In *Summarizing Text for Intelligent Communication*, pages 71-75, Dagstuhl, Germany.
- (1996). Reuters News. WWW site, URL: <http://www.yahoo.com/headlines/>.
- Robin, Jacques (1994). *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. Doctoral dissertation, Computer Science Department, Columbia University.
- Robin, Jacques and McKeown, Kathleen R. (1993). "Corpus analysis for revision-based generation of complex sentences." In *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington, D.C., July 1993.
- Robin, Jacques and McKeown, Kathleen R. (1995). Empirically Designing and Evaluating a New Revision-Based Model for Summary Generation. *Artificial Intelligence Journal*. In press.
- Rösner, Michael (1987). "SEMTEX: A Text Generator for German." In *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, edited by Gerard Kempen. Martinus Nijhoff Publishers.
- Rothkegel, Annely (1993). "Abstracting from the perspective of text-production." In *Workshop on Summarizing Text for Intelligent Communication*, Dagstuhl, Germany, December 1993.
- Shaw, James (1995). "Conciseness through Aggregation in Text Generation." In *Proceedings of the 33rd ACL (Student Session)*, pages 329-331.
- Sparck Jones, K. (1993). "What might be in a summary?" In *Proceedings of Information Retrieval 93: Von der Modellierung zur Anwendung*, pages 9-26, Universitätsverlag Knstanz.

- Tait, John I. (1983). *Automatic summarising of English texts*. Doctoral dissertation, University of Cambridge, Cambridge, England.
- Wacholder, Nina; Ravin, Yael; and Choi, Misook (1997). "Disambiguation of proper names in text." In *Proceedings of the Fifth Applied Natural Language Processing Conference*, Washington, D.C. Association for Computational Linguistics.
- Weischedel, Ralph; Ayuso, Damaris; Boisen, Sean; Fox, Heidi; Ingria, Robert; Matsukawa, Tomoyoshi; Papageorgiou, Constantine; MacLaughlin, Dawn; Kitagawa, Masaichiro; Sakai, Tsutomu; Abe, June; Hoshi, Hiroto; Miyamoto, Yoichi; and Miller, Scott (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., August 1993.
- Young, S.R. and Hayes, P.J. (1985). "Automatic classification and summarization of banking telexes." In *Proceedings of the Second Conference on Artificial Intelligence Applications*, pages 402–408.