Generating Summaries of Multiple News Articles

Kathleen McKeown and Dragomir R. Radev

Department of Computer Science
Columbia University
New York, NY 10027
{kathy,radev}@cs.columbia.edu

Abstract

We present a natural language system which summarizes a series of news articles on the same event. It uses summarization operators, identified through empirical analysis of a corpus of news summaries, to group together templates from the output of the systems developed for ARPA's Message Understanding Conferences. Depending on the available resources (e.g., space), summaries of different length can be produced. Our research also provides a methodological framework for future work on the summarization task and on the evaluation of news summarization systems.

Keywords: Natural language summarization, Natural language generation, Summarization of multiple texts

1 Introduction

In this age of information overload, the ability to automatically summarize news stories would allow readers more ability to control the quantity of text that they read. Given the accuracy of current information retrieval systems, a typical search request returns many irrelevant documents. This is especially true for newswire information, since there are typically many articles on the same event. Online summaries could aid readers in determining if they want to access and read the full news articles as well as allow them to get the gist of the reported event by reading the summary only. While some previous approaches use statistical techniques to extract one or more sentences from the text which can serve as a summary with modest success (e.g., [Rau et. al. 1994; Paice 1990; Economist 1994]), summarization in general has remained an elusive task. In this paper, we present a system, SUMMONS (SUMMarizing Online NewS articles), to summarize full text input using templates produced by the message understanding systems developed under the ARPA human language technology program [MUC 1992]. Unlike previous approaches, our system summarizes a *series* of news articles on the same event, producing a paragraph consisting of one or more sentences. Our research focuses on techniques to summarize how the perception of an event changes over time, using multiple points of view over the same event or series of events.

Our system attempts to generate fluent text from sets of templates that contain the salient facts reported in the input texts. To produce these templates, we rely upon the ARPA message understanding systems. These systems accept full text as input, extracting specific pieces of information from a given newspaper article. To test our system, we used the templates produced by systems participating in MUC-4 [MUC 1992], available from the Linguistic Data Consortium (LDC), as input. MUC-4 systems operate on the terrorist domain and extract information by filling fields such as perpetrator, victim, and type of event, for a total number of 25 fields. In addition, we filled the same template forms by hand from current news articles for further testing[1]. Note that while our system uses templates as input, if it were integrated with one of the existing message understanding systems, the resulting larger system would automatically produce summaries of raw text in a modular way.

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. While critics of summarization have argued that it would be difficult to both develop principled summarization techniques and evaluate summarization systems, our approach indicates otherwise. We have collected a corpus of newswire summaries that we used as data for developing the planning operators and gathering a large set of lexical constructions used in summarization and which will eventually aid in a full system evaluation. Since news articles often summarize previous reports of the same event, we collected a corpus of articles which included short summaries of previous articles.

We used this corpus to develop both the content planner (i.e., the module which determines what information to include in the summary) and the linguistic component (i.e., the module which determines the words and surface syntactic form of the summary) of our system. We used the corpus to identify operators which are used to combine information; this includes techniques for linking information together in

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

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 1. This paragraph summarizes four articles on the World Trade Center bombing, using two different operators. The second sentence shows a contradiction between sources (Reuters and Associated Press) on the number of victims. The final sentence shows a refinement because the initial report did not contain information about the perpetrator. The resulting summary text uses lexical cues such as "however," "exactly," and "finally" to mark summary material.

> In the afternoon of February 26, 1993, Reuters reported that a suspected bomb killed at least five people in the World Trade Center. However, Associated Press announced that exactly five people were killed in the blast. Finally, Associated Press announced that Arab terrorists were possibly responsible for the terrorist act.

Figure 1: Use of multiple operators.

While the system we report on is fully implemented, our work is still at early stages. We need to increase the robustness of the system, which currently includes 7 different planning operators, a testbed of 60 input templates, and can produce content for all pairs of related input templates but fully lexicalized summaries for approximately 20 cases. Nonetheless, our work at this point shows that full text summarization using symbolic techniques is possible. It provides a methodology for increasing the vocabulary size and the robustness of the system using a collected corpus, and moreover, it shows how summarization can be used to evaluate the message understanding systems, identifying future research directions that would not be pursued under the current MUC evaluation cycle[2]. Due to inherent difficulties in the summarization task, our work is a substantial first step and provides the framework for a number of different research directions.

In the following sections, we provide a description of the components of SUMMONS, then turn to the planning operators for summarization, and 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 close by showing the demands that summarization creates for interpretation.

2 Overview of the system

SUMMONS is based on the traditional language generation system architecture [McKeown 1985; McDonald 1986; Hovy 1988]. A typical language generator is divided into two main components, a content planner, which selects information from an underlying knowledge base to include in a text, and a linguistic component, which selects words to refer to concepts contained in the selected information and arranges those words, appropriately inflecting them, to form an English sentence. The content planner produces a conceptual representation of text meaning (e.g., a frame, a logical form, or an internal representation of text) and typically does not include any linguistic information. The linguistic component uses a lexicon and a grammar of English to perform its task. The lexicon contains the vocabulary for the system and encodes constraints about when each word can be used. As shown in Figure 2, SUMMONS' content planner determines what information from the input MUC templates should be included in the summary using a set of planning operators that are specific to summarization and to some extent, the terrorist domain. Its linguistic component determines the phrases and surface syntactic form of the summary. The linguistic component consists of

- a lexical chooser, which determines the high level sentence structure of each sentence and the words which realize each semantic role, and
- the FUF (Functional Unification Formalism) [Elhadad 1991; Elhadad 1993] sentence generator, which uses a large systemic grammar of English, called SURGE[3] [Halliday 1985; Elhadad 1993; Robin 1994] to fill in syntactic constraints, build a syntactic tree, choose closed class words, and eventually linearize the tree as a sentence.

Input to SUMMONS is a set of templates, where each template represents the information extracted from one or more articles by a message understanding system. We restricted the domain to articles on terrorism, since this was what was available from the LDC. However, the hand-constructed templates included terrorist events such as the World Trade Center bombing, the Hebron Mosque massacre, and airline hijackings, which may or may not have been handled by the original message understanding systems. We also created by hand a set of templates unrelated to real newswire messages which we used for testing some techniques of our system. We enriched the templates for these cases by adding four slots: the primary source, the secondary source and the times when both sources made their reports[4]. We find having the source of the report immensely useful for summarization, because there are often conflicts between different reports of an event and these can indicate the level of confidence in the report, particularly as reports change over time. For example, if many sources all report the same incidents for a single event, it is more likely that this is the way the event really happened, while if there are many contradictions between reports, it is likely that the facts are not yet known. We assume that some other system has selected a reasonable set of templates, or articles, to summarize; that is, input should contain a set of templates which report on the same event. The articles may be written at any point in time and may be written by the same or many sources.

Output is a paragraph consisting of one or more sentences, where the length of the summary is controlled by a variable input parameter. At this point, we have no theory on how to determine the length of a summary, but assume that like any good paper writer, given more space, SUMMONS can use it to include more information. Information is rated 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

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

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

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

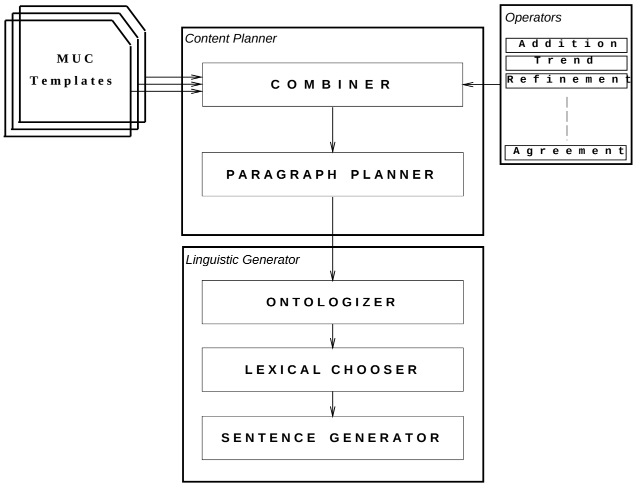

Figure 2: System Architecture.

highly. When space allows, SUMMONS may choose to include the base facts from two separate articles as well as the conclusion that can be drawn from both, while given less space, only the summarizing fact would be included.

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

The main point of departure for SUMMONS is in the stage of identifying what information to include and how to group it together, as well as the use of a corpus to guide this and later processes. In PLANDoc, successive messages are very similar and the problem is to form a grouping that puts the most similar messages together, allowing the use of conjunction and ellipsis to delete repetitive material. For summarizing multiple news articles, the task is almost the opposite; we need to find the differences from one article to the next, identifying how the news has changed. Thus, the main problem was the identification of summarization strategies, which indicate how information is linked together to form a concise and cohesive summary. As we have found in other work [Robin 1994], what information is included is often dependent on the language available to make concise additions. Thus, using a corpus summary was critical to identifying the different summaries possible.

3 Methodology: collecting and using a summary corpus

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

From the so collected corpora 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. Some examples of such phrases are included in Figures 7–9. In addition to the summary cue phrases collected from the corpus, we also tried to incorporate as many phrases as possible that have relevance to the message understanding conference domain. Due to domain variety, such phrases were essentially scarce in the newswire corpora and we needed to collect them from other sources (e.g., modifying templates that we acquired from the summary corpora to provide a wider coverage).

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

4 Summary operators for content planning

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

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

At each step, a summary operator is selected based on existing similarities between messages in the database. This operator is then applied to the input templates, resulting in a new template which combines, or synthesizes, information from the old. Each operator is independent from the other and several can be applied in succession to the input templates. Each of the seven major operators is further subdivided to cover various modifications of its input and output. Figure 3 shows part of the rules for the Contradiction operator.

$$\begin{gathered}((\#TEMPLATES == 2)\&\&\\ (T[1].INCIDENT.LOCATION == T[2].INCIDENT.LOCATION)\&\&\\ (T[1].INCIDENT.TIME < T[2].INCIDENT.TIME)\&\&\ldots\\ (T[1].SECSOURCE.SOURCE! = T[2].SECSOURCE.SOURCE)) ==>\\ (apply(``contradiction'', ``with-new-account'', T[1], T[2]))\end{gathered}$$

Given two templates, if INCIDENT.LOCATION is the same, the time of first report is before time of second report, the report sources are different, and at least one other slot differs in value (this rule not shown), apply the contradiction operator to combine the templates.

Figure 3: Rules for the Contradiction operator.

A summary operator encodes a means for linking information in two different templates. Often it results in synthesis of new information. For example, a generalization may be formed from two independent facts. Alternatively, since we are summarizing reports written over time, highlighting how knowledge of the event changed is important and thus, summaries sometimes must identify differences between reports. A description of the operators we identified in our corpus follows, accompanied by an example of system output for each operator. Each example primarily summarizes two input templates, as this is the result from applying a single operator once. More complex summaries can be produced by applying multiple operators on the same input, as shown in the introductory example.

4.1 Change of perspective

When an initial report gets a fact wrong or has incomplete information, the change is usually included in a summary. In order for this operator to apply, the source field must be the same, while the value of another field changes so that it is not compatible with the original value. For example, if the number of victims changes, we know that the first report was *wrong* if the number goes down, while the source had *incomplete information* (or additional people died) if the number goes up. The first two sentences from the following example were generated using the change of perspective operator. The initial estimate of "at least five people" killed in the incident becomes "exactly five people":

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

4.2 Contradiction

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

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

4.3 Addition

When a subsequent report indicates that additional facts became known, this is reported in a summary. Additional results of the event may occur after the initial report or additional information may become known. The operator determines this by the way the value of a template slot changes (up for numbers).

```
   January 1st 1994, Reuters announced that three
terrorists killed four civilians in the first
assault.  Later, Reuters reported that three people
were killed in the second assault.  A total of
seven people were killed in the two assaults.
```

4.4 Refinement

In subsequent reports a more general fact may be refined. Thus, if the location is originally reported to be New York City, it might later be noted as a particular borough of New York. Or, if a terrorist group is identified as Palestinian, later the exact name of the terrorist group may be determined. Since the update is assigned a higher value of "importance", it will be favored over the original message in a shorter summary. Sentence 3 from the introductory example was generated using the refinement operator, since the responsibility for the terrorist act was attributed to Arab terrorists, whereas earlier reports did not include perpetrator information.

```
   Finally, Associated Press announced that Arab
terrorists were possibly responsible for the
terrorist act.
```

4.5 Agreement

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

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

4.6 Superset

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.

```
   According to UPI, three terrorists were arrested
in Medellin last Tuesday.  Reuters announced that
the police arrested two drug traffickers in Bogota
last Wednesday.  A total of five criminals were
arrested in Colombia last week.
```

4.7 Trend

There is a trend if two or more messages reflect similar patterns over time. Thus, we might notice that three consecutive bombings occurred at the same location and summarize them into a single sentence. This is the only operator which is not implemented in the current version of the system.

4.8 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. An example of use of the *no information operator* is given in Figure 8.

5 Algorithm

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

5.1 Input

At this stage, the system receives a set of templates from the Message Understanding Conferences or a similar set of messages from a related domain. All templates are described as lists of attribute/value pairs (see Figure 5). These pairs are defined in the MUC guidelines [MUC 1992].

5.2 Preprocessing

This stage includes the following substages:

- The templates are sorted in chronological order. A later stage will take care of their conceptual ordering.
- Messages that have obviously been incorrectly generated by a MUC system are identified and filtered out by hand.
- A database of all fields and messages is created. This database is used later as a basis for grouping and collapsing messages.
- All irrelevant fields or fields containing bad values are manually marked as such and don't participate in further analyses.
- Knowledge of the source of the information is marked as the specific Message Understanding System for the site submitting the template if it is not present in the input template. Note that since the current Message Understanding Systems do not extract the source, this is the most specific we can be for such cases.

5.3 Heuristic combination

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

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

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

5.4 Discourse planning

Given the relative importance of the messages included in the database after the Heuristic Combination stage, the content planner is called to organize the presentation of information within a paragraph. It looks at consecutive messages in the database, marked as separate paragraphs from the previous stage, and assigns values to "realization switches" [McKeown et. al. 1994] 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.

5.5 Format conversion

All messages included in the database and augmented through the content planner are sent to a so-called "lispize" module which converts the records of the message database[5] into FUF Functional descriptions (FD's) [Elhadad 1993].

5.6 Ordering of templates and linguistic generation

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

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

[5] All database access functions are written in PERL. Acknowledgments to its creator, Larry Wall.

- The surface generation from the augmented message FD's is performed using SURGE and FUF. We have written additional generation code to handle paragraph-level constructions (the summarization operators).

6 An example of system operation

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

> An explosion apparently caused by a car
> bomb in an underground garage shook the
> World Trade Center in lower Manhattan with
> the force of a small earthquake shortly
> after noon yesterday, collapsing walls and
> floors, igniting fires and plunging the
> city's largest building complex into a
> maelstrom of smoke, darkness and fearful
> chaos.
> The police said the blast killed at least
> five people and left more than 650 others
> injured, mostly with smoke inhalation or
> minor burns.

Figure 4: Excerpts from the initial newswire and newspaper articles.

The four news articles result in four different templates which correspond to four separate accounts of the same event and will be included in the set of templates from which the template combiner will work. Initially, all four templates will be given equal importance. Since two of the templates contain similar information, they will be replaced by a single one (actually, one of the two will be assigned a negative "importance" so that it will be ignored in the later stages). In this case, the incident type, the location, and the date are the same in the two templates and therefore, the heuristic combiner will replace them with a single template. Since there are no other templates that need to have their "importance" values adjusted, the factor used in ordering the output will be chronological order.

Let's now consider the first two templates in the order that they appear in the list of templates (note, however, that the output example in Figure 1 covers all three relevant templates). These templates are shown in Figure 5 and Figure 6 respectively. They are generated manually from the input newswire texts. Information about the primary and secondary sources of information (PRIMSOURCE and SECSOURCE) is added. Due to lack of space, only a few template slots are shown. The differences in the two templates (which will trigger certain operators) are shown in **bold face**.

MESSAGE: ID	TST-COL-0001
SECSOURCE: SOURCE	**Reuters**
SECSOURCE: DATE	**26 FEB 93** **EARLY AFTERNOON**
INCIDENT: DATE	26 FEB 93
INCIDENT: LOCATION	WORLD TRADE CENTER
INCIDENT: TYPE	BOMBING
HUM TGT: NUMBER	**AT LEAST 5**

Figure 5: Template for newswire article 1.

MESSAGE: ID	TST-COL-0002
SECSOURCE: SOURCE	**Associated Press**
SECSOURCE: DATE	**26 FEB 93 19:00**
INCIDENT: DATE	26 FEB 93
INCIDENT: LOCATION	WORLD TRADE CENTER
INCIDENT: TYPE	BOMBING
HUM TGT: NUMBER	**5**

Figure 6: Template for newswire article 2.

The different values of SECSOURCE:SOURCE and SECSOURCE:DATE in the two message templates trigger the activation of the contradiction operator. That operator makes use of the fact that there has been an apparent change in the number of victims (more specifically, the **perception** of this change between two sources of information, in this case, two SECSOURCEs). By looking at the diverging values as well as the times of the two reports, the content planner, activates the contradiction operator and decides that:

- the information contained in the second template should follow the first one.
- there is a logical connection between the two templates in that a value of one field changes over time. Accordingly, the content planner includes an appropriate realization switch for the second template which will be used by the lexical chooser module.

On the other hand, the lexical choice component of the linguistic module will make the following word choices specifically related to summarization:

- Since the "with-new-account" realization switch is present, the linguistics module will include an appropriate connective (in this case "later").
- Because of the change in the value of a specific field (in this case, "HUM TGT: NUMBER"), a cue phrase "exactly" will be inserted.

The different operators specify what combinations of values in certain template fields are used in the output.

7 Phrases used in summarization

Phrases used in summarization, such as connectives, and explicit examples of summarization phrases were collected from newswire corpora. Some example summarization phrases taken from the corpora and implemented in the system are shown below. The sequence in Figure 7 illustrates the use of summary cues, shown in italics, as well as the use of two operators, *addition* and *no information*. This sequence is a short summary appearing in a later article, highlighting changes. The sequence in Figure 8 illustrates the heavy reliance on multiple sources and disagreement between them to aid in conveying the level of confidence in what is actually known. This sequence also illustrates the *no information* operator; the text explicitly mentions that certain information was not reported, when all other indications lead one to expect it should have been. Finally, Figure 9 shows a sequence of text from an earlier article and the summary sentence which appeared in a later article, with summary cue phrases italicized. These are but a few examples from our collection.

... *another* ten people were killed ...
Reuters *didn't confirm* the shooting
in addition to this killing.

Figure 7: Summary cues.

NICOSIA, Cyprus (AP) – Two bombs exploded near government ministries in Baghdad, but there was no immediate word of any casualties, Iraqi dissidents reported Friday. There was no independent confirmation of the claims by the Iraqi National Congress. Iraq's state-controlled media have not mentioned any bombings.

Figure 8: Multiple sources disagree.

8 Related work

There has been very little work on automatic text summarization primarily because it requires substantial capabilities in both interpretation and generation, and it is only recently that systems have reached these levels. In order to avoid solving the full natural language problem and to allow summarization in arbitrary domains, some researchers have applied statistical techniques to the summarization task [Rau et. al. 1994; Paice 1990; Economist 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 identify important phrases using various statistical measures. Such an approach can only work if there are sentences contained in the article which already serve as a summary. While this approach appears to have had modest success in some domains [Economist 1994], Rau reports that statistical summaries of individual news articles were rated lower by evaluators than simply using the lead sentence or two from the article. Paice [Paice 1990] also notes that problems for this approach center around accidentally including any pronouns which have no previous reference in the extracted text or, in the case of extracting several sentences, of including incoherent text when the extracted sentences are not consecutive in the original text and don't naturally follow one another. Paice has developed techniques for modifying the extracted text to replace unresolved references. Note, in any case, that these approaches cannot handle the task that we address, summarization of multiple articles, since this requires information about similarities and differences across articles.

Work in summarization using symbolic techniques has tended to focus more on identifying information in text that can serve as a summary (e.g., [Young and Hayes 1985; Rau 1988; Hahn 1990]) as opposed to generating the summary, and often relies heavily on scripts (e.g., [Dejong 1979; Tait 1983]). One exception is work at Cambridge University on identifying strategies for summarization [Sparck Jones 1993] which 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.

Sentences from earlier article:
The bodies of three men and a woman were pulled from the bay. The men were not immediately identified. Authorities were trying to identify the woman, Bergen said...

Summary sentence in later article:
In *the most serious incident of the year*,
four people drowned...

Figure 9: Summary cues.

9 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 signal 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 any new summarization strategies to add to our stock of operators.

By creating a corpus of summaries, we are now prepared to do a quantitative evaluation of the system. The best way to do the evaluation would be to run a message understanding system on earlier articles in a sequence of articles on the same event to create a set of templates as input and then score the automatically generated summary against the existing summaries in the corpus articles[6]. Scoring must rate the summary generator both on operators and phrasing, measuring coverage (did the generator use as wide a range of operators as occurred in the test corpus?) and match (did the generator use the same operators for a given set of input templates as were used in the corpus?). Since there is choice in how several articles are summarized, evaluation would need to use different summaries found in the corpus for the same earlier articles and rate the generated summary against these multiple models (see [Hatzivassiloglou and McKeown 1993] for an evaluation metric using multiple models).

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 mention; 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 form than a knowledge representation. This means that the generator's knowledge of different features of the event and relations between them is somewhat shallow. An ideal solution would be to team with a message understanding project so that advances in interpretation and in

[6] We would need to use a reserved portion of the corpus as test material, a portion which was not used in the original system development. In order to get a large enough test corpus, we would need to collect additional articles, but this is not difficult given the number of online news services.

generation could be influenced by demands and restrictions on each side.

We also plan to incorporate a multilingual component to our system so that we can generate summaries of the same series of articles in multiple languages. Our goal is to add a FUF grammar in one other language to start and to develop the content planning and lexicalization components so that they facilitate making choices in multiple languages simultaneously.

In addition to generation in multiple languages, one might expand the system to handle templates produced from news articles in multiple languages. Since the structure of the templates is not language-dependent, one might try to incorporate an understanding system in another language that would generate templates compatible with the ones used as input by our system.

10 Conclusions

Our prototype system demonstrates the feasibility of generating summaries of a series of news articles on the same event, highlighting changes over time. The ability to automatically provide summaries of textual material will critically aid in effective use of the internet in order to avoid overload of information. We show how planning operators can be used to synthesize summary content from individual templates, each representing a single article. These planning operators are empirically based, coming from analysis of existing summaries, and allow for the generation of concise summaries. Our framework allows for experimentation with different length summaries and for the combination of multiple, independent summary operators to produce more complex summaries.

Our system was developed under somewhat primitive conditions, without the ability to access and change the interpretation component of the system, without access to the full set of current output produced by message understanding systems, and without an existing corpus of summaries. Consequently, our results include the development of a methodological framework to ease future implementation of news summarization systems; this includes collecting the summary corpus, structuring it around threads of articles on the same events with later articles including summaries, and identifying summarizing phrases. This seed work will allow us to apply automated techniques (e.g., [Smadja 1991; Smadja, McKeown, and Hatzivassiloglou 1995; Robin 1994]) to further corpus analysis since we now have a database of lexical phrases that are used to mark summary material. By identifying the characteristics of these phrases, we can use this seed set to guide and control further automatic analysis. Such a corpus will also allow for evaluation of the generated summaries; development of an evaluation procedure will require additional work to develop a set of good metrics.

Acknowledgments: We wish to thank Jacques Robin and James Shaw for the help using FUF and SURGE as well as for some hints from their work on the PLANDoc system. We are also grateful to Vasileios Hatzivassiloglou for his thoughtful comments. This work was partially supported by NSF Faculty Award for Women GER-90-2406.

References

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

Short Cuts. Science and Technology Section. *Economist*, 17:85–86, December 1994.

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

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

S. Feiner and K.R. McKeown. Automating the Generation of Coordinated Multimedia Explanations. *IEEE Computer*, 24(10):33–41, October 1991.

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

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

V. Hatzivassiloglou and K.R. McKeown. Towards the Automatic Identification of Adjectival Scales: Clustering Adjectives According to Meaning. In *Proceedings of the 31st Conference of the ACL*, Columbus, Ohio, 1993. Association for Computational Linguistics.

E.H. Hovy. Planning Coherent Multisentential Text. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, N.Y., June 1988. Asscoication for Computational Linguistics.

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

K.R. McKeown, J. Robin, and M. Tanenblatt. 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.

K.R. McKeown, K.K. Kukich, and J. Shaw. Practical Issues in Automatic Documentation Generation. In *Proceedings of the ACL Applied Natural Language Conference*, Stuttgart, Germany, October 1994.

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

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

C. Paice. Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing and Management*, 26:171–186, 1990.

L.F. Rau, R. Brandow, and K. Mitze. Domain-Independent Summarization of News. In *Summarizing Text for Intelligent Communication*, pages 71–75, Dagstuhl, Germany, 1994.

L.F. Rau. 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, MA, 1988.

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

F. Smadja, K.R. McKeown, and V. Hatzivassiloglou. Automatic Development of Bilingual Lexicons. *Journal of Computational Linguistics, to appear*, 1995.

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

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

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

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