Information Fusion in the Context of Multi-Document Summarization

Regina Barzilay and **Kathleen R. McKeown**
Dept. of Computer Science
Columbia University
New York, NY 10027, USA

Michael Elhadad
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel

Abstract

We present a method to automatically generate a concise summary by identifying and synthesizing similar elements across related text from a set of multiple documents. Our approach is unique in its usage of language generation to reformulate the wording of the summary.

1 Introduction

Information overload has created an acute need for summarization. Typically, the same information is described by many different online documents. Hence, summaries that synthesize common information across documents and emphasize the differences would significantly help readers. Such a summary would be beneficial, for example, to a user who follows a single event through several newswires. In this paper, we present research on the automatic fusion of similar information across multiple documents using language generation to produce a concise summary.

We propose a method for summarizing a specific type of input: news articles presenting different descriptions of the same event. Hundreds of news stories on the same event are produced daily by news agencies. Repeated information about the event is a good indicator of its importancy to the event, and can be used for summary generation.

Most research on single document summarization, particularly for domain independent tasks, uses sentence extraction to produce a summary (Lin and Hovy, 1997; Marcu, 1997; Salton et al., 1991). In the case of multi-document summarization of articles about the same event, the original articles can include both similar and contradictory information. Extracting *all* similar sentences would produce a verbose and repetitive summary, while extracting *some* similar sentences could produce a summary biased towards some sources.

Instead, we move beyond sentence extraction, using a comparison of extracted similar sentences to select the phrases that should be included in the summary and sentence generation to reformulate them as new text. Our work is part of a full summarization system (McKeown et al., 1999), which extracts sets of similar sentences, *themes* (Eskin et al., 1999), in the first stage for input to the components described here.

Our model for multi-document summarization represents a number of departures from traditional language generation. Typically, language generation systems have access to a full semantic representation of the domain. A content planner selects and orders propositions from an underlying knowledge base to form text content. A sentence planner determines how to combine propositions into a single sentence, and a sentence generator realizes each set of combined propositions as a sentence, mapping from concepts to words and building syntactic structure. Our approach differs in the following ways:

- **Content planning operates over full sentences, producing sentence fragments.** Thus, content planning straddles the border between interpretation and generation. We preprocess the similar sentences using an existing shallow parser (Collins, 1996) and a mapping to predicate-argument structure. The content planner finds an intersection of phrases by comparing the predicate-argument structures; through this process it selects the phrases that can adequately convey the common information of the theme. It also orders selected phrases and augments them with

On 3th of September 1995, 120 hostages were released by Bosnian Serbs. Serbs were holding over 250 U.N. personnel. Bosnian serb leader Radovan Karadjic said he expected "a sign of goodwill" from the international community. U.S. F-16 fighter jet was shot down by Bosnian Serbs. Electronic beacon signals, which might have been transmitted by a downed U.S. fighter pilot in Bosnia, were no longer being received. After six days, O'Grady, downed pilot, was rescued by Marine force. The mission was carried out by CH-53 helicopters with an escort of missile- and rocket-armed Cobra helicopters.

Figure 1: Summary produced by our system using 12 news articles as input.

information needed for clarification (entity descriptions, temporal references, and newswire source references).

- **Sentence generation begins with phrases.** Our task is to produce fluent sentences that combine these phrases, arranging them in novel contexts. In this process, new grammatical constraints may be imposed and paraphrasing may be required. We developed techniques to map predicate-argument structure produced by the content-planner to the functional representation expected by FUF/SURGE(Elhadad, 1993; Robin, 1994) and to integrate new constraints on realization choice, using surface features in place of semantic or pragmatic ones typically used in sentence generation.

An example summary automatically generated by the system from our corpus of themes is shown in Figure 1. We collected a corpus of themes, that was divided into a training portion and a testing portion. We used the training data for identification of paraphrasing rules on which our comparison algorithm is built. The system we describe has been fully implemented and tested on a variety of input articles; there are, of course, many open research issues that we are continuing to explore.

In the following sections, we provide an overview of existing multi-document summarization systems, then we will detail our sentence comparison technique, and describe the sentence generation component. We provide examples of generated summaries and conclude with a discussion of evaluation.

2 Related Work

Automatic summarizers typically identify and extract the most important sentences from an input article. A variety of approaches exist for determining the salient sentences in the text: statistical techniques based on word distribution (Salton et al., 1991), symbolic techniques based on discourse structure (Marcu, 1997), and semantic relations between words (Barzilay and Elhadad, 1997). Extraction techniques can work only if summary sentences already appear in the article. Extraction cannot handle the task we address, because summarization of multiple documents requires information about similarities and differences across articles.

While most of the summarization work has focused on single articles, a few initial projects have started to study multi-document summarization documents. In constrained domains, *e.g.*, terrorism, a coherent summary of several articles can be generated, when a detailed semantic representation of the source text is available. For example, information extraction systems can be used to interpret the source text. In this framework, (Radev and McKeown, 1998) use generation techniques to highlight changes over time across input articles about the same event. In an arbitrary domain, statistical techniques are used to identify similarities and differences across documents. Some approaches directly exploit word distribution in the text (Salton et al., 1991; Carbonell and Goldstein, 1998). Recent work (Mani and Bloedorn, 1997) exploits semantic relations between text units for content representation, such as synonymy and co-reference. A spreading activation algorithm and graph matching is used to identify similarities and differences across documents. The output is presented as a set of paragraphs with similar and unique words highlighted. However, if the same information is mentioned several times in different documents, much of the summary will be redundant. While some researchers address this problem by selecting a subset of the repetitions (Carbonell and Goldstein, 1998), this approach is not always satisfactory. As we will see in the next section, we can both eliminate redundancy from the output and retain balance through the selection of common information.

On Friday, a U.S. F-16 fighter jet was shot down by Bosnian Serb missile while policing the no-fly zone over the region.

A Bosnian Serb missile shot down a U.S. F-16 over northern Bosnia on Friday.

On the eve of the meeting, a U.S. F-16 fighter was shot down while on a routine patrol over northern Bosnia.

O'Grady's F-16 fighter jet, based in Aviano, Italy, was shot down by a Bosnian Serb SA-6 anti-aircraft missile last Friday and hopes had diminished for finding him alive despite intermittent electronic signals from the area which later turned out to be a navigational beacon.

Figure 2: A collection of similar sentences — *theme*.

3 Content Selection: Theme Intersection

To avoid redundant statements in a summary, we could select one sentence from the set of similar sentences that meets some criteria (*e.g.*, a threshold number of common content words). Unfortunately, any representative sentence usually includes embedded phrases containing information that is *not* common to other similar sentences. Therefore, we need to intersect the theme sentences to identify the common phrases and then generate a new sentence. Phrases produced by theme intersection will form the content of the generated summary.

Given the theme shown in Figure 2, how can we determine which phrases should be selected to form the summary content? For our example theme, the problem is to determine that only the phrase "*On Friday, U.S. F-16 fighter jet was shot down by a Bosnian Serb missile*" is common across all sentences.

The first sentence includes the clause; however, in other sentences, it appears in different paraphrased forms, such as "*A Bosnian Serb missile shot down a U.S. F-16 on Friday.*". Hence, we need to identify similarities between phrases that are not identical in wording, but do report the same fact. If paraphrasing rules are known, we can compare the predicate-argument structure of the sentences and find common parts. Finally, having selected the common parts, we must decide how to combine phrases, whether additional information is needed for clarification, and how to order the resulting sentences to form the summary.

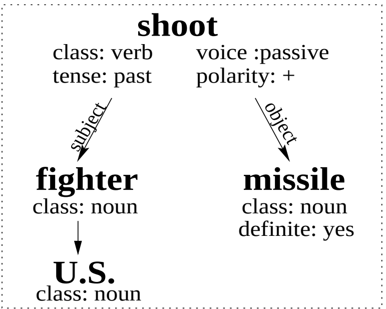

Figure 3: DSYNT of the sentence "*U.S. fighter was shot by missile.*"

3.1 An Algorithm for Theme Intersection

In order to identify theme intersections, sentences must be compared. To do this, we need a sentence representation that emphasizes sentence features that are relevant for comparison such as dependencies between sentence constituents, while ignoring irrelevant features such as constituent ordering. Since predicate-argument structure is a natural way to represent constituent dependencies, we chose a dependency based representation called *DSYNT* (Kittredge and Mel'čuk, 1983). An example of a sentence and its DSYNT tree is shown in Figure 3. Each non-auxiliary word in the sentence has a node in the DSYNT tree, and this node is connected to its direct dependents. Grammatical features of each word are also kept in the node. In order to facilitate comparison, words are kept in canonical form.

In order to construct a DSYNT we first run our sentences through Collin's robust, statistical parser (Collins, 1996). We developed a rule-based component that transforms the phrase-structure output of the parser to a DSYNT representation. Functional words (determiners and auxiliaries) are eliminated from the tree and the corresponding syntactic features are updated.

The comparison algorithm starts with all sentence trees rooted at verbs from the input DSYNT, and traverses them recursively: if two nodes are identical, they are added to the output tree, and their children are compared. Once a full phrase (a verb with at least two constituents) has been found, it is added to the intersection. If nodes are not identical, the algorithm tries to apply an appropriate paraphrasing rule from a set of rules described in the next section. For example, if the phrases "*group of students*" and "*students*" are compared, then the *omit empty head* rule is applicable, since "*group*" is an empty noun and can be dropped from the comparison, leaving two identical words, "*students*". If there is no applicable paraphrasing rule, then the comparison is finished and the intersection result is empty.

All the sentences in the theme are compared in pairs. Then, these intersections are sorted according to their frequencies and all intersections above a given threshold result in theme intersection.

For the theme in Figure 2, the intersection result is "*On Friday, a U.S. F-16 fighter jet was shot down by Bosnian Serb missile.*"[1]

3.2 Paraphrasing Rules Derived from Corpus Analysis

Identification of theme intersection requires collecting paraphrasing patterns which occur in our corpus. Paraphrasing is defined as alternative ways a human speaker can choose to "say the same thing" by using linguistic knowledge (as opposed to world knowledge) (Iordanskaja et al., 1991). Paraphrasing has been widely investigated in the generation community (Iordanskaja et al., 1991; Robin, 1994). (Dras, 1997) considered sets of paraphrases required for text transformation in order to meet external constraints such as length or readability. (Jacquemin et al., 1997) investigated morphology-based paraphrasing in the context of a term recognition task. However, there is no general algorithm capable of identifying a sentence as a paraphrase of another.

In our case, such a comparison is less difficult since theme sentences are *a priori* close semantically, which significantly constrains the kinds of paraphrasing we need to check. In order to verify this assumption, we analyzed paraphrasing patterns through themes of our training corpus derived from the Topic Detection and Tracking corpus (Allan et al., 1998). Overall, 200 pairs of sentences conveying the same information were analyzed. We found that 85% of the paraphrasing is achieved by syntactic and lexical transformations. Examples of paraphrasing that require world knowledge are presented below:

1. "*The Bosnian Serbs freed 121 U.N. soldiers last week at Zvornik*" and "*Bosnian Serb leaders freed about one-third of the U.N. personnel*"
2. "*Sheinbein showed no visible reaction to the ruling.*" and "*Samuel Sheinbein showed no reaction when Chief Justice Aharon Barak read the 3-2 decision*"

Since "surface" level paraphrasing comprises the vast majority of paraphrases in our corpus and is easier to identify than those requiring world-knowledge, we studied paraphrasing patterns in the corpus. We found the following most frequent paraphrasing categories:

1. ordering of sentence components: "*Tuesday they met...*" and "*They met ... tuesday*";
2. main clause vs. a relative clause: "*...a building was devastated by the bomb*" and "*...a building, devastated by the bomb*";
3. realization in different syntactic categories, *e.g.*, classifier vs. apposition: "*Palestinian leader Arafat*" and "*Arafat, palestinian leader*", "*Pentagon speaker*" and "*speaker from the Pentagon*";
4. change in grammatical features: active/passive, time, number. "*...a building was devastated by the bomb*" and "*...the bomb devastated a building*";
5. head omission: "*group of students*" and "*students*";
6. transformation from one part of speech to another: "*building devastation*" and "*...building was devastated*";
7. using semantically related words such as synonyms: "*return*" and "*alight*", "*regime*" and "*government*".

The patterns presented above cover 82% of the syntactic and lexical paraphrases (which is, in turn, 70% of all variants). These categories form the basis for paraphrasing rules used by our intersection algorithm.

The majority of these categories can be identified in an automatic way. However, some of the rules can only be approximated to a certain degree. For example, identification of similarity based on semantic relations between words depends on the coverage of the thesaurus. We

[1] To be exact, the result of the algorithm is a DSYNT that linearizes as this sentence.

identify word similarity using synonym relations from WordNet. Currently, paraphrasing using part of speech transformations is not supported by the system. All other paraphrase classes we identified are implemented in our algorithm for theme intersection.

3.3 Temporal Ordering

A property that is unique to multi-document summarization is the effect of time perspective (Radev and McKeown, 1998). When reading an original text, it is possible to retrieve the correct temporal sequence of events which is usually available explicitly. However, when we put pieces of text from different sources together, we must provide the correct time perspective to the reader, including the order of events, the temporal distance between events and correct temporal references.

In single-document summarization, one of the possible orderings of the extracted information is provided by the input document itself. However, in the case of multiple-document summarization, some events may not be described in the same article. Furthermore, the order between phrases can change significantly from one article to another. For example, in a set of articles about the Oklahoma bombing from our training set, information about the "*bombing*" itself, "*the death toll*" and "*the suspects*" appear in three different orders in the articles. This phenomenon can be explained by the fact that the order of the sentences is highly influenced by the focus of the article.

One possible discourse strategy for summaries is to base ordering of sentences on chronological order of events. To find the time an event occurred, we use the publication date of the phrase referring to the event. This gives us the best approximation to the order of events without carrying out a detailed interpretation of temporal references to events in the article, which are not always present. Typically, an event is first referred to on the day it occurred. Thus, for each phrase, we must find the earliest publication date in the theme, create a "time stamp", and order phrases in the summary according to this time stamp.

Temporal distance between events is an essential part of the summary. For example, in the summary in Figure 1 about a "*U.S. pilot downed in Bosnia*", the lengthy duration between "*the helicopter was shot down*" and "*the pilot was rescued*" is the main point of the story. We want to identify significant time gaps between events, and include them in the summary. To do so, we compare the time stamps of the themes, and when the difference between two subsequent time stamps exceeds a certain threshold (currently two days), the gap is recorded. A time marker will be added to the output summary for each gap, for example "*According to a Reuters report on the 10/21*".

Another time-related issue that we address is *normalization* of temporal references in the summary. If the word "*today*" is used twice in the summary, and each time it refers to a different date, then the resulting summary can be misleading. Time references such as "*today*" and "*Monday*" are clear in the context of a source article, but can be ambiguous when extracted from the article. This ambiguity can be corrected by substitution of this temporal reference with the full time/date reference, such as "*10/21*". By corpus analysis, we collected a set of patterns for identification of ambiguous dates. However, we currently don't handle temporal references requiring inference to resolve (e.g., "the day before the plane crashed," "around Christmas").

4 Sentence Generation

The input to the sentence generator is a set of phrases that are to be combined and realized as a sentence. Input features for each phrase are determined by the information recovered by shallow analysis during content planning. Because this input structure and the requirements on the generator are quite different from typical language generators, we had to address the design of the input language specification and its interaction with existing features in a new way, instead of using the existing SURGE syntactic realization in a "black box" manner.

As an example, consider the case of temporal modifiers. The DSYNT for an input phrase will simply note that it contains a prepositional phrase. FUF/SURGE, our language generator, requires that the input contain a semantic role, *circumstantial* which in turn contains a temporal feature.

The labelling of the circumstantial as *time* allows SURGE to make the following decisions given a sentence such as: "*After they made an emergency landing, the pilots were reported missing.*"

- The selection of the position of the *time* circumstantial in front of the clause
- The selection of the mood of the embedded clause as "finite".

The semantic input also provides a solid basis to authorize sophisticated revisions to a base input. If the sentence planner decides to adjoin a *source* to the clause, SURGE can decide to move the time circumstantial to the end of the clause, leading to: "*According to Reuters on Thursday night, the pilots were reported missing after making an emergency landing.*" Without such paraphrasing ability, which might be decided based on the semantic roles, *time* and *sources*, the system would have to generate an awkward sentence with both circumstantials appearing one after another at the front of the sentence.

While in the typical generation scenario above, the generator can make choices based on semantic information, in our situation, the generator has only a low-level syntactic structure, represented as a DSYNT. It would seem at first glance that realizing such an input should be easier for the syntactic realization component. The generator in that case is left with little less to do than just linearizing the input specification. The task we had to solve, however, is more difficult for two reasons:

1. The input specification we define must allow the sentence planner to perform revisions; that is, to attach new constituents (such as source) to a base input specification without taking into account all possible syntactic interactions between the new constituent and existing ones;
2. SURGE relies on semantic information to make decisions and verify that these decisions are compatible with the rest of the sentence structure. When the semantic information is not available, it is more difficult to predict that the decisions are compatible with the input provided in syntactic form.

We modified the input specification language for FUF/SURGE to account for these problems. We added features that indicate the ordering of circumstantials in the output. Ordering of circumstantials can easily be derived from their ordering in the input. Thus, we label circumstantials with the features *front-i* (*i*-th circumstantial at the front of the sentence) and *end-i* (*i*-th circumstantial at the end), where *i* indicates the relative ordering of the circumstantial within the clause.

In addition, if possible, when mapping input phrases to a SURGE syntactic input, the sentence planner tries to determine the semantic type of circumstantial by looking up the preposition (for example: "*after*" indicates a "*time*" circumstantial). This allows FUF/SURGE to map the syntactic category of the circumstantial to the semantic and syntactic features expected by SURGE. However, in cases where the preposition is ambiguous (e.g., "*in*" can indicate "*time*" or "*location*") the generator must rely solely on ordering circumstantials based on ordering found in the input.

We have modified SURGE to accept this type of input: in all places SURGE checks the semantic type of the circumstantial before making choices, we verified that the absence of the corresponding input feature would not lead to an inappropriate default being selected. In summary, this new application for syntactic realization highlights the need for supporting hybrid inputs of variable abstraction levels. The implementation benefited from the bidirectional nature of FUF unification in the handling of hybrid constraints and required little change to the existing SURGE grammar. While we used circumstantials to illustrate the issues, we also handled revision for a variety of other categories in the same manner.

5 Evaluation

Evaluation of multi-document summarization is difficult. First, we have not yet found an existing collection of human written summaries of multiple documents which could serve as a gold standard. We have begun a joint project with the Columbia Journalism School which will provide such data in the future. Second, methods used for evaluation of extraction-based systems are not applicable for a system which involves text regeneration. Finally, the manual effort needed to develop test beds and to judge system output is far more extensive than for single document summarization; consider that a human judge would have to read many input articles (our largest test set contained 27 input articles) to rate the validity of a summary.

Consequently, the evaluation that we performed to date is limited. We performed a quantitative evaluation of our content-selection component. In order to prevent noisy input from the theme construction component from skewing the evaluation, we manually constructed 26 themes, each containing 4 sentences on average. Far more training data is needed to tune the generation portion. While we have tuned the system to perform with minor errors on the manual set of themes we have created (the missing article in the fourth sentence of the summary in Figure 1 is an example), we need more robust input data from the theme construction component, which is still under development, to train the generator before beginning large scale testing. One problem in improving output is determining how to recover from errors in tools used in early stages of the process, such as the tagger and the parser.

5.1 Intersection Component

The evaluation task for the content selection stage is to measure how well we identify common phrases throughout multiple sentences. Our algorithm was compared against intersections extracted by human judges from each theme, producing 39 sentence-level predicate-argument structures. Our intersection algorithm identified 29 (74%) predicate-argument structures and was able to identify correctly 69% of the subjects, 74% of the main verbs, and 65% of the other constituents in our list of model predicate-argument structures. We present system accuracy separately for each category, since identifying a verb or a subject is, in most cases, more important than identifying other sentence constituents.

6 Conclusions and Future Work

In this paper, we presented an implemented algorithm for multi-document summarization which moves beyond the sentence extraction paradigm. Assuming a set of similar sentences as input extracted from multiple documents on the same event (McKeown et al., 1999; Eskin et al., 1999), our system identifies common phrases across sentences and uses language generation to reformulate them as a coherent summary. The use of generation to merge similar information is a new approach that significantly improves the quality of the resulting summaries, reducing repetition and increasing fluency.

The system we have developed serves as a point of departure for research in a variety of directions. First is the need to use learning techniques to identify paraphrasing patterns in corpus data. As a first pass, we found paraphrasing rules manually. This initial set might allow us to automatically identify more rules and increase the performance of our comparison algorithm.

From the generation side, our main goal is to make the generated summary more concise, primarily by combining clauses together. We will be investigating what factors influence the combination process and how they can be computed from input articles. Part of combination will involve increasing coherence of the generated text through the use of connectives, anaphora or lexical relations (Jing, 1999).

One interesting problem for future work is the question of how much context to include from a sentence from which an intersected phrase is drawn. Currently, we include no context, but in some cases context is crucial even though it is not a part of the intersection. This is the case, for example, when the context negates, or denies, the embedded sub-clause which matches a sub-clause in another negating context. In such cases, the resulting summary is actually false. This occurs just once in our test cases, but it is a serious error. Our work will characterize the types of contextual information that should be retained and will develop algorithms for the case of negation, among others.

Acknowledgments

We would like to thank Yael Dahan-Netzer for her help with SURGE. This material is based upon work supported by the National Science Foundation under grant No. IRI-96-1879. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

James Allan, Jaime Carbonell, George Doddington, Jon Yamron, and Y. Yang. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the Broadcast News Understanding and Transcription Workshop*, pages 194–218.

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

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

Michael Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, California.

Mark Dras. 1997. Reluctant paraphrase: Textual restructuring under an optimisation model. In *Proceedings of PACLING97*, pages 98–104, Ohme, Japan.

Michael Elhadad. 1993. *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation*. Ph.D. thesis, Department of Computer Science, Columbia University, New York.

Eleazar Eskin, Judith Klavans, and Vasileios Hatzivassiloglou. 1999. Detecting similarity by apllying learning over indicators. submitted.

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

Cristian Jacquemin, Judith L. Klavans, and Evelyne Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *proceedings of the 35th Annual Meeting of the ACL*, pages 24–31, Madrid, Spain, July. ACL.

Hongyan Jing. 1999. Summary generation through intelligent cutting and pasting of the input document. PhD thesis proposal.

Richard Kittredge and Igor A. Mel'čuk. 1983. Towards a computable model of meaning-text relations within a natural sublanguage. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pages 657–659, Karlsruhe, West Germany, August.

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

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

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

Kathleen R McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multi-document summarization by reformulation: Progress and prospects. submitted.

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

Jacques Robin. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background: Corpus-Based Analysis, Design, Implementation, and Evaluation*. Ph.D. thesis, Department of Computer Science, Columbia University, NY.

Gerald Salton, James Allan, Chris Buckley, and Amit Singhal. 1991. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264:1421–1426, June.