In *Advances in Automatic Text Summarization*, I. Mani and M. Maybury (editors), 1999.

Automated Text Summarization in SUMMARIST

Eduard Hovy and Chin-Yew Lin

Information Sciences Institute
of the University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292-6695
U.S.A.
Tel: +1-310-822-1511
Fax: +1-310-823-6714
Email: {hovy,cyl}@isi.edu

Abstract

SUMMARIST is an attempt to create a robust automated text summarization system, based on the 'equation': *summarization = topic identification + interpretation + generation*. Each of these stages contains several independent modules, many of them trained on large corpora of text. We describe the system's architecture and provide details of some of its modules.

1. Introduction

1.1 Extract, Abstract, or Something Else?

The task of a text summarizer is to produce a synopsis of any document (or set of documents) submitted to it. The level of sophistication of a synopsis can vary from a simple list of isolated keywords that indicate the major content of the document(s), through a list of independent single sentences that together express the major content, to a coherent, fully planned and generated text that compresses the document(s). The more sophisticated a synopsis, the more effort it generally takes to produce.

Several existing systems, including some Web browsers, claim to perform summarization. However, a cursory analysis of their output shows that their summaries are simply portions of the text, produced verbatim. While there is nothing wrong with such *extracts*, per se, the word 'summary' usually connotes something more, involving the fusion of various concepts of the text into a smaller number of concepts, to form an *abstract*. We define extracts as consisting wholly of portions extracted verbatim from the original (they may be single words or whole passages) and abstracts as consisting of novel phrasings describing the content of the original (which might be paraphrases or fully synthesized text). Generally, producing an abstract requires stages of topic fusion and text generation not needed for extracts.

In addition to extracts and abstracts, summaries may differ in several other ways. Some of the major types of summary that have been identified include indicative (keywords indicating topics) vs. informative (content-laden); generic (author's perspective) vs. query-oriented (user-specific); background vs. just-the-news; single-document vs. multi-document; neutral vs. evaluative. A full understanding of the major dimensions of variation, and the types of reasoning required to produce each of them, is still a matter of investigation. This makes the study of automated text summarization an exciting area in which to work.

1.2 SUMMARIST

Over the past two years we have been developing the text summarization system SUMMARIST. Our goal is to investigate the nature of text summarization, using SUMMARIST both as a research tool and as an engine to produce summaries for people upon demand. In order to maintain functionality while we experiment with new aspects, and since not all kinds of summary require the same processing steps, we have adopted a very open, modular design.

In this paper, we describe the architecture of SUMMARIST and provide details on the evaluated results of two of its component modules. Since it is still under development, not all the modules of SUMMARIST are at the same level of completeness. We describe the states of various modules in Sections 3.2, 3.3, and 3.4.

The goal of SUMMARIST is to provide both extracts and abstracts for arbitrary English and other-language text. SUMMARIST combines robust NLP processing (using IR and statistical techniques) with symbolic world knowledge (embodied in the concept thesaurus WordNet, dictionaries, and similar resources) to overcome the problems endemic to either approach alone. These problems arise because existing robust NLP methods tend to operate at the word level, and hence miss concept-level generalizations (which are provided by symbolic world knowledge), while on the other hand symbolic knowledge is too difficult to acquire in large enough scale to provide adequate coverage and robustness. For high-quality yet robust summarization, both aspects are needed.

To produce abstract-type summaries, the core process is a step of interpretation. In this step, two or more topics are fused together to form a third, more general, one. (We

define *topic* as a particular subject that we write about or discuss.). This step must occur in the middle of the summarization procedure: First, an initial stage of topic identification and extraction is required to find the central topics in the input text; finally, to produce the summary, a concluding stage of sentence generation is needed. Thus SUMMARIST is based on the following 'equation':

summarization = topic identification + interpretation + generation

This breakdown is motivated as follows:

1. Identification: The goal is to filter the input to retain only the most important, central, topics. For generality we assume that a text can have many (sub)-topics, and that the topic extraction process can be parameterized in at least two ways: first, to include more or fewer topics to produce longer or shorter summaries, and second, to include only topics relating to the user's expressed interests. Typically, topic identification can be achieved using various complementary techniques, including those based on stereotypical text structure, cue words, high-frequency indicator phrases, and discourse structure. We describe these in Section 3.2.

2. Interpretation: Once the desired central topics have been identified, they can simply be output, to form an extract. In human summaries, however, a process of interpretation is usually performed to achieve further compaction. In one study, (Marcu 98) counted how many clauses had to be extracted from a text in order to fully contain all the material included in a human abstract of that text. Working with a newspaper corpus of 10 texts and 14 judges, he found a compression factor of 2.76—in this genre, extracts are almost three times as long (counting words) as their corresponding abstracts! Results of this kind indicate the need for summarization systems to further process extracted material: to remove redundancies, rephrase sentences to pack material more densely, and, importantly, to merge or fuse related topics into more 'general' ones. The various types of fusion are not yet known, but they include at least simple concept generalization (*he ate pears, apples, and bananas → he ate fruit*) and script identification (*he sat down, read the menu, ordered, ate, and left → he visited the restaurant*). See Section 3.3.

3. Generation: The goal is to reformulate the extracted and fused material into a coherent, densely phrased, new text. If this stage is skipped, the output is a verbatim quotation of some portion(s) of the input, and is not likely to be high-quality text (although this might be sufficient for the application). The modules implemented or planned for SUMMARIST are described in Section 3.4.

2. Related Work: A Summary of Methods

2.1 Older Approaches

Automated summarization is not a new idea. However, the techniques tried during the 1950's and 60's were characterized by their simplicity of processing, since at that time neither large corpora of text, nor sophisticated NLP modules, nor powerful computers with large memory existed. Pioneering work (Luhn 59; Edmundson 68) studied the following techniques:

- *Position in the text:* Sentences in privileged locations (first paragraph, or immediately following section headings "Introduction", "Purpose", "Conclusions", etc.) contain the topic(s).
- *Lexical cues:* The presence of words such as *significant, hardly, impossible* signals topic sentences.
- *Location:* First (and last) sentences of each paragraph contain topic information.

Although each of these approaches has some utility, they depend very much on the particular format and style of writing. The strategy of taking the first paragraph, for example, works only in the newspaper and news magazine genres, and not always then either. No automatic techniques were developed for determining optimal positions, relevant cues, etc.

2.2 Traditional Semantic NLP Approaches

Compared to the complex processing people perform when summarizing (see for example (Endres-Niggemeyer 97)), automated summarization techniques are likely to remain mere approximations for a long time yet. True summarizing requires the understanding and interpretation of the text into a new synthesis, at different levels of abstraction. Semantics-based Artificial Intelligence (AI) techniques developed in the 1970's and early 80's promised to provide the necessary reasoning capabilities.

Lehnert's work on Plot Units (Lehnert 83) is an interesting historical example. Plot Units represent high-level interpersonal interactions such as *denied-request, give-up, success-born-of-adversity*. By representing the series of interactions of protagonists in a story as a connected network of Plot Units, and by simply counting the number of interconnections from each Plot Unit to its neighbors, Lehnert could capture the centrality of each action to the story. She was able to generate a summary of stories represented as chains of Plot Units to any level of detail, simply by leaving out more or fewer of the peripheral Units. Unfortunately, Lehnert did not succeed in developing a parser powerful enough to parse stories into Plot Units in more than a toy domain.

Plot Units are a rather abstract representation scheme. More recent approaches instead use frames or templates that house the most pertinent aspects of stereotypical situations and objects (Mauldin 91; Rau 91). As outlined in (McKeown and Radev 95), such templates form an obvious basis from which to generate summaries. Once you know what kind of information you want in a summary, you can specify a template for it, and then you simply need a powerful enough parser/analyzer to identify and extract the appropriate pieces of information from the text.

The recent TIPSTER funding program in the USA has supported the development of analyzers that perform information extraction from real-world newspaper texts in circumscribed topic domains such as terrorism. Using a variety of methods, these systems pinpoint and extract the types of information that have been prespecified to be interesting. TIPSTER/MUC systems such as FASTUS (Hobbs 92), GE-CMU (Jacobs 90), CIRCUS (Lehnert 91), and others are great achievements.

If the goal is to provide a detailed analysis, according to a predefined template, of the content of a text in a circumscribed but still fairly large domain, then systems of this ilk are the best available in the world today. But if one wants a system that can reflect what appears in the text and not just what the analyst has predefined to be of interest, then this approach is not adequate. A fixed-output template system is by its definition limited to the contents of the template, and it can never exceed this boundary. One is forced to turn to less semantic, more robust techniques.

2.3 IR Approaches

One place to turn for robust text processing techniques is Information Retrieval (IR). Active since the 1950's, IR researchers have spent a great deal of effort in developing methods of locating texts based on their characteristics, categorizing texts into predefined classes, and searching for incisive characterizations of the contents of texts (Salton 88; Rijsbergen 79; Paice 90).

Scaling down one's perspective from a large text collection to a single text (i.e., a collection of words and phrases), topic identification for extracts can be seen as a localized IR task. Can the IR techniques that pinpoint the significant passages in a collection of texts operate successfully when working on a single text? The question is still open, though recent research, and a majority of systems (see the other chapters of this book), seem to indicate that they can, at least to some extent.

The pure IR approach does have limitations, however. IR researchers have tended to eschew symbolic representations; anything deeper than the word level has often been viewed with suspicion. This attitude is both the strength and the weakness of IR. It is a strength, because it frees IR researchers from the seductive call of some magical powerful internal representation that will solve all the problems easily; it is a weakness, because it prevents researchers from employing reasoning at the non-word level. Unfortunately, abstract-type summaries require analysis and interpretation at levels deeper than the word level. This is mostly due to step 2 of the 'equation' above: without topic reinterpretation / fusion these systems can do no more than word counting and word recombination. Unless they have recourse to significant, large repositories of world knowledge, word-level systems can never know that the sequence *enter + order + wait + eat + pay + leave* can be summarized as *restaurant-visit.*

Although word-level techniques have been well developed and applied in many practical cases, they have been criticized in several respects (Mauldin 91; Riloff 94; Hull 94) for the following reasons:

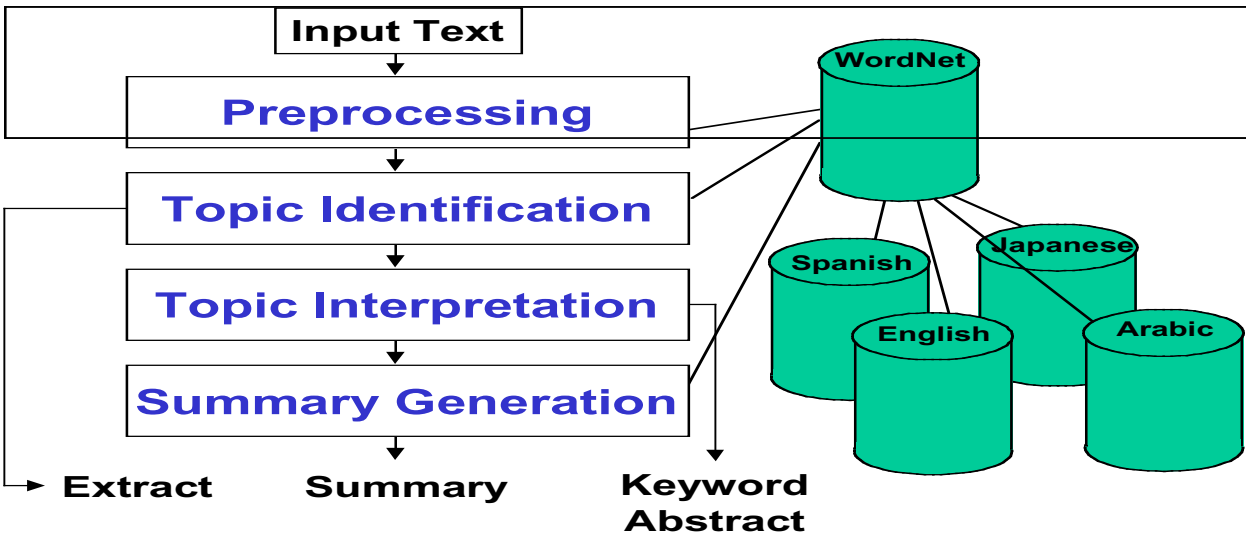
- *Synonymy:* One concept can be expressed by different words. For example, *cycle* and *bicycle* can both refer to some kind of vehicle (Hudson 95).
- *Polysemy:* One word can have several meanings. For example, *cycle* could mean *life cycle* or *bicycle.*
- *Phrases:* A phrase may have meaning different from the words in it. For example, an *alleged murderer* is not a *murderer*.
- *Term Dependency:* Terms are not totally independent of each other (as with synonymy).

Synonymy, polysemy, phrases, and term dependency problems all relate to semantics. A natural question is: why not use existing, pre-compiled, semantic knowledge sources as approximations to world knowledge, to help perform the interpretation step? Online dictionaries, thesauri, and wordlists are increasingly available. Using a thesaurus, one can identify synonyms. Using a sense disambiguation algorithm (e.g., Yarowsky 92), one can select the correct sense of a polysemous word. Using a syntactic parser, one can extract phrase segments and use them as terms (Lewis 92). Latent semantic indexing (Deerwester et al. 90; Hull 94) has been used to remedy the term dependency problem. All these efforts are attempts to bridge the gap between word form and word meaning. Following this trend, there is increasing interest in integrating shallow semantic processing and word-based statistical techniques to improve the performance of automatic text categorization systems (Liddy 94; Riloff 94).

Our approach with SUMMARIST is to employ IR techniques as far as they can take us, and then to augment them with symbolic/semantic and statistical methods. For example, SUMMARIST performs not only word counting (an IR technique for determining central topics), but also *concept counting* (using WordNet and similar resources) so that it can operate on a level 'deeper' than surface lexis. At this time, the interpretation stage in SUMMARIST is still rudimentary; most attention has been placed on the development of the topic identification modules. It therefore currently resembles other IR-based summarization systems such as DimSum (Aone et al. 97). Later, SUMMARIST will perform topic fusion at this deeper, non-lexical level, a step impossible to perform with pure IR techniques. SUMMARIST embodies one variant of knowledge-rich, assisted summarization, in which the requisite topic fusion/interpretation knowledge is acquired by statistical NLP with the help of online semantic and lexical resources.

3. The Structure of SUMMARIST

For each of the three steps of the above 'equation', SUMMARIST uses a mixture of symbolic world knowledge (from WordNet and similar resources) and statistical or IR-based techniques. Each stage employs



several different, complementary, methods. To date, we have developed some methods for each stage of processing, and are busy developing additional methods and linking them into a single system. In the next sections we describe some methods from each stage. The overall architecture is shown in Figure 1.

Figure 1. Architecture of SUMMARIST

Within each stage, each module is designed to operate independently (though some modules use the results of other modules). During the preprocessing stage, each word of the input document is written on a separate line of the processing file. Every module inspects the file and adds information to the relevant word(s), usually in the form of some numerical value or rating. At any time during processing, one can inspect the file and see which modules have run and how they have rated and/or augmented each portion of the input text. At the end of each stage, an integrator module combines the scores using a combination function and adds the resulting overall scores and/or values into the processing file. An example of SUMMARIST's extract production is provided in Section 4.

3.1 Preprocessing

The system's architecture most easily supports extensions with new modules when all input texts are converted into a standardized internal format. Before summarization starts, several preprocessing modules are activated. Each module either performs certain preprocessing tasks (such as tokenization) or attaches additional features (such as part-of-speech tags) to the input texts. These modules are:

- **tokenizer:** reads English texts and outputs tokenized texts.
- **part-of-speech tagger:** reads tokenized texts and outputs part-of-speech tagged texts. This tagger is based on Brill's part-of-speech tagger (Brill 93).
- **converter:** converts tagged texts into SUMMARIST internal representation.
- **morpher:** finds all root forms of each input token, using a modification of WordNet's (Miller et al. 90) demorphing program.
- **phraser:** finds collocations (multi-word phrases), as recorded in WordNet.
- **token frequency counter:** counts the occurrence of each token in an input text.
- **tf.idf weight calculator:** calculates the *tf.idf* weight (Salton 88) for each input token, and ranks the tokens according to this weight.
- **query relevance calculator:** to produce query-sensitive summaries, this module records with each sentence the number of (demorphed) content words in the user's query that also appear in that sentence.

The results of these modules are shown in Section 4.

3.2 Topic Identification

Several techniques for topic identification have been reported in the literature, including methods based on Position (Luhn 58, Edmundson 69), Cue Phrases (Baxendale 58), word frequency, and Discourse Segmentation (Marcu 97). SUMMARIST will eventually contain modules that employ each of these methods. To date, modules for position, various types of word frequency, and cue phrases have been implemented, and a module based discourse structure is under construction. When each module has rated each sentence, a combination function implemented in the Topic Id Integration Module combines their scores to produce the overall ranking. The Topic Identification stage then returns the top-ranked *n%* of sentences as its final result.

3.2.1 Position Module

As first described by (Luhn 59; Edmundson 68), this method exploits the fact that in some genres, regularities of discourse structure and/or methods of exposition mean

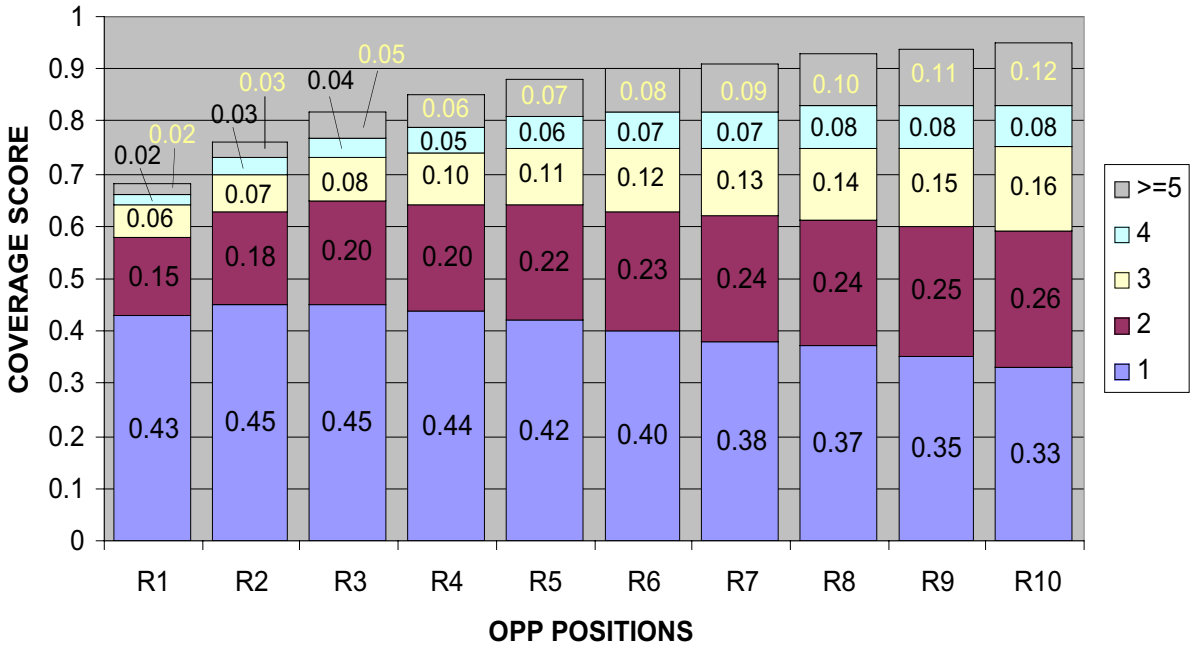


Figure 2. Coverage scores for top ten OPP sentence positions, window sizes 1 to 5.

that certain sentence positions tend to carry important topics. In (Lin and Hovy 97), we generalize their results with a method for automatically identifying the sentence positions most likely to yield good summary sentences. We define the *Optimal Position Policy* (OPP) as a list that indicates in what ordinal positions in the text high-topic-bearing sentences occur. We developed scoring metrics and normalization techniques for the automatic training of new OPPs, given a collection of genre-related texts with abstracts. To our knowledge, this work is the first systematic study and evaluation of the Position method.

For the Ziff-Davis corpus (13,000 newspaper articles announcing computer products) we have found that the OPP is

[T1, P2S1, P3S1, P4S1, P1S1, P2S2, {P3S2, P4S2, P5S1, P1S2}, P6S1, ...]

i.e., the title (T1) is the most likely to bear topics, followed by the first sentence of paragraph 2, the first sentence of paragraph 3, etc. (Paragraph 1 is invariably a teaser sentence in this corpus.) In contrast, for the *Wall Street Journal*, the OPP is

[T1, P1S1, P1S2, ...]

Evaluation: We evaluated the OPP method in various ways. In one of them, *coverage* is the fraction of the (human-supplied) keywords that are included verbatim in the sentences selected under the policy. (A random selection policy would extract sentences with a random distribution of topics; a good position policy would extract rich topic-bearing sentences.) We measured the effectiveness of an OPP by taking cumulatively more of its sentences: first just the title, then the title plus P2S1, and so on. In order to determine the effect of multi-word key phrases, we matched using windows of increasing size, from 1 word to 5 words. The resulting coverage scores are shown in Figure 2, by window size. Summing together the multi-word contributions (window sizes 1 to 5) in the top ten sentence positions (R10), the columns reach 95% over an extract of 10 sentences (approx. 15% of a typical Ziff-Davis text): an encouraging result

3.2.2 Cue Phrases

Phrases such as “in summary”, “in conclusion”, and superlatives such as “the best”, “the most important” can be good indicators of important content (Edmundson 69). Cue phrases are usually genre dependent. For example, “Abstract” and “in conclusion” are more likely to occur in scientific literature than in newspaper articles.

In one experiment, we manually compiled a list of cue phrases from a training corpus of paragraphs that themselves were summaries of texts. In this corpus, sentences containing phrases such as “this paper”, “this article”, “this document”, and “we conclude” fairly reliably reflected the major content of the paragraphs. This indicated to us the possibility of summarizing a summary. Figure 3 contains an example, with sentences containing cue phrases underlined.

During processing, the Cue Phrase Module recognizes the occurrence of cue phrases and ‘rewards’ each word in the sentence containing the phrase with an appropriate score

(constant per cue phrase).

In another experiment, we examined methods to automatically generate cue phrases (Liu and Hovy 98). By comparing the ratios (w_s/w_t) of occurrences of words in

Projections of levels of radioactive fallout from a nuclear war are sensitive to assumptions about the structure of the nuclear stockpiles as well as the assumed scenarios for a nuclear war. Recent arms control proposals would change these parameters. This paper examines the implications of the proposed (Intermediate-range Nuclear Forces) INF treaty and (Strategic Arms Reduction Treaty) START on fallout projections from a major nuclear war. We conclude that the INF reductions are likely to have negligible effects on estimates of global and local fallout, whereas the START reductions could result in reductions in estimates of local fallout that range from significant to dramatic, depending upon the nature of the reduced strategic forces. Should a major war occur, projections of total fatalities from direct effects of blast, thermal radiation, and fallout, and the phenomenon known as nuclear winter, would not be significantly affected by INF and START initiatives as now drafted. 14 refs.

Figure 3. Highlighted summary, generated by SUMMARIST using cue phrases.

summaries (w_s) and in the corresponding texts (w_t), we extracted the words showing the highest increase in occurrence density between text and associated abstract. We then searched for frequent concatenations of such privileged words into phrases. While we found no useful phrases in a corpus of 1,000 newspaper articles, we found the following in 87 articles on Computational Linguistics:

w_s/w_t	*phrase*
11.500	multilingual natural language
8.500	paper presents the
7.500	paper gives
6.000	paper presents
6.000	now present
5.199	this paper presents
4.555	paper describes
2.510	this paper

A more comprehensive study of the automated gathering of cue phrases is reported in (Teufel and Moens 97).

3.2.3 Concept Signatures for Topic Identification

A Concept Signature is a topic word (the *head*) together with a list of associated (*keyword weight*) pairs. Concept signatures represent concepts using word co-occurrence patterns. The idea is based on a simple observation: when some concept plays an important role in a text, a related set of words occurs fairly predictably. This idea is used in IR systems to achieve query term expansion.

We describe in Section 3.3.2 how we automatically build Concept Signatures and plan to use them for topic interpretation. It is, however, also possible to use concept signatures for topic identification. In one experiment, we created a Signature for each of five groups of 200 documents, drawn from five domains. When performing topic identification for a document, the Topic Id Signature Module assigned to each occurrence of a Signature keyword that keyword's weight. Each sentence received a Signature score equal to the total of all Signature words contained in it, normalized by its length. This score indicates the relevance of the sentence to the signature topic. Examples appear in Section 4.

3.2.4 Topic Identification Integration Module

Each separate topic identification module assigns its score to each sentence. How should the various scores be combined for the best result?

Various approaches have been tried. Most of them employ some sort of combination function, in which coefficients assign various weights to the individual scores, which are then summed. (Kupiec et al. 95) and (Aone et al. 97) employ the Expectation Maximization algorithm to derive coefficients for their systems.

Initially, SUMMARIST contained a linear combination function, in which the coefficients were determined by manual experimentation. The results of this function were not optimal, as found in the formal TIPSTER/SUMMAC evaluation of various summarization systems (Firmin Hand and Sundheim 98).

In subsequent work, we tested two automated methods of creating a better combination function. First, we used the C4.5 (Quinlan 86) to build a decision tree automatically in the standard manner. As training data, we used a portion of the results of the TIPSTER/SUMMAC summarization evaluation dry run, annotated to indicate the relevance and popularity of each sentence (Baldwin 98). The algorithm generated a tree of 1,611 nodes, of which the top (most informative) questions pertain to the query signature, term frequency, overlap with title, and OPP. Compared with the manually built function, the decision tree is considerably better. SUMMARIST now scores 58.07% (Recall and Precision) on an unseen test set of 82 dry-run texts in a 5-way cross-validation run. (On the same data, the system used to score 33.02%.) We also implemented a 6-node perceptron. Training it on the same data produced results within 1% of the decision tree.

3.3 Topic Interpretation (Concept Fusion)

The second step in the summarization process is that of topic interpretation. In this step, two or more extracted topics are 'fused' into one (or more) unifying concept(s). Topic fusion can be as simple as part-whole construction, as when *wheel, chain, pedal, saddle, light, frame,* and *handlebars* fuse together to *bicycle*. Generally, though, it is more complex, ranging from direct concept/word clustering as used in IR for query expansion (Paice 90) to script-based inference such as *drive in + pay + fill tank + close tank → gasoline fill up* (Schank and Abelson 77).

Fusing topics into one or more characterizing concepts is the most difficult step of automated text summarization. The reason is that fusion requires knowledge about the world that is seldom included in the text explicitly. Consider the following example:

John and Bill wanted money. They bought ski-masks and

guns and stole an old car from a neighbor. Wearing their ski-masks and waving their guns, the two entered the bank, and within minutes left the bank with several bags of $100 bills. They drove away happy, throwing away the ski-masks and guns in a trash can. They were never caught.

Word counting would indicate that the story is about ski-masks and guns, both of which are mentioned more than any other content word. Clearly, however, the story is about a robbery, and any summary of it must mention this fact. Some process of interpreting the individual words as part of some encompassing concept is required.

A variety of methods can be employed. All of them associate a set of concepts (the *indicators*) with a characteristic generalization (the *fuser* or *head*). The challenge is to develop methods that work reliably and to construct a large enough collection of indicator-fuser sets to achieve effective topic reduction.

SUMMARIST's topic interpretation methods currently include *Concept Wavefront* (Lin 95) and *Concept Signature* (Lin and Hovy 98).

3.3.1 Concept Counting and the Wavefront

To identify the topics of texts, IR researchers make the assumption that the more a word is used in a text, the more important it is in that text. But although word frequency counting operates robustly across different domains without relying on stereotypical text structure or semantic models, they cannot handle synonyms, pronominalization, or other forms of coreferentiality, and they miss conceptual generalizations:

John bought some vegetables, fruit, bread, and milk.→ John bought some groceries.

Using a concept generalization taxonomy, we have developed a method to recognize that *vegetables*, *fruit*, etc., can be summarized as *groceries*. We count *concepts* instead of words, and generalize them using WordNet (Miller et al. 90) (though we could have used any machine-readable thesaurus) for inter-concept links. In the limit case, when WordNet does not contain concepts for the words in the text, this technique defaults to word counting.

The idea is simple. We count the number of occurrences of each content word in the text, and assign that number to the word's associated concept in WordNet. We then propagate all these weights upward, assigning to each node the sum of its weight plus all its childrens' weights. Next, we proceed back down, deciding at each node whether to stop or to continue downward. We stop when the node is an appropriate generalization of its children; that is, when its weight derives so equally from two or more of its children that no child is the clear majority contributor to its weight. This algorithm picks the most specific generalization of a set of concepts as their fuser.

In fact, one can find layers of fuser concepts. As described in (Lin 95), we locate the most appropriate generalizations by finding concepts on the *interesting wavefront*, a set of nodes representing concepts that each generalize a set of approximately equally strongly represented subconcepts (ones that have no obvious dominant subconcept to specialize to). To find the wavefront, we define a concept's *weight* to be the sum of the frequency of occurrence of the concept *C* plus the weights of all its subconcepts. We then define the *concept frequency ratio* between a concept and its subconcepts:

$$R = \frac{MAX(sum\,of\,all\,children\,of\,C)}{SUM(sum\,of\,all\,children\,of\,C)}$$

R is a way to identify the degree of summarization informativeness. We use this ratio to find interesting concepts in a hierarchical concept taxonomy. Starting from the top of the hierarchy, we proceed downward along each child branch whenever the branch ratio is greater than or equal to a cutoff value R_t, and stopping at that node otherwise. The resulting set of nodes we call the interesting wavefront. We can start another exploration of interesting concepts downward from this interesting wavefront, resulting in a second, lower, wavefront, and so on. By repeating this process until we reach the leaf concepts of the hierarchy, we can derive a set of interesting wavefronts. From the interesting wavefronts, we choose the most general one below a certain depth *D* to ensure a good balance of generality and specificity. For WordNet, we found *D=6*, by experimentation.

Evaluation: We selected 26 articles about new computer products from *BusinessWeek* (1993–94) of average 750 words each. For each text we extracted the eight sentences containing the most interesting concepts using the wavefront technique, and comparing them to the contents of a professional's abstracts of these 26 texts from an online service. We developed several weighting and scoring variations and tried various ratio and depth parameter settings for the algorithm. We also implemented a random sentence selection algorithm as a baseline comparison.

The average recall (*R*) and precision (*P*) values over the three scoring variations were *R=0.32* and *P=0.35*, when the system produces extracts of 8 sentences. In comparison, the random selection method had *R=0.18* and *P=0.22* precision in the same experimental setting. While these R and P values are not tremendous, they show that semantic knowledge—even as limited as that in WordNet—does enable improvements over traditional IR word-based techniques. However, the limitations of WordNet are serious drawbacks: there is no domain-specific knowledge, for example to relate *customer*, *waiter*, *cashier*, *food*, and *menu* together with *restaurant*. We thus developed a second technique of concept interpretation, using topic signatures.

3.3.2 Interpretation using Topic Signatures

Can one automatically find a set of related words that can collectively be fused into a single word appropriate for summarization? To test this idea we developed the Topic Signature method (Lin 97; Lin and Hovy 98). We define a signature to be a *head* together with a list of (*keyword score*) pairs, where each score provides the relative strength of association of its keyword.

To construct signatures automatically, we used a set of 30,000 texts from the 1987 *Wall Street Journal* (WSJ) corpus. The paper's editors have classified each text into one of 32 classes—AROspace, BNKing, ENVironment, TELecommunications, etc. We counted the occurrences of each content word (canonicalized morphologically to remove plurals, etc.), in the texts of each class, relative to the number of times they occur in the whole corpus, using the standard *tf.idf* method. We then selected the top-scoring 300 terms for each category and created a signature with the category name as its head. The top terms of four example signatures are shown in Figure 4. It is quite easy to determine the identity of the signature head just by inspecting the top few signature indicators.

RANK	ARO	BNK	ENV	TEL
1	contract	bank	epa	at&t
2	air_force	thrift	waste	network
3	aircraft	banking	environmental	fcc
4	navy	loan	water	cbs
5	army	mr.	ozone	cable
6	space	deposit	state	bell
7	missile	board	incinerator	long-distance
8	equipment	fslic	agency	telephone
9	mcdonnell	fed	clean	telecomm.
10	northrop	institution	landfill	mci
11	nasa	federal	hazardous	mr.
12	pentagon	fdic	acid_rain	doctrine
13	defense	volcker	standard	service
14	receive	henkel	federal	news
15	boeing	banker	lake	turner

Figure 4. Portions of the signatures of several concepts.

SUMMARIST will use signatures for summary creation as follows. After the topic identification stage identifies a set of important topics, the topic signature interpretation module will identify from its library of signatures the one(s) most fully subsuming the topic words, and these signatures' heads will then be used as the summarizing fuser concepts. Matching the identified topic terms against all signature indicators involves several problems, including taking into account the relative frequencies of occurrence and resolving matches with multiple signatures, and specifying thresholds of acceptability. Since this work has not yet been completed, the ultimate effectiveness of this method remains unknown.

Evaluation. We needed to evaluate the quality of the signatures formed by our algorithm. Recognizing the similarity of signature recognition to document categorization, we evaluated the effectiveness of each signature by seeing how well it served as a selection criterion on texts. As data we used a set of 2,204 previously unseen WSJ news articles from 1988.

For each test text, we created a single-text 'document signature' using the same *tf.idf* measure as before, and then matched this document signature against the category signatures. The closest match provided the class into which the text was categorized. We tested several matching functions, including a simple *binary* match (count 1 if a term match occurs; 0 otherwise); *curve-fit* match (minimize the difference in occurrence frequency of each term between document and concept signatures), and *cosine* match (minimize the cosine angle in the hyperspace formed when each signature is viewed as a vector and each word frequency specifies the distance along the dimension for that word). These matching functions all provided approximately the same results. The values for Recall and Precision (R=0.7566 and P=0.6931) are encouraging and compare well with recent IR results (TREC 95). Current experiments are investigating the use of contexts smaller than a full text to create more accurate signatures.

3.3.3 Clustering Tool

Extending the above-mentioned concept signature work will require the creation of signatures for hundreds, and eventually thousands, of different topics, as needed for robust summarization. An important step is obviously the grouping of documents about the same or similar topics before signatures can be trained. For this reason, we have implemented a variety of standard clustering techniques (CLINK, SLINK, Median, Ward's Method; see (Rasmussen 92)) in a Clustering Tool, and are adding more recent, faster, clustering methods based on sparse matrix reordering (Berry et al. 96). In addition, we have recently embarked on a large-scale signature building enterprise.

In order to build accurate signatures, we had to ensure that the document sets are pure; i.e., that they do not contain too much unrelated material. Therefore, we tested document set purification using clustering in the following experiment. We used a set of 1000 documents, pre-compiled from five domains, extracted from three sources (*Wall Street Journal*, *Associated Press*, and *Federal Register*) by the TIPSTER Summarization Evaluation committee (Firmin Hand 97). After mixing

the documents, we used the Clustering Tool to regenerate the five clusters. Precision of the results varied from over 99% to about 70%, depending on method, with CLINK (Defays 77) giving the best performance. Using the

Cluster 1		*Cluster 2*		*Cluster 3*	
bush	8.16	inmate	12.28	south	19.51
navy	7.00	prison	10.50	africa	9.69
defense	6.85	prisoner	4.08	congress	8.36
stealth	6.28	jail	3.67	black	7.39
house	5.67	county	3.63	african	7.14
bomber	5.11	sunday	2.89	white	4.60
program	4.96	correction	2.78	apartheid	3.62
missile	4.80	riot	2.35	sanction	3.16
billion	4.29	guard	2.27	de	2.98
plane	4.23	cell	2.26	anti-apartheid	2.97
air	3.95	court	2.16	bush	2.84
propose	3.93	department	2.13	leader	2.80
company	3.92	hold	2.01	political	2.65
fighter	3.70	federal	2.00	party	2.60
include	3.68	serve	1.98	government	2.52

Figure 5. Three concept signatures produced from automatically generated clusters.

above-mentioned signature training module, we then produced concept signatures for each cluster. The top-scoring elements of three of these signatures are shown in Figure 5. They seem to separate very clearly into distinct semantic domains.

3.4 Summary Generation

The final step in the summarization process is the generation of a summary. A range of possibilities occurs here, from simple word or phrase printing to sophisticated sentence planning and surface-form realization. SUMMARIST will eventually contain three generation modules, associated as appropriate with the various levels needed for various applications.

Extraction: For extract-type summaries, no generation is required. The terms or sentences selected by the Topic Identification stage can simply be reproduced. Despite the likely incoherence of the result, it may contain enough information to support humans performing tasks such as document routing. This output mode is currently used in SUMMARIST.

Topic lists: Sometimes no summary is really needed; a simple list of the summarizing topics is enough. SUMMARIST can simply print the extracted keywords or interpreted fuser concepts, sorted in decreasing importance.

Phrase concatenation: SUMMARIST will include a rudimentary generator that composes noun phrase-sized and clause-sized units into simple sentences. It will follow links from the fuser concepts through the words that support them in the input text, and from those sentences gather related noun phrases and clauses.

Full sentence planning and generation: SUMMARIST will employ the sentence planner being built at ISI in collaboration with the HealthDoc project from the University of Waterloo (Hovy and Wanner 96), together with a sentence generator, such as Penman (Penman 88, Matthiessen and Bateman 91), FUF (Elhadad 92), or NITROGEN (Knight and Hatzivassiloglou 95). Producing well-formed, fluent, summaries is not a trivial generation task, as shown by (McKeown and Radev 95): a considerable amount of planning is required to achieve dense packing of content. The input will be a list of the fuser concepts and their most closely related topics, as identified by SUMMARIST's topic identification stage.

4. An Example Extract Summary

Figure 6 shows a portion of SUMMARIST's internal preamble for text AP880212-0009. It includes a document number (*docno*); the title of the document (*title*); the modules that have processed the document (*module*); the words contained in the user's query (*query_signature*; the signature term | its weight. Terms are listed in descending weight order. The same format is used for *tf, tfidf, sig* and *opp* keywords); term frequency keywords (*tf_keywords*); *tf.idf* keywords (*tfidf_keywords*); the OPP rule used (*opp_rule*; p: the paragraph rule, with vertical bars separating paragraph number and rank. Title is indicated as paragraph 0); and OPP keywords (*opp_keywords*); the top three most similar topic signatures (*signature*, the first number is the topic/cluster number and the second one is the similarity of this signature to the document); signature keywords (*sig_keywords*). Note that keywords selected by term frequency, *tf.idf*, signature, and OPP are different. Figure 7 shows the content portion of the internal format of text AP880212-0009. Each line contains one word of the text followed by its attribute list. The attributes are

[*pno*, paragraph number], [*sno*, sentence number], [*pos*, part-of-speech tag], [*cwd*, common word (true or false)], [*ttl*, word appears in title (true or false)], [*mph*, root form], [*wnc*, the file identifer(s) of the WordNet file(s) containing this word], [*frq*, word frequency count], [*tfidf*, tf.idf weight], [*opp*, OPP weight (global, local)], [*sig*, signature weights of the top three most pertinent signatures], [*qry*, word in user query (true or false)]. Figure 9 shows the original full text of document AP880212-0009, and Figure 8 the generic summary of it.

<*docno=AP880212-0009>

<*title="90 Soldiers Arrested After Coup Attempt In Tribal Homeland. ">

<*module=PRE|TTL|MPH|CAT|TFIDF|OPP|SIG|QRY>

<*query_signature=black,13.406|blacks,13.406|africa,12.737|south,12.737|anc,5.855|challenge,5.855|domination,5.855|greatest,5.855|white,5.855|political,5.272|african,5.162|apartheid,5.162|majority,5.162|effort,5.118|efforts,5.118|minority,4.756|armed,4.469|congress,3.909|form,3.552|personnel,3.457|change,3.370|national,2.859|economic,2.764|military,2.720|interest,2.677|activities,2.559|activity,2.559|state,2.094|states,2.094|government,2.005>

<*tf_keywords=south,13.000|homeland,11.000|african,8.000|africa,7.000|bophuthatswana,7.000|mangope,7.000|soldier,7.000|coup,6.000|minister,5.000|arrest,4.000>

<*tfidf_keywords=bophuthatswana,43.453|mangope,43.453|homeland,37.446|coup,25.570|soldier,16.729|south,13.895|malebanemetsing,13.801|mmabatho,13.801|rathebe,13.801|african,12.622>

<*opp_rule=p:0,1|1,2|2,4|3,4 s:-,->

<*opp_keywords=homeland,42.667|south,37.917|coup,26.083|african,26.000|soldier,23.583|mangope,20.667|bophuthatswana,20.167|africa,19.667|attempt,14.333|tribal,14.333>

<*signature=110,0.269|138,0.184|151,0.137|->

<*sig_keywords=africa,9.074|anc,8.636|south,8.465|black,6.857|african,6.429|apartheid,3.390|botha,2.832|political,2.521|government,2.448|national,2.397|police,2.208|leaders,2.019|congress,1.981|party,1.779|organization,1.729|leader,1.706|group,1.552|pretoria,1.513|country,1.295|homeland,1.243|majority,1.243>

Figure 6. Preamble of text AP880212-0009.

About <pno=1 sno=1 pos=IN cwd=1 ttl=0 mph=- wnc=- frq=0 tfidf=0 opp=-,- sig=-,-|-,-|-,-|-,- qry=->

90 <pno=1 sno=1 pos=CD cwd=1 ttl=1 mph=- wnc=- frq=0 tfidf=0 opp=-,- sig=-,-|-,-|-,-|-,- qry=->

soldiers <pno=1 sno=1 pos=NNS cwd=0 ttl=1 mph=soldier wnc=18,5 frq=7 tfidf=16.729 opp=23.583,4.500 sig=-,-|53,1.530|-,-|-,- qry=->

have <pno=1 sno=1 pos=VBP cwd=1 ttl=0 mph=- wnc=- frq=0 tfidf=0 opp=-,- sig=-,-|-,-|-,-|-,- qry=->

been <pno=1 sno=1 pos=VBN cwd=1 ttl=0 mph=- wnc=- frq=0 tfidf=0 opp=-,- sig=-,-|-,-|-,-|-,- qry=->

arrested <pno=1 sno=1 pos=VBN cwd=0 ttl=1 mph=arrest wnc=35,33,38 frq=4 tfidf=9.064 opp=13.000,4.500 sig=166,0.555|-,-|121,0.564|-,- qry=->

and <pno=1 sno=1 pos=CC cwd=1 ttl=0 mph=- wnc=- frq=0 tfidf=0 opp=-,- sig=-,-|-,-|-,-|-,- qry=->

face <pno=1 sno=1 pos=VB cwd=0 ttl=0 mph=- wnc=32,33,42,38,35 frq=1 tfidf=1.783 opp=4.500,4.500 sig=-,-|-,-|-,-|-,- qry=->

possible <pno=1 sno=1 pos=JJ cwd=1 ttl=0 mph=- wnc=- frq=0 tfidf=0 opp=-,- sig=-,-|-,-|-,-|-,- qry=->

death <pno=1 sno=1 pos=NN cwd=0 ttl=0 mph=- wnc=11,19,28,18,26,4 frq=2 tfidf=3.554 opp=7.333,4.500 sig=273,0.426|64,1.350|149,0.497|-,- qry=->

...

Figure 7. Word-attribute list of text AP880212-0009, produced by SUMMARIST.

<DOC>

<DOCNO>AP880212-0009</DOCNO>

<TITLE>90 Soldiers Arrested After Coup Attempt In Tribal Homeland.</TITLE>

<TEXT>

About 90 soldiers have been arrested and face possible death sentences stemming from a coup attempt in Bophu-thatswana , leaders of the tribal homeland said Friday .

Rebel soldiers staged the takeover bid Wednesday , detaining homeland President Lucas Mangope and several top Cabinet officials for 15 hours before South African soldiers and police rushed to the homeland , rescuing the leaders and restoring them to power .

Bophuthatswana , which has a population of 1.7 million spread over seven separate land blocks , is one of 10 tribal homelands in South Africa .

Hennie Riekert , the homeland 's defense minister , said South African troops were to remain in Bophuthatswana but will not become a " permanent presence ."

</TEXT>

</DOC>

Figure 8. Generic summary produced by SUMMARIST.

<DOCNO> AP880212-0009 </DOCNO>

<HEAD>90 Soldiers Arrested After Coup Attempt In Tribal Homeland</HEAD>

<DATELINE>MMABATHO, South Africa (AP) </DATELINE>

<TEXT>

About 90 soldiers have been arrested and face possible death sentences stemming from a coup attempt in Bophuthatswana, leaders of the tribal homeland said Friday.

Rebel soldiers staged the takeover bid Wednesday, detaining homeland President Lucas Mangope and several top Cabinet officials for 15 hours before South African soldiers and police rushed to the homeland, rescuing the leaders and restoring them to power.

At least three soldiers and two civilians died in the uprising.

Bophuthatswana's Minister of Justice G. Godfrey Mothibe told a news conference that those arrested have been charged with high treason and if convicted could be sentenced to death. He said the accused were to appear in court Monday.

All those arrested in the coup attempt have been described as young troops, the most senior being a warrant officer.

During the coup rebel soldiers installed as head of state Rocky Malebane-Metsing, leader of the opposition Progressive Peoples Party. Malebane-Metsing escaped capture and his whereabouts remained unknown, officials said. Several unsubstantiated reports said he fled to nearby Botswana.

Warrant Officer M.T.F. Phiri, described by Mangope as one of the coup leaders, was arrested Friday in Mmabatho, capital of the nominally independent homeland, officials said.

Bophuthatswana, which has a population of 1.7 million spread over seven separate land blocks, is one of 10 tribal homelands in South Africa. About half of South Africa's 26 million blacks live in the homelands, none of which are recognized internationally.

Hennie Riekert, the homeland's defense minister, said South African troops were to remain in Bophuthatswana but will not become a "permanent presence."

Bophuthatswana's Foreign Minister Solomon Rathebe defended South Africa's intervention.

"The fact that ... the South African government (was invited) to assist in this drama is not anything new nor peculiar to Bophuthatswana," Rathebe said. "But why South Africa, one might ask? Because she is the only country with whom Bophuthatswana enjoys diplomatic relations and has formal agreements."

Mangope described the mutual defense treaty between the homeland and South Africa as "similar to the NATO agreement," referring to the Atlantic military alliance. He did not elaborate.

Asked about the causes of the coup, Mangope said, "We granted people freedom perhaps ... to the extent of planning a thing like this."

The uprising began around 2 a.m. Wednesday when rebel soldiers took Mangope and his top ministers from their homes to the national sports stadium.

On Wednesday evening, South African soldiers and police stormed the stadium, rescuing Mangope and his Cabinet.

South African President P.W. Botha and three of his Cabinet ministers flew to Mmabatho late Wednesday and met with Mangope, the homeland's only president since it was declared independent in 1977.

The South African government has said, without producing evidence, that the outlawed African National Congress may be linked to the coup.

The ANC, based in Lusaka, Zambia, dismissed the claims and said South Africa's actions showed that it maintains tight control over the homeland governments. The group seeks to topple the Pretoria government.

The African National Congress and other anti-government organizations consider the homelands part of an apartheid system designed to fragment the black majority and deny them political rights in South Africa.

</TEXT>

Figure 9. Full text AP890417-0617.

5. Conclusion

As outlined in Section 1, extract summaries require only the stage of topic identification. Accordingly, most of our early efforts have been devoted to the preprocessing and topic identification stages. At this time, we plan to include into the topic identification stage only one new

module. The Discourse Structure module, on which work is currently well underway (Marcu 98), will form the heart of the topic identification stage. By determining the importance of each clause within the overall discourse structure, this module will contribute directly to the scores of individual sentences. In addition, this module will also provide the interclause discourse relations (relations signaled by cue phrases such as "but", "although", "in order to") required by the generation stage to produce coherent text. Thus the current internal representation scheme of SUMMARIST, a linear sequence of sentences, will be changed into a discourse tree of the kind used in (Marcu 97).

We are beginning to address the subsequent stages of summarization as well. By including modules to perform topic interpretation and summary generation, SUMMARIST will also be able to produce abstract summaries. In order to perform concept interpretation, SUMMARIST requires a rather more elaborated concept taxonomy than it currently has. Work is underway to extend the SENSUS ontology (Knight and Luk 94; Hovy 98) by including other ontologies' contents and by parsing dictionary entries. In addition, in order to perform signature-based concept interpretation, SUMMARIST requires a large library of topic signatures. Work is also underway to acquire such a library using text searches on the web.

Automated summarization is simultaneously an old topic—work on it dates from the 1950's—and a new topic—it is so difficult that interesting headway can be made for many years to come. We are excited about the possibilities offered by the combination of semantic and statistical techniques in what is surely one of the most complex tasks of all natural language processing!

Acknowledgments

We thank Benjamin Liberman for his implementation of the Clustering Tool and Sara Shelton for her enthusiastic support, comments, and suggestions.

References

Aone, C., M.E. Okurowski, J. Gorlinsky, B. Larsen. 1997. A Scalable Summarization System using Robust NLP. *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, 66–73. ACL/EACL Conference, Madrid, Spain.

Baldwin, B. 1998. Reworking of TIPSTER/SUMMAC Dry Run Evaluation Results. University of Pennsylvania.

Baxendale, P.B. 1958. Machine-Made Index for Technical Literature—An Experiment. *IBM Journal* (October) 354–361.

Berry, M.W., B. Hendrickson, and P. Raghavan. 1996. Sparse Matrix Reordering Schemes for Browsing Hypertext. *Lectures in Applied Mathematics* 32: 99–123.

Brill, E. 1993. A Corpus-Based Approach to Language Learning. Ph.D. diss., University of Pennsylvania.

Deerwester, S., S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6): 391–407.

Defays, D. 1977. An Efficient Algorithm for a Complete Link Method. *Computer Journal* 20, 346–366.

Edmundson, H.P. 1968. New Methods in Automatic Extraction. *Journal of the ACM* 16(2), 264–285.

Elhadad, M. 1992. Using Argumentation to Control Lexical Choice: A Functional Unification-Based Approach. Ph.D. diss, Columbia University.

Endres-Niggemeyer, B. 1997. SimSum: Simulation of Summatizing. In *Proceedings of the Workshop on Intelligent Scalable Summarization* at the ACL/EACL Conference, 89–96. Madrid, Spain.

Firmin Hand, T. 1997. A Proposal for Task-Based Evaluation of Text Summarization Systems. In *Proceedings of the Workshop on Intelligent Scalable Summarization, a*t the ACL/EACL Conference, 31–38. Madrid, Spain.

Firmin Hand, T. and B. Sundheim. 1998. TIPSTER-SUMMAC Summarization Evaluation. *Proceedings of the TIPSTER Text Phase III Workshop*. Washington.

Hobbs, J.R., D. Appelt, M. Tyson, J. Baer, and D. Israel. 1992. Description of the FASTUS System used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 268–275. McLean, VA.

Hovy, E.H. and L. Wanner. 1996. Managing Sentence Planning Requirements. In *Proceedings of the Workshop on Gaps and Bridges in NL Planning and Generation*, 53–58. ECAI Conference. Budapest, Hungary.

Hovy, E.H. 1998. Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, 535–542. Granada, Spain.

Hudson, R. 1995. *Word Meaning*. London, England: Routledge.

Hull, D.A. 1994. Information Retrieval Using Statistical Classification. Ph.D. diss., Stanford University.

Jacobs, P.S. and L.F. Rau. 1990. SCISOR: Extracting Information from On-Line News. *Communications of the*

ACM 33(11): 88–97.

Knight, K. and S.K. Luk. 1994. Building a Large-Scale Knowledge Base for Machine Translation. *Proceedings of the Conference of the American Association of Artificial Intelligence* (AAAI-94), 773–778. Seattle, WA.

Knight, K. and V. Hatzivassiloglou. 1995. Two-Level Many-Paths Generation. In *Proceedings of the Thirty-third Conference of the Association of Computational Linguistics (ACL-95)*, 252–260. Boston, MA.

Kupiec, J., J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizer. In *Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR), 68–73. Seattle, WA.

Lehnert, W.G. 1983. Narrative complexity based on summarization algorithms. In *Proceedings of the Eighth International Joint Conference of Artificial Intelligence (IJCAI*-83), 713–716. Karlsruhe, Germany.

Lehnert, W.G. 1991. *Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds*, vol. 1, 135–164. Norwood, NJ: Ablex.

Lewis, D.D. 1992. An evaluation of phrasal and clustered representations of a text categorization task. In *Proceedings of the Fifteenth Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR), 37–50. New York, NY.

Liddy, E., W. Paik, and E.S. Yu. 1994. Text Categorization for Multiple Users Based on Semantic Features from a Machine-Readable Dictionary. *ACM Transactions on Information Systems*, vol. 12: 278–295.

Lin, C-Y. 1995. Topic Identification by Concept Generalization. In *Proceedings of the Thirty-third Conference of the Association of Computational Linguistics (ACL-95)*, 308–310. Boston, MA.

Lin, C-Y. 1997. Robust Automated Topic Identification. Ph.D. diss., University of Southern California.

Lin, C-Y. and E.H. Hovy. 1997. Identifying Topics by Position. In *Proceedings of the Applied Natural Language Processing Conference (ANLP-97)*, 283–290. Washington.

Lin, C-Y. and E.H. Hovy. 1998. Automatic Text Categorization: A Concept-Based Approach. In prep.

Liu, H. and E.H. Hovy. 1998. Automated Learning of Cue Phrases for Text Summarization. In prep.

Luhn, H.P. 1959. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 159–165.

Marcu, D. 1997. The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Ph.D. diss. University of Toronto.

Marcu, D. 1998. *The Automatic Construction of Large-scale Corpora for Summarization Research*. Forthcoming.

Matthiessen, C.M.I.M. and J.A. Bateman. 1991. *Text Generation and Systemic-Functional Linguistics*. London, England: Pinter.

Mauldin, M.L. 1991. *Conceptual Information Retrieval—A Case Study in Adaptive Partial Parsing*. Boston, MA: Kluwer Academic Publishers.

McKeown, K.R. and D.R. Radev. 1995. Generating Summaries of Multiple News Articles. In *Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR), 74–82. Seattle, WA.

Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University.

Paice, C.D. 1990. Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing and Management* 26(1): 171–186.

The Penman Primer, User Guide, and Reference Manual. 1988. Unpublished documentation, Information Sciences Institute, University of Southern California.

Quinlan, J.R. 1986. Induction of Decision Trees. *Machine Learning* 81–106.

Rasmussen, E. 1992. Clustering Algorithms. In W.B. Frakes and R. Baeza-Yates (eds.), *Information Retrieval: Data Structures and Algorithms*, 419–441. Upper Saddle River, NJ: Prentice-Hall.

Rau, L.S. and P.S. Jacobs. 1991. Creating Segmented Databases from Free Text for Text Retrieval. In *Proceedings of the Fourteenth Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR), 337–346. New York, NY.

Van Rijsbergen, C.J. 1979. *Information Retrieval* (2nd edition). London, England: Butterworths.

Riloff, E. and W.G. Lehnert. 1994. Information Extraction as a Basis for High-Precision Text Classification. *ACM Transactions on Information Systems*, vol. 12: 296–333.

Salton, G. 1988. *Automatic Text Processing*. Reading, MA: Addison-Wesley.

Salton, G., J. Allen, C. Buckley, and A. Singhal. 1994. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. *Science* 264: 1421–1426.

Schank, R.C. and R.P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Teufel, S. and M. Moens. 1997. Sentence Extraction as a Classification Task. In *Proceedings of the Workshop on Intelligent Scalable Summarization*. ACL/EACL Conference, 58–65. Madrid, Spain.

TREC. Harman, D. (ed). 1995. *Proceedings of the TREC Conference*.

Yarowsky, D. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*, 454–460. Nantes, France.