Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries

Michael J. Witbrock Vibhu O. Mittal

Just Research
4616 Henry Street
Pittsburgh, PA 15213
mwitbrock@lycos.com, mittal@justresearch.com

Abstract

Using current extractive summarization techniques, it is impossible to produce a coherent document summary shorter than a single sentence, or to produce a summary that conforms to particular stylistic constraints. Ideally, one would prefer to understand the document, and to generate an appropriate summary directly from the results of that understanding. Absent a comprehensive natural language understanding system, an approximation must be used. This paper presents an alternative statistical model of a summarization process, which jointly applies statistical models of the term selection and term ordering process to produce brief coherent summaries in a style learned from a training corpus.

1 Introduction

Summarization is one of the most important capabilities required in writing. Effective summarization, like effective writing, is neither easy nor innate; rather, it is a skill that is developed through instruction and practice [Hidi and Anderson, 1986; Hooper *et al.*, 1994]. Generating an effective summary requires the summarizer to select, evaluate, order and aggregate items of information according to their relevance to a particular subject or for a particular purpose. In the absence of a comprehensive natural language understanding system, an approximation must be used. Almost all previous work on computational implementations of summarization has focused on *extractive summarization*: selecting text spans - either complete sentences or paragraphs – from the original document. These extracts are then arranged in a linear order (usually the same order as in the original, larger document) to form a new, summary, document. There are several drawbacks to this approach, but our focus, in this paper, is on addressing one particular important limitation: the inability of extractive summarizers to generate summaries shorter than the text-spans being evaluated and ranked. Since most extractive summarizers have in the past considered the sentence to be the minimal unit of extracted text,[1] this means that the shortest summaries that these systems can produce must be at least one sentence long. This can be problematic in many cases, especially if a short "headline" is desired. This is due to the fact that (1) sentences selected for summaries often tend to be longer than the

[1] Some researchers have also looked at extracting paragraphs rather than sentences [Strzalkowski *et al.*, 1998; Mitra *et al.*, 1997].

average sentence in the document, and (2) the most important information in the document is often scattered across multiple sentences; extractive summarization cannot combine, either syntactically or semantically, concepts mentioned in the different text spans of the source document without using the whole spans.

This paper describes an alternative approach to summarization, not based on sentence extraction, capable of generating summaries of any desired length: it does so by statistically learning models of both content selection and realization; given an appropriate training corpus, it can generate summaries similar to the training ones, of any desired length. This approach has several advantages and novel applications compared to the text span extraction based approaches. The rest of the paper discusses our framework, some of the pros and cons of this technique, illustrates preliminary examples of its working with examples from our training and test corpora, and concludes with a brief description of our ongoing work.

2 Background and Related Work

Most of the previous work on summarization has focused on extractive methods. Starting with some of the earlier references to Luhn's work in the fifties [Luhn, 1958], researchers have focused on issues such as the use of lexical occurrence statistics, positional indicators (beginning of the document versus end of the document, for instance) [Edmundson, 1964], possible negative factors (for instance, words that might indicate lesser significance) [Mathis *et al.*, 1973], etc. More recently, Salton and his colleagues experimented with probabilistic measures for word importance [Salton *et al.*, 1997], Marcu looked at learning structural importance [Marcu, 1997], and Hovy and Lin looked at machine learning approaches for positional importance [Hovy and Lin, 1997].

In contrast to the large amount of work that has been undertaken in extractive summarization, there has been much less work on a generative model of summarization. The earliest approaches to generative models were discussed in the context of the FRUMP system [DeJong, 1982], which possessed a set of templates for extracting information from news stories and presenting it in the form of a summary. However, neither the content selection part, nor the generation part was learned by the system: the extraction templates were handcrafted for a particular application domain and the generation process required a set of manually specified sentence templates. Systems such as SUSY [Fum *et al.*, 1986], TOPIC [Reimer and Hahn, 1988] and SCISOR [Rau *et al.*, 1989] were similar, each experimenting with different aspects (underlying knowledge representation structures, number of features to be considered, etc.) The most recently reported work on generative summarization consists of the Columbia summarizer [Radev and McKeown, 1998], which uses a manually specified generative grammar of English to construct English sentences from an underlying

knowledge representation that uses manually crafted rules for content selection. However, none of these systems can: (1) generate summaries that may be a single noun phrase, and not a complete sentence, (2) learn rules/procedures/templates for either content selection or generation from a suitable training corpus.

The work reported in this paper is perhaps more closely related to work on statistical machine translation than on summarization. For instance, the CANDIDE system at IBM [Brown *et al.*, 1993] uses a translation model describing correspondences between sets of words in a source language and sets of words in a target language, and an ordering model describing the likelihood of sequences in a target language to achieve the goal of natural language translation. Thus, in some sense, our system can be considered to be 'translating' between two languages, one verbose and the other succinct. However, this analogy only holds at a very general level; there are several important differences between the two systems. The most important of these is that whereas our system can, in principle, produce a variety of derived documents, chiefly summarizations and brief characterizations of larger documents or document sets, these derivations need not be either semantically equivalent or complete. Because the IBM system was designed to be a translation system, it was forced to (statistically) capture the complete set of correct senses and nuances of the concepts in the source document and express them in the target document. As we will discuss, relaxing this constraint allows us considerable flexibility.

3 System Design and Operation

A high-level view of the system is shown in Figure 1. The main steps in the processing are:

1. A suitable corpus of documents with their corresponding headlines or summaries is assembled. In our case, we used news-wire articles from Reuters and the Associated Press available from the LDC. The target documents – the summaries – that the system needs to learn the translation mapping to, were the headlines accompanying the news stories.

2. The documents are preprocessed to identify items that can be used in determining summary contents. In the system described in this paper, the pre-processing included tokenization. Currently, the tokens are contiguous character sequences, not including punctuation symbols, spaces or carriage returns. In principle, tokens may include not only the words, but also additional information such as parts of speech tags,[2] semantic tags applied to words, even phrases. Conceivably, long distance relationships between words or phrases in the document, structural information obtained from the document such

[2]We have a preliminary version of a system that takes advantage of part of speech tags, but have not completed its evaluation.

1: A high level view of the system architecture.

as positions of words or phrases, mark-up information obtained from the document such as existence of different font, etc. could also be used.

3. The same pre-processing model is applied to the target documents.

4. A statistical model is built describing the relationship between the source text units in a document and the target text units to be used in the summary of that document. This model describes both the order and likelihood of appearance of the tokens in the target documents in the context of certain tokens in the source and a partial target document.

5. The statistical models generated in step (4), together with information about user or task requirements, are used to produce the headline/summary of a document.

Everyone in whom Ms. Lewinsky confided in detail believed she was telling the truth about her relationship with the President. Ms. Lewinsky told her psychologist, Dr. Irene Kassorla, about the affair shortly after it began. Thereafter, she related details of sexual encounters soon after they occurred (sometimes calling from her White House office).

Figure 2: Example document (excerpted from the Starr Report)

Consider the document shown in Figure 2.[3] The goal is to get the system to generate a summary for this document based on a set of training documents and their corresponding summaries. Possible headlines/summaries one might imagine for the article in Figure 2 include: (1) *Nature of President Clinton's Relationship with Monica Lewinsky: Ms Lewinsky's Confidants.* (Based on actual section headers in the Starr report). (2) *Lewinsky Confidants on Clinton Affair* (A "headline style" summary.) As in generation, there are, conceptually, at least two sub-tasks that the system must undertake: (1) *content selection*: (a) information to present in the summary, (b) level of detail to include in the summary, and (2) *surface realization* or linearization: how to phrase the in a syntactically valid and coherent fashion. The goal of our system is to learn operational metrics for both these sub-tasks automatically from the training data – corpora containing large numbers of matched source and target documents (or documents and headlines in our case); mechanisms for both (1) selecting the contents of the most likely summaries of a particular length, and (2) generating coherent English (or any other language) text to express the content selected in step (1).

3.1 Content Selection

The training corpus is used to learn a model of the relationship between the appearance of some features[4] in the document, and the appearance of features in the summary. In the simplest case this model consists of a mapping between the appearance of a word in the document, and the likelihood of some word appearing in the summary. For computational reasons, the early implementation evaluated here simply models the conditional probability of a word occurring in the summary given that the same word appeared in the document. Table 1 shows part of this mapping for words in the example excerpt.

The content selection score for the phrase "details of sexual", or any reordering of those words, under

[3] From the use of which no political inferences should be drawn.

[4] These features are word tokens in these initial experiments, but could also be any text spans, labels, or other syntactic and semantic features of the document.

Word	Probability
Details	0.7500
Of	0.9977
Sexual	1.0000
White	0.9223
House	0.9641

Table 1: Conditional probability of a word appearing in the summary, given that it appears in the document.

this scheme is simply the product of the individual probabilities:

$$Pr(\text{“}Details''|''Details'' in\ document) \cdot Pr(\text{“}of''|\text{“}of'' in\ document) \cdot Pr(\text{“}sexual''|\text{“}sexual'' in\ document) \quad (1)$$

Clearly, it is trivial to extend this approach to model more complex relationships amongst arbitrary subsets of tokens in the source and target documents. These relationships need not be just between tokens, but also between characterizations of these tokens, such as POS tags, token lengths, or derived statistics such as the proportion of nouns in the document, average sentence length, etc. It should be noted that a consequence of this freedom in choosing a content selection model is that the system is now capable of learning relationships between target-summary terms that are *not* in the document and terms that are in the document, and then apply those relationships to new documents, thereby introducing new terms into the summary.[5]

Once a content selection model has been trained on a suitable document/summary corpus, it can be used to compute selection scores for candidate summary terms, given the terms occurring in a particular source document. In conjunction with the summary structure model, described below, these scores can be used to compute the most likely summary candidates for particular parameters (such as summary length) and their rankings against one another. Since the probability of a word appearing in a summary can be considered to be independent of the structure of the summary,[6] the overall probability of a particular summary candidate can be computed by multiplying the probabilities of the content in the summary with the probability of that content being expressed using a particular summary structure.

It is worth noting that since there is no limitation on the types of relationships that can be expressed

[5]Pragmatic constraints, such as lack of sufficient memory to test this approach on our corpus, have thus far prevented us from producing such a model, but there is no reason why this model should not be learned; in fact, it is likely to improve the quality of the mappings learned.

[6]Note that this is not necessarily so; this independence assumption is a modeling choice.

Word	Log probability of word in Reuters headlines
Details	-4.0764
Of	-2.0272
Sexual	-3.903
White	-2.8417
House	-2.5623

Table 2: Probability of finding particular words in a summary (in this case, a Reuters' headline).

Word pair (word 1, word 2)	Log probability of word 2 given word 1
Details of	-0.8129
Of sexual	-2.6516
White house	-0.0304

Table 3: Probability of finding pairs of words in sequence in training summaries (in Reuters' headlines).

in the content selection model, variations on this approach can use appropriate training corpora to produce cross-lingual summaries. In this case, a model of the probability that an English word should appear in a summary for a Japanese document containing a certain set of terms could be used to simultaneously translate and summarize Japanese documents. We have conducted preliminary experiments on this task; more details can be obtained from [Witbrock and Mittal, 1998]. More speculatively, one could imagine cross-media summarization, in which an inventory of spoken word forms could be used, together with a concatenative synthesis algorithm and a table of conditional probabilities that speech segments would be used in a spoken summary of a particular document, to generate spoken summaries. Similarly, corresponding video or other media could be chosen to represent the content of a document.

3.2 Surface Realization

The probability of any particular surface form (as a headline candidate) – such as, "details of sexual...." can be computed by modeling the probability of word sequences. One of the simplest such models is a bigram language model, where the probability of a word sequence is approximated by multiplying out the probabilities of seeing each term given its left context. In the case of the candidate given above, the value would be given by:

$$Log(Pr(\text{“}Details''))+Log(Pr(\text{“}of''|\text{“}details''))+Log(Pr(\text{“}sexual''|\text{“}of'')) \quad (2)$$

1:	house	-3.49	Beam 31
2:	white house	-3.42	Beam 12
3:	the white house	-6.37	Beam 29
4:	white house of affair	-7.79	Beam 80
5:	white house of sexual affair	-8.29	Beam 81
6:	the White House of sexual affair	-11.12	Beam 81
7:	white house of sexual affair with it	-12.20	Beam 81
8:	white house of sexual affair with it soon	-13.36	Beam 81
9:	the white house of sexual affair with it soon	-16.15	Beam 81

Figure 3: Sample output from the system using only word-level mappings. The figures to the right are the overall log probabilities of the proposed summaries, and the number of terms being considered, on average, for each emitted word.

which, using the values in the tables, yields a log probability of -7.5409. Alternative sequences, using the same words, such as "of sexual details" have probabilities that can be calculated similarly. In this case, the sequence "sexual details" is so unlikely that it has not appeared in the training data, and is estimated using a back-off weight [Katz, 1987]:

$$Log(Pr(\text{“}of''))+Log(Pr(\text{“}sexual''|\text{“}of'')+(Log(\text{backoff}(\text{“}sexual''))+Log(Pr(\text{“}details'')))), \quad (3)$$

yielding an estimated log probability for the sequence of -10.034, indicating that this sequence is about 310 times less like part of a headline than the previous one.

As mentioned earlier, these calculations can be extended to take into account the likelihood of additional information (semantic tags such as named-entities, or syntactic tags such as POS information), both at the word or phrase level, or can be carried out with respect to any textual spans from characters on up. They can also, of course, be extended to use higher order n-grams, providing that sufficient numbers of training headlines are available to estimate the probabilities.

3.3 Search

Even though content selection and summary structure generation have been presented separately, there is no reason for them to occur independently, and in fact, in our current implementation, they are used simultaneously to contribute to an overall weighting scheme that ranks possible summary candidates against each other. In the case of the phrase discussed above, the overall weighting used in ranking can be obtained as a

1:	time	-3.76	Beam 40
2:	new customers	-4.41	Beam 81
3:	dell computer products	-5.30	Beam 88
4:	new power macs strategy	-6.04	Beam 90
5:	apple to sell macintosh users	-8.20	Beam 86
6:	new power macs strategy on internet	-9.35	Beam 88
7:	apple to sell power macs distribution strategy	-10.32	Beam 89
8:	new power macs distribution strategy on internet products	-11.81	Beam 88
9:	apple to sell power macs distribution strategy on internet	-13.09	Beam 86

Figure 4: Sample output from the system using word-level mappings.

weighted combination of the content and structure model log probabilities:[7]

$$\begin{aligned} \alpha * \quad & \Big[Log(Pr(\text{"}Details\text{"}|\text{"}Details\text{"} in\ doc)) + Log(Pr(\text{"}of|\text{"}of\text{"} in\ doc)) + \\ & \qquad Log(Pr(\text{"}sexual\text{"}|\text{"}sexual\text{"} in\ doc))\Big] + \\ \beta * \quad & \Big[Log(Pr(\text{"}Details\text{"})) + Log(Pr(\text{"}of\text{"}|\text{"}details\text{"})) + Log(Pr(\text{"}sexual\text{"}|\text{"}of\text{"}))\Big] \end{aligned} \tag{4}$$

To generate a summary, it is necessary to find a sequence of words that maximizes the probability, under the content selection and summary structure models, that it was generated from the document to be summarized. Since, in this initial implementation, each summary term is selected independently, and the summary structure model is first order Markov, Viterbi beam search [Forney, 1973] could be used to efficiently find a near-optimal summary.[8] Other statistical models might require the use of a different heuristic search algorithm. An example of the results of commanding the search to output the most highly ranked candidate, for a variety of values of the summary length control parameter, is shown in Figure 3.

Figure 4 shows the set of headlines generated by the system when run against a real news story discussing Apple Computer's decision to start direct internet sales and comparing it to the strategy of other computer makers.

[7] In our current implementation, we set both the weights α and β to 1.0.

[8] In the first implementation, a beam width of three, and a minimum beam size of twenty states was used. The Markov assumption was violated by using backtracking at every state to strongly discourage paths that repeated terms, since bigrams that start repeating often seem to pathologically overwhelm the search otherwise.

4 Experiments and Discussion

To gain a better understanding of how well this approach works, this section briefly discusses two sets of experiments that we conducted. We trained the system on approximately 8000 news articles from Reuters dated between 1/1/97 and 6/1/97. These contained almost 44,000 unique tokens in the articles and slightly more than 15,000 tokens in the headlines (after stripping punctuation marks). Since representing the conditional probabilities for each pair of these words would have required a matrix with $6.6x10^7$ entries, and our computer resources and training data were limited, we decided to take a simpler approach and initially investigate the effectiveness of training on a smaller set of words: those words that appeared in the headlines.[9] Thus, the system calculated conditional probabilities for words in the headlines that also appeared in the article bodies. To keep the model as simple as possible, we also limited the system to learn only bigram transition probabilities for the headline syntax. Sample output from the first runs of system is shown in Figure 3 and Figure 4. For such a simple system, it performed surprisingly well. Of course, there are some obvious problems, but they should be relatively straightforward to fix, given sufficient training data. Ignoring the grammatical problems, the system was able to pick out the main issues in the stories: the white house, and an affair,[10] and Apple and internet distribution, respectively. Another problem is that there does not seem to be an obvious stopping point for the system – it can generate longer and longer headlines. We believe that it will be possible to learn a model of headline length as a function of story content, but simply parameterizing the length was more straightforward for our initial experiments.

To evaluate this version of the system, we decided to compare its output against the actual headlines for an untrained set of 1000 input Reuters news stories. Since we are cannot compare phrasing, we compared the generated headlines against (i) the actual headlines, as well as (ii) the top ranked summary sentence of the story.[11] Since the system does not currently have a mechanism to determine the optimal length of a headline, we generated six headlines for each story, ranging in length from 4 to 10 words and measured the term-overlap between each of the generated headlines and the test "standard" (both the actual headline and the summary sentence). For each story, we found the maximum overlap between these two and noted the length at which this overlap was maximal. We also measured a stricter measure of effectiveness: for headlines that matched completely – that is, all of the words in the generated headline were present in the

[9] An alternative approach to limiting the size of the mappings that need to be estimated would be to use only the top n words, where n could have a small value in the hundreds, rather than the thousands.

[10] The perfect headline would have been "Affair at the White House."

[11] This was done in an effort to overcome problems in which headlines used a different vocabulary from that used in the story itself.

Gen. Headline Length (words)	Overlap w/ actual headline (Min: 0.12, Max: 1.0)	Overlap w/ top summary sentence (Min: 0, Max: 1.0)	Percentage of complete matches
4	0.89	0.91	19.71%
5	0.87	0.85	14.10%
6	0.89	0.90	12.14%
7	0.90	0.89	08.70%
8	0.87	0.89	11.90%
9	0.89	0.95	19.40%

Table 4: Evaluating the system's effectiveness at generating headlines from 1000 Reuters' news articles.

actual headline – we noted the lengths of the generated headline. These statistics illustrate how well the system does at selecting content words for the headlines. (Phrasing quality is difficult, if not impossible to measure objectively. Actual headlines are often ungrammatical, incomplete phrases. We expect that longer n-gram models and part-of-speech based models will help our system generate headlines that are very similar in phrasing to real headlines.) The statistics for these experiments are shown in Table 4.

It should be noted in the system's defense that many of the headlines generated by the system were very good, but were penalized because they did not match the original ones. For instance, in the case of a story about a NASA satellite rescue mission, the system generated the following headline: "*space shuttle satellite rescue bid*." But this was not scored as a good headline because it was being compared against the "standard" one which was "*NASA Considers Satellite Rescue Bid*." Thus, the results presented here probably err on the stricter side.

5 Conclusions and Future Work

This paper has presented an alternative to extractive summarization: an approach that makes it possible to generate coherent summaries that are shorter than a single sentence and conform to particular stylistic constraints. Our approach applies statistical models of the term selection and term ordering processes to produce novel, brief summaries of any desired length. The strength of this approach is that it enables summaries that are more compact than previously possible; furthermore, these summaries need not contain any of the words in the original document, unlike previous statistical summarization systems. Given sufficiently good quality training corpora, this approach can be used to generate headline-style summaries from a variety of formats in various applications: for instance, we have been experimenting with corpora that contain

Japanese documents and English headlines. (Since this corpora was constructed by running an extremely unsophisticated lexical translation system over Japanese headlines, the results are not very good, as yet.)

Further experiments with this approach are currently under way. There are clear short-comings of the system that need to be addressed. Some of these shortcomings can be fixed by better training data. A corpus of suitably annotated data (with the requisite mark-up to indicate additional information, such as focus, discourse structure, or even co- or anaphoric-reference information) would help us greatly in evaluating the improvements possible. We would also like to be able to incorporate, in our model, external information, such as user interactions or other biases to optimize both the content and the form of generated summaries.

Other deficiencies can be addressed by the use of more sophisticated content selection and summary structure models. As a first step in this direction, we have begun work on a system that uses automated part of speech markup to allow better modeling of summary structure. We are also working on a model that uses the distance between words in the original story to condition the probability that they should appear separated by some distance in the headline. In the future, we hope to extend this work by using, for example, subject-verb relationships in the story to constrain subject-verb relationships in the generated headlines.

More speculatively, future work may permit the application of this summarization scheme to the problem of learning summaries that are less indicative of the content, and more like an evaluation. Thus, the system could learn the mappings between documents and assessments, such as "this is a good essay, but has your choice of terms could be improved, and some punctuation is missing", or "this editorial column is tendentious puffery".[12]

References

[Aone *et al.*, 1997] Chinatsu Aone, M. E. Okurowski, J. Gorlinsky, and B. Larsen. A scalable summarization system using robust NLP. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 66–73, Madrid, Spain, 1997.

[Brown *et al.*, 1993] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, (2):263–312, 1993.

[DeJong, 1982] Gerald F. DeJong. An overview of the FRUMP system. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*, pages 149–176. Lawrence Erlbaum Associates, Hillsdale, NJ, 1982.

[Edmundson, 1964] H. P. Edmundson. Problems in automatic extracting. *Communications of the ACM*, 7:259–263, 1964.

[12] This is similar in intent to the work on automatic essay grading using LSA [Larkey, 1998].

[Forney, 1973] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, pages 268–278, 1973.

[Fum *et al.*, 1986] D. Fum, G. Guida, and C. Tasso. Tailoring importance evaluation to reader's goals: a contribution to descriptive text summarization. In *Proceedings of COLING-86*, pages 256–259, 1986.

[Hidi and Anderson, 1986] S. Hidi and V. Anderson. Producing written summaries: Task demands, cognitive operations, and implications for instruction. *Review of Educational Research*, 56:473–493, 1986.

[Hooper *et al.*, 1994] S. Hooper, G. Sales, and S. D. Rysavy. Generating summaries and analogies alone and in pairs. *Contemporary Educational Psychology*, 19(1):53–62, January 1994.

[Hovy and Lin, 1997] Eduard Hovy and Chin-Yew Lin. Automated text summarization in SUMMARIST. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 18–24, Madrid, Spain, 1997.

[Katz, 1987] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24, 1987.

[Kupiec *et al.*, 1995] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of ACM/SIGIR '95*, pages 68–73. ACM, 1995.

[Larkey, 1998] Leah Larkey. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st ACM/SIGIR (SIGIR-98)*, pages 90–96. ACM, 1998.

[Luhn, 1958] P. H. Luhn. Automatic creation of literature abstracts. *IBM Journal*, pages 159–165, 1958.

[Marcu, 1997] Daniel Marcu. From discourse structures to text summaries. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain, 1997.

[Mathis *et al.*, 1973] B. A. Mathis, J. E. Rush, and C. E. Young. Improvement of automatic abstracts by the use of structural analysis. *Journal of the American Society for Information Science*, 24:101–109, 1973.

[Mitra *et al.*, 1997] M. Mitra, Amit Singhal, and Chris Buckley. Automatic text summarization by paragraph extraction. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 1997.

[Radev and McKeown, 1998] Dragomir Radev and Kathy McKeown. Generating natural language summaries from multiple online sources. *Compuutational Linguistics*, 1998.

[Rau *et al.*, 1989] Lisa F. Rau, Paul S. Jacobs, and Udi Zernick. Information extraction and text summarization using linguistic knowledge acquisition. *Info. Proc. and Management*, 25(4):419–428, 1989.

[Reimer and Hahn, 1988] U. Reimer and U. Hahn. Text condensation as knowledge base abstraction. In *Proceedings of the Fourth Conference on Artificial Intelligence Applications*, pages 338–344, March 1988.

[Salton *et al.*, 1997] Gerard Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summary. *Info. Proc. and Management*, 33(2):193–207, March 1997.

[Strzalkowski *et al.*, 1998] T. Strzalkowski, J. Wang, and B. Wise. A robust practical text summarization system. In *AAAI Intelligent Text Summarization Workshop*, pages 26–30, Stanford, CA, March 1998.

[Witbrock and Mittal, 1998] Michael J. Witbrock and Vibhu O. Mittal. A statistical approach to generating summaries, headlines or synopses: Representing and reasoning with translation models. Technical report, Justsystem Pittsburgh Research Center, Pittsburgh, PA 15213, December 1998.