Cut and Paste Based Text Summarization

Hongyan Jing and Kathleen R. McKeown
Department of Computer Science
Columbia University
New York, NY 10027, USA
hjing, kathy@cs.columbia.edu

Abstract

We present a cut and paste based text summarizer, which uses operations derived from an analysis of human written abstracts. The summarizer edits extracted sentences, using reduction to remove inessential phrases and combination to merge resulting phrases together as coherent sentences. Our work includes a statistically based sentence decomposition program that identifies where the phrases of a summary originate in the original document, producing an aligned corpus of summaries and articles which we used to develop the summarizer.

1 Introduction

There is a big gap between the summaries produced by current automatic summarizers and the abstracts written by human professionals. Certainly one factor contributing to this gap is that automatic systems can not always correctly identify the important topics of an article. Another factor, however, which has received little attention, is that automatic summarizers have poor text generation techniques. Most automatic summarizers rely on extracting key sentences or paragraphs from an article to produce a summary. Since the extracted sentences are disconnected in the original article, when they are strung together, the resulting summary can be inconcise, incoherent, and sometimes even misleading.

We present a cut and paste based text summarization technique, aimed at reducing the gap between automatically generated summaries and human-written abstracts. Rather than focusing on how to identify key sentences, as do other researchers, we study how to generate the text of a summary once key sentences have been extracted.

The main idea of cut and paste summarization is to reuse the text in an article to generate the summary. However, instead of simply extracting sentences as current summarizers do, the cut and paste system will "smooth" the extracted sentences by editing them. Such edits mainly involve cutting phrases and pasting them together in novel ways.

The key features of this work are:

(1) The identification of cutting and pasting operations. We identified six operations that can be used alone or together to transform extracted sentences into sentences in human-written abstracts. The operations were identified based on manual and automatic comparison of human-written abstracts and the original articles. Examples include sentence reduction, sentence combination, syntactic transformation, and lexical paraphrasing.

(2) Development of an automatic system to perform cut and paste operations. Two operations - sentence reduction and sentence combination - are most effective in transforming extracted sentences into summary sentences that are as concise and coherent as in human-written abstracts. We implemented a sentence reduction module that removes extraneous phrases from extracted sentences, and a sentence combination module that merges the extracted sentences or the reduced forms resulting from sentence reduction. Our sentence reduction model determines what to cut based on multiple sources of information, including syntactic knowledge, context, and statistics learned from corpus analysis. It improves the conciseness of extracted sentences, making them concise and on target. Our sentence combination module implements combination rules that were identified by observing examples written by human professionals. It improves the coherence of extracted sentences.

(3) Decomposing human-written summary sentences. The cut and paste technique we propose here is a new computational model which we based on analysis of human-written abstracts. To do this analysis, we developed an automatic system that can match a phrase in a human-written abstract to the corresponding phrase in the article, identifying its most likely location. This decomposition program allows us to analyze the construction of sentences in a human-written abstract. Its results have been used to train and test the sentence reduction and sentence combination module.

In Section 2, we discuss the cut and paste technique in general, from both a professional and computational perspective. We also describe the six cut and paste operations. In Section 3, we describe the system architecture. The major components of the system, including sentence reduction, sentence combination, decomposition, and sentence selection, are described in Section 4. The evaluation results are shown in Section 5. Related work is discussed in Section 6. Finally, we conclude and discuss future work.

2 Cut and paste in summarization

2.1 Related work in professional summarizing

Professionals take two opposite positions on whether a summary should be produced by cutting and pasting the original text. One school of scholars is opposed; "(use) your own words... Do not keep too close to the words before you", states an early book on abstracting for American high school students (Thurber, 1924). Another study, however, shows that professional abstractors actually rely on cutting and pasting to produce summaries: "Their professional role tells abstractors to avoid inventing anything. They follow the author as closely as possible and reintegrate the most important points of a document in a shorter text" (Endres-Niggemeyer et al., 1998). Some studies are somewhere in between: "summary language may or may not follow that of author's" (Fidel, 1986). Other guidelines or books on abstracting (ANSI, 1997; Cremmins, 1982) do not discuss the issue.

Our cut and paste based summarization is a *computational* model; we make no claim that humans use the same cut and paste operations.

2.2 Cut and paste operations

We manually analyzed 30 articles and their corresponding human-written summaries; the articles and their summaries come from different domains (15 general news reports, 5 from the medical domain, 10 from the legal domain) and the summaries were written by professionals from different organizations. We found that reusing article text for summarization is almost universal in the corpus we studied. We defined six operations that can be used alone, sequentially, or simultaneously to transform selected sentences from an article into the corresponding summary sentences in its human-written abstract:

(1) sentence reduction

Remove extraneous phrases from a selected sentence, as in the following example [1]:

> Document sentence: *When it arrives sometime next year in new TV sets,* the V-chip will give parents a *new and potentially revolutionary* device to block out programs they don't want their children to see.
> Summary sentence: The V-chip will give parents a device to block out programs they don't want their children to see.

The deleted material can be at any granularity: a word, a phrase, or a clause. Multiple components can be removed.

(2) sentence combination

Merge material from several sentences. It can be used together with sentence reduction, as illustrated in the following example, which also uses paraphrasing:

> Text Sentence 1: But it also raises serious questions about the privacy of such highly personal information *wafting about the digital world.*
> Text Sentence 2: The issue thus fits squarely into the broader debate about privacy and security on the internet, *whether it involves protecting credit card number or keeping children from offensive information.*
> Summary sentence: But it also raises the issue of privacy of such personal information *and* this issue hits the head on the nail in the broader debate about privacy and security on the internet.

(3) syntactic transformation

In both sentence reduction and combination, syntactic transformations may be involved. For example, the position of the subject in a sentence may be moved from the end to the front.

(4) lexical paraphrasing

Replace phrases with their paraphrases. For instance, the summaries substituted *point out* with *note*, and *fits squarely into* with a more picturesque description *hits the head on the nail* in the previous examples.

(5) generalization or specification

Replace phrases or clauses with more general or specific descriptions. Examples of generalization and specification include:

> Generalization: *"a proposed new law that would require Web publishers to obtain parental consent before collecting personal information from children"* → *"legislation to protect children's privacy on-line"*
> Specification: *"the White House's top drug official"* → *"Gen. Barry R. McCaffrey, the White House's top drug official"*

[1] All the examples in this section were produced by human professionals

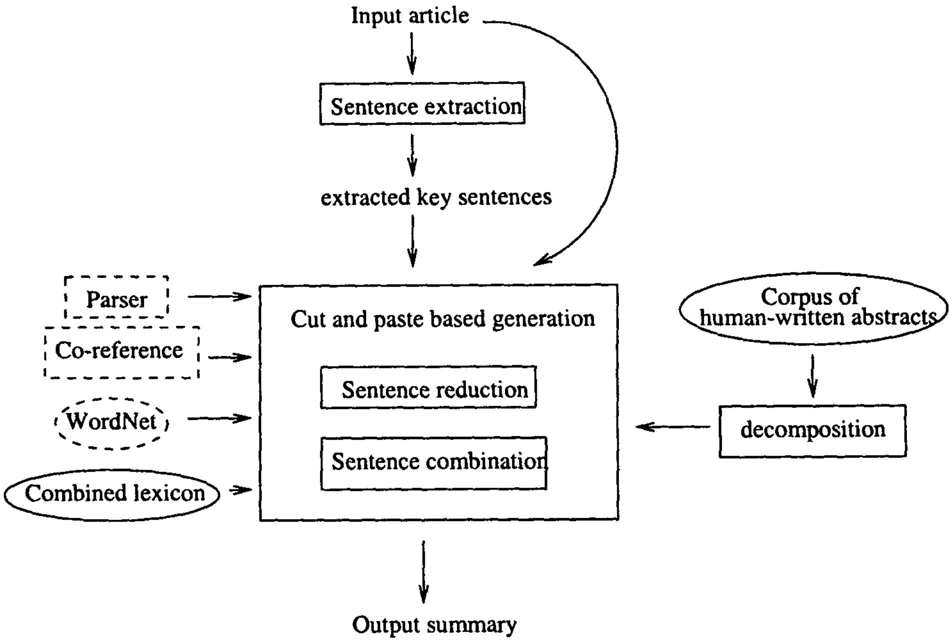

Figure 1: System architecture

(6) reordering
Change the order of extracted sentences. For instance, place an ending sentence in an article at the beginning of an abstract.

In human-written abstracts, there are, of course, sentences that are not based on cut and paste, but completely written from scratch. We used our decomposition program to automatically analyze 300 human-written abstracts, and found that 19% of sentences in the abstracts were written from scratch. There are also other cut and paste operations not listed here due to their infrequent occurrence.

3 System architecture

The architecture of our cut and paste based text summarization system is shown in Figure 1. Input to the system is a single document from any domain. In the first stage, *extraction*, key sentences in the article are identified, as in most current summarizers. In the second stage, *cut and paste based generation*, a sentence reduction module and a sentence combination module implement the operations we observed in human-written abstracts.

The cut and paste based component receives as input not only the extracted key sentences, but also the original article. This component can be ported to other single-document summarizers to serve as the generation component, since most current summarizers extract key sentences - exactly what the extraction module in our system does.

Other resources and tools in the summarization system include a corpus of articles and their human-written abstracts, the automatic decomposition program, a syntactic parser, a co-reference resolution system, the WordNet lexical database, and a large-scale lexicon we combined from multiple resources. The components in dotted lines are existing tools or resources; all the others were developed by ourselves.

4 Major components

The main focus of our work is on decomposition of summaries, sentence reduction, and sentence combination. We also describe the sentence extraction module, although it is not the main focus of our work.

4.1 Decomposition of human-written summary sentences

The decomposition program, see (Jing and McKeown, 1999) for details, is used to analyze the construction of sentences in human-written abstracts. The results from decomposition are used to build the training and testing corpora for sentence reduction and sentence combination.

The decomposition program answers three questions about a sentence in a human-written abstract: (1) Is the sentence constructed by cutting and pasting phrases from the input article? (2) If so, what phrases in the sentence come from the original article? (3) Where in the article do these phrases come from?

We used a Hidden Markov Model (Baum, 1972) solution to the decomposition problem. We first mathematically formulated the problem, reducing it to a problem of finding, for each word in a summary

Summary sentence:
(F0:S1 **arthur b sackler vice president for law and public policy of time warner inc**) (F1:S-1 *and*) (F2:S0 **a member of the direct marketing association told**) (F3:S2 **the communications subcommittee of the senate commerce committee**) (F4:S-1 *that legislation*) (F5:S1**to protect**) (F6:S4 **children' s**) (F7:S4 **privacy**) (F8:S4 **online**) (F9:S0 **could destroy the spontaneous nature that makes the internet unique**)

Source document sentences:
Sentence 0: a proposed new law that would require web publishers to obtain parental consent before collecting personal information from children **(F9 could destroy the spontaneous nature that makes the internet unique) (F2 a member of the direct marketing association told)** a senate panel thursday
Sentence 1: **(F0 arthur b sackler vice president for law and public policy of time warner inc**) said the association supported efforts **(F5 to protect**) children online but he urged lawmakers to find some middle ground that also allows for interactivity on the internet
Sentence 2: for example a child's e-mail address is necessary in order to respond to inquiries such as updates on mark mcguire's and sammy sosa's home run figures this year or updates of an online magazine sackler said in testimony to **(F3 the communications subcommittee of the senate commerce committee)**
Sentence 4: the subcommittee is considering the **(F6 children's) (F8 online) (F7 privacy)** protection act which was drafted on the recommendation of the federal trade commission

Figure 2: Sample output of the decomposition program

sentence, a document position that it most likely comes from. The position of a word in a document is uniquely identified by the position of the sentence where the word appears, and the position of the word within the sentence. Based on the observation of cut and paste practice by humans, we produced a set of general heuristic rules. Sample heuristic rules include: two adjacent words in a summary sentence are most likely to come from two adjacent words in the original document; adjacent words in a summary sentence are not very likely to come from sentences that are far apart in the original document. We use these heuristic rules to create a Hidden Markov Model. The Viterbi algorithm (Viterbi, 1967) is used to efficiently find the most likely document position for each word in the summary sentence.

Figure 2 shows sample output of the program. For the given summary sentence, the program correctly identified that the sentence was combined from four sentences in the input article. It also divided the summary sentence into phrases and pinpointed the exact document origin of each phrase. A phrase in the summary sentence is annotated as *(FNUM:SNUM actual-text)*, where *FNUM* is the sequential number of the phrase and *SNUM* is the number of the document sentence where the phrase comes from. SNUM = -1 means that the component does not come from the original document. The phrases in the document sentences are annotated as *(FNUM actual-text)*.

4.2 Sentence reduction

The task of the sentence reduction module, described in detail in (Jing, 2000), is to remove extraneous phrases from extracted sentences. The goal of reduction is to "reduce without major loss"; that is, we want to remove as many extraneous phrases as possible from an extracted sentence so that it can be concise, but without detracting from the main idea that the sentence conveys. Ideally, we want to remove a phrase from an extracted sentence *only if it is irrelavant to the main topic.*

Our reduction module makes decisions based on multiple sources of knowledge:

(1) Grammar checking. In this step, we mark which components of a sentence or a phrase are obligatory to keep it grammatically correct. To do this, we traverse the sentence parse tree, produced by the English Slot Grammar(ESG) parser developed at IBM (McCord, 1990), in top-down order and mark for each node in the parse tree, which of its children are obligatory. The main source of knowledge the system relies on in this step is a large-scale, reusable lexicon we combined from multiple resources (Jing and McKeown, 1998). The lexicon contains subcategorizations for over 5,000 verbs. This information is used to mark the obligatory arguments of verb phrases.

(2) Context information. We use an extracted sentence's local context in the article to decide which components in the sentence are likely to be most relevant to the main topic. We link the words in the extracted sentence with words in its local context, if they are repetitions, morphologically related, or linked with each other in WordNet through certain type of lexical relation, such as synonymy, antonymy, or meronymy. Each word in the extracted sentence gets an importance score, based on the number of links it has with other words and the types of links. Each phrase in the sentence is then assigned a score

Example 1:
Original sentence : *When it arrives sometime next year in new TV sets,* **the V-chip will give parents a new and potentially revolutionary device to block out programs they don't want their children to see.**
Reduction program: The V-chip will give parents a new and potentially revolutionary device to block out programs they don't want their children to see.
Professionals : The V-chip will give parents a device to block out programs they don't want their children to see.

Example 2:
Original sentence : **Som and Hoffman's creation would allow broadcasters to insert multiple ratings into a show,** *enabling the V-chip to filter out racy or violent material but leave unexceptional portions of a show alone.*
Reduction Program: Som and Hoffman's creation would allow broadcasters to insert multiple ratings into a show.
Professionals : Som and Hoffman's creation would allow broadcasters to insert multiple ratings into a show.

Figure 3: Sample output of the sentence reduction program

by adding up the scores of its children nodes in the parse tree. This score indicates how important the phrase is to the main topic in discussion.

(3) Corpus evidence. The program uses a corpus of input articles and their corresponding reduced forms in human-written abstracts to learn which components of a sentence or a phrase can be removed and how likely they are to be removed by professionals. This corpus was created using the decomposition program. We compute three types of probabilities from this corpus: the probability that a phrase is removed; the probability that a phrase is reduced (i.e., the phrase is not removed as a whole, but some components in the phrase are removed); and the probability that a phrase is unchanged at all (i.e., neither removed nor reduced). These corpus probabilities help us capture human practice.

(4) Final decision. The final reduction decision is based on the results from all the earlier steps. A phrase is removed only if it is not grammatically obligatory, not the focus of the local context (indicated by a low context importance score), and has a reasonable probability of being removed by humans. The phrases we remove from an extracted sentence include clauses, prepositional phrases, gerunds, and to-infinitives.

The result of sentence reduction is a shortened version of an extracted sentence [2]. This shortened text can be used directly as a summary, or it can be fed to the sentence combination module to be merged with other sentences.

Figure 3 shows two examples produced by the reduction program. The corresponding sentences in human-written abstracts are also provided for comparison.

4.3 Sentence combination

To build the combination module, we first manually analyzed a corpus of combination examples produced by human professionals, automatically created by the decomposition program, and identified a list of combination operations. Table 1 shows the combination operations.

To implement a combination operation, we need to do two things: decide when to use which combination operation, and implement the combining actions. To decide when to use which operation, we analyzed examples by humans and manually wrote a set of rules. Two simple rules are shown in Figure 4. Sample outputs using these two simple rules are shown in Figure 5. We are currently exploring using machine learning techniques to learn the combination rules from our corpus.

The implementation of the combining actions involves joining two parse trees, substituting a subtree with another, or adding additional nodes. We implemented these actions using a formalism based on Tree Adjoining Grammar (Joshi, 1987).

4.4 Extraction Module

The extraction module is the front end of the summarization system and its role is to extract key sentences. Our method is primarily based on lexical relations. First, we link words in a sentence with other words in the article through repetitions, morphological relations, or one of the lexical relations encoded in WordNet, similar to step 2 in sentence reduction. An importance score is computed for each word in a sentence based on the number of lexical links it has with other words, the type of links, and the directions of the links.

After assigning a score to each word in a sentence, we then compute a score for a sentence by adding up the scores for each word. This score is then normal-

[2] It is actually also possible that the reduction program decides no phrase in a sentence should be removed, thus the result of reduction is the same as the input.

Categories	Combination Operations
Add descriptions or names for people or organizations	add description (see Figure 5)
	add name
Aggregations	extract common subjects or objects (see Figure 5)
	change one sentence to a clause
	add connectives (e.g., *and* or *while*)
	add punctuations (e.g., ";")
Substitute incoherent phrases	substitute dangling anaphora
	substitute dangling noun phrases
	substitute adverbs (e.g., *here*)
	remove connectives
Substitute phrases with more general or specific information	substitute with more general information
	substitute with more specific information
Mixed operations	combination of any of above operations (see Figure 2)

Table 1: Combination operations

Rule 1:
IF: ((a person or an organization is mentioned the first time) and (the full name or the full description of the person or the organization exists somewhere in the original article but is missing in the summary))
THEN: replace the phrase with the full name plus the full description
Rule 2:
IF: ((two sentences are close to each other in the original article) and (their subjects refer to the same entity) and (at least one of the sentences is the reduced form resulting from sentence reduction))
THEN: merge the two sentences by removing the subject in the second sentence, and then combining it with the first sentence using connective "and".

Figure 4: Sample sentence combination rules

ized over the number of words a sentence contains. The sentences with high scores are considered important.

The extraction system selects sentences based on the importance computed as above, as well as other indicators, including sentence positions, cue phrases, and tf*idf scores.

5 Evaluation

Our evaluation includes separate evaluations of each module and the final evaluations of the overall system.

We evaluated the decomposition program by two experiments, described in (Jing and McKeown, 1999). In the first experiment, we selected 50 human-written abstracts, consisting of 305 sentences in total. A human subject then read the decomposition results of these sentences to judge whether they are correct. 93.8% of the sentences were correctly decomposed. In the second experiment, we tested the system in a summary alignment task. We ran the decomposition program to identify the source document sentences that were used to construct the sentences in human-written abstracts. Human subjects were also asked to select the document sentences that are semantic-equivalent to the sentences in the abstracts. We compared the set of sentences identified by the program with the set of sentences selected by the majority of human subjects, which is used as the gold standard in the computation of precision and recall. The program achieved an average 81.5% precision, 78.5% recall, and 79.1% f-measure for 10 documents. The average performance of 14 human judges is 88.8% precision, 84.4% recall, and 85.7% f-measure. Recently, we have also tested the system on legal documents (the headnotes used by Westlaw company), and the program works well on those documents too.

The evaluation of sentence reduction (see (Jing, 2000) for details) used a corpus of 500 sentences and their reduced forms in human-written abstracts. 400 sentences were used to compute corpus probabilities and 100 sentences were used for testing. The results show that 81.3% of the reduction decisions made by the system agreed with those of humans. The humans reduced the length of the 500 sentences by 44.2% on average, and the system reduced the length of the 100 test sentences by 32.7%.

The evaluation of sentence combination module is not as straightforward as that of decomposition or reduction since combination happens later in the pipeline and it depends on the output from prior

Example 1: add descriptions or names for people or organization

Original document sentences:

"We're trying to prove that there are big benefits to the patients by involving them more deeply in their treatment", **said Paul Clayton, Chairman of the Department dealing with computerized medical information at Columbia.**

"The economic payoff from breaking into health care records is a lot less than for banks", *said Clayton at Columbia.*

Combined sentence:

"The economic payoff from breaking into health care records is a lot less than for banks", said Paul Clayton, Chairman of the Department dealing with computerized medical information at Columbia.

Professional: *(the same)*

Example 2: extract common subjects

Original document sentences:

The new measure is an echo of the original bad idea, *blurred just enough to cloud prospects both for enforcement and for court review.*

Unlike the 1996 act, **this one applies only to commercial Web sites** *- thus sidestepping 1996 objections to the burden such regulations would pose for museums, libraries and freewheeling conversation deemed "indecent" by somebody somewhere.*

The new version also replaces the vague "indecency" standard, *to which the court objected, with the better-defined one of material ruled "harmful to minors."*

Combined sentences:

The new measure is an echo of the original bad idea.

The new version applies only to commercial web sites and replaces the vague "indecency" standard with the better-defined one of material ruled "harmful to minors."

Professional:

While the new law replaces the "indecency" standard with "harmful to minors" and now only applies to commercial Web sites, the "new measure is an echo of the original bad idea."

Figure 5: Sample output of the sentence combination program

modules. To evaluate just the combination component, we assume that the system makes the same reduction decision as humans and the co-reference system has a perfect performance. This involves manual tagging of some examples to prepare for the evaluation; this preparation is in progress. The evaluation of sentence combination will focus on the accessment of combination rules.

The overall system evaluation includes both intrinsic and extrinsic evaluation. In the intrinsic evaluation, we asked human subjects to compare the quality of extraction-based summaries and their revised versions produced by our sentence reduction and combination modules. We selected 20 documents; three different automatic summarizers were used to generate a summary for each document, producing 60 summaries in total. These summaries are all extraction-based. We then ran our sentence reduction and sentence combination system to revise the summaries, producing a revised version for each summary. We presented human subjects with the full documents, the extraction-based summaries, and their revised versions, and asked them to compare the extraction-based summaries and their revised versions. The human subjects were asked to score the conciseness of the summaries (extraction-based or revised) based on a scale from 0 to 10 - the higher the score, the more concise a summary is. They were also asked to score the coherence of the summaries based on a scale from 0 to 10. On average, the extraction-based summaries have a score of 4.2 for conciseness, while the revised summaries have a score of 7.9 (an improvement of 88%). The average improvement for the three systems are 78%, 105%, and 88% respectively. The revised summaries are on average 41% shorter than the original extraction-based summaries. For summary coherence, the average score for the extraction-based summaries is 3.9, while the average score for the revised summaries is 6.1 (an improvement of 56%). The average improvement for the three systems are 69%, 57%, and 53% respectively.

We are preparing a task-based evaluation, in which we will use the data from the Summarization Evaluation Conference (Mani et al., 1998) and compare how our revised summaries can influence humans' performance in tasks like text categorization and ad-hoc retrieval.

6 Related work

(Mani et al., 1999) addressed the problem of revising summaries to improve their quality. They suggested three types of operations: elimination, aggregation, and smoothing. The goal of the elimination operation is similar to that of the sentence reduction operation in our system. The difference is that while elimination always removes parentheticals, sentence-initial PPs and certain adverbial phrases for every extracted sentence, our sentence reduction module aims to make reduction decisions according to each case and removes a sentence component only if it considers it appropriate to do so. The goal of the aggregation operation and the smoothing operation is similar to that of the sentence combination operation in our system. However, the combination operations and combination rules that we derived from corpus analysis are significantly different from those used in the above system, which mostly came from operations in traditional natural language generation.

7 Conclusions and future work

This paper presents a novel architecture for text summarization using cut and paste techniques observed in human-written abstracts. In order to automatically analyze a large quantity of human-written abstracts, we developed a decomposition program. The automatic decomposition allows us to build large corpora for studying sentence reduction and sentence combination, which are two effective operations in cut and paste. We developed a sentence reduction module that makes reduction decisions using multiple sources of knowledge. We also investigated possible sentence combination operations and implemented the combination module. A sentence extraction module was developed and used as the front end of the summarization system.

We are preparing the task-based evaluation of the overall system. We also plan to evaluate the portability of the system by testing it on another corpus. We will also extend the system to query-based summarization and investigate whether the system can be modified for multiple document summarization.

Acknowledgment

We thank IBM for licensing us the ESG parser and the MITRE corporation for licensing us the coreference resolution system. This material is based upon work supported by the National Science Foundation under Grant No. IRI 96-19124 and IRI 96-18797. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

ANSI. 1997. Guidelines for abstracts. Technical Report Z39.14-1997, NISO Press, Bethesda, Maryland.

L. Baum. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process. *Inequalities*, (3):1–8.

Edward T. Cremmins. 1982. *The Art of Abstracting.* ISI Press, Philadelphia.

Brigitte Endres-Niggemeyer, Kai Haseloh, Jens Müller, Simone Peist, Irene Santini de Sigel, Alexander Sigel, Elisabeth Wansorra, Jan Wheeler, and Brünja Wollny. 1998. *Summarizing Information.* Springer, Berlin.

Raya Fidel. 1986. Writing abstracts for free-text searching. *Journal of Documentation*, 42(1):11–21, March.

Hongyan Jing and Kathleen R. McKeown. 1998. Combining multiple, large-scale resources in a reusable lexicon for natural language generation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, volume *1*, pages 607–613, Université de Montréal, Quebec, Canada, August.

Hongyan Jing and Kathleen R. McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR'99)*, pages 129–136, University of Berkeley, CA, August.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of ANLP 2000.*

Aravind.K. Joshi. 1987. Introduction to tree-adjoining grammars. In A. Manaster-Ramis, editor, *Mathematics of Language.* John Benjamins, Amsterdam.

Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Leo Obrst, Therese Firmin, Michael Chrzanowski, and Beth Sundheim. 1998. The TIPSTER SUMMAC text summarization evaluation final report. Technical Report MTR 98W0000138, The MITRE Corporation.

Inderjeet Mani, Barbara Gates, and Eric Bloedorn. 1999. Improving summaries by revising them. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics(ACL'99)*, pages 558–565, University of Maryland, Maryland, June.

Michael McCord, 1990. *English Slot Grammar.* IBM.

Samuel Thurber, editor. 1924. *Précis Writing for American Schools.* The Atlantic Monthly Press, INC., Boston.

A.J. Viterbi. 1967. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269.