In Proceedings of the Sixth International Workshop on Information Retrieval with Asian Language (IRAL 2003), Sapporo, Japan, July 7, 2003.

Improving Summarization Performance by Sentence Compression – A Pilot Study

Chin-Yew Lin
University of Southern California/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292, USA
cyl@isi.edu

Abstract

In this paper we study the effectiveness of applying sentence compression on an extraction based multi-document summarization system. Our results show that pure syntactic-based compression does not improve system performance. Topic signature-based reranking of compressed sentences does not help much either. However reranking using an oracle showed a significant improvement remains possible.

Keywords: *Text Summarization, Sentence Extraction, Sentence Compression, Evaluation.*

1 Introduction

The majority of systems participating in the past Document Understanding Conference (DUC, 2002) (a large scale summarization evaluation effort sponsored by the United States government), and the Text Summarization Challenge (Fukusima and Okumura, 2001) (sponsored by Japanese government) are extraction based. Extraction-based automatic text summarization systems extract parts of original documents and output the results as summaries (Chen et al., 2003; Edmundson, 1969; Goldstein et al., 1999; Hovy and Lin, 1999; Kupiec et al., 1995; Luhn, 1969). Other systems based on information extraction (McKeown et al., 2002; Radev and McKeown, 1998; White et al., 2001) and discourse analysis (Marcu, 1999; Strzalkowski et al., 1999) also exist but they are not yet usable for general-domain summarization. Our study focuses on the effectiveness of applying sentence compression techniques to improve the performance of extraction-based automatic text summarization systems.

Sentence compression aims to retain the most salient information of a sentence, rewritten in a short form (Knight and Marcu, 2000). It can be used to deliver compressed content to portable devices (Buyukkokten et al., 2001; Corston-Oliver, 2001) or as a reading aid for aphasic readers (Carroll et al., 1998) or the blind (Grefenstette, 1998). Earlier research in sentence compression focused on compressing single sentences, and were evaluated on a sentence by sentence basis. For example, Jing (2000) trained her system on a set of 500 sentences from the Benton Foundation (http://www.benton.org) and their reduced forms written by humans. The results were evaluated at the parse tree level against the reduced trees; while Knight and Marcu (2000) trained their system on a set of 1,067 sentences from Ziff-Davis magazine articles and evaluated their results on grammaticality and importance rated by humans. Both reported success in their evaluation criteria. However, neither of them reported their techniques' effectiveness in improving the overall performance of automatic text summarization systems. The goal of this pilot study is set to answer this question and provide a guideline for future research.

Section 2 gives an overview of Knight and Marcu's sentence compression algorithm that we used to compressed summary sentences. Section 3 describes the multi-document summarization system, NeATS, which was used as our testbed. Section 4 introduces a recall-based unigram co-occurrence automatic evaluation metric. Section 5 presents the experimental design. Section 6 shows the empirical results. Section 7 concludes this paper and discusses future directions.

2 A Noisy-Channel Model for Sentence Compression

Knight and Marcu (K&M) (2000) introduced two sentence compression algorithms, one based on the noisy-channel model and the other decision-based. We use the noisy-channel model in our experiments since it is able to generate a list of ranked candidates, while the decision-based is not.

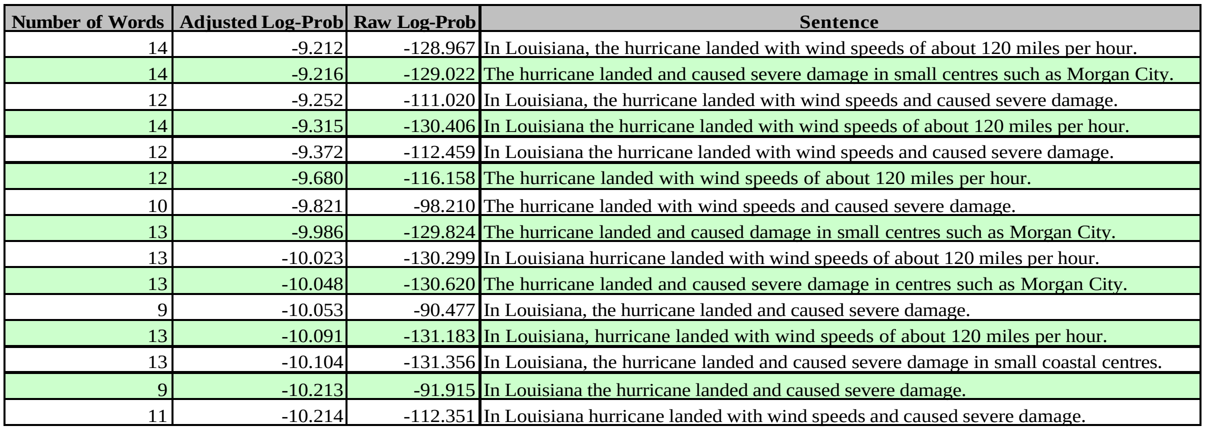
- Source model $P(s)$ – The compressed sentence language model. This would assign low probability to short sentences with undesirable features, for example, ungrammatical or too short.
- Channel model $P(t \mid s)$ – Given a compressed sentence s, the channel model assigns the probability of an original sentence, t, which could have been generated by s.
- Decoder – Given the original sentence t, find the best short sentence s generated from t, i.e. maximizing $P(s \mid t)$. This is equivalent to maximizing $P(t \mid s) \cdot P(s)$.

We used K&M's sentence compression algorithm as it was and did not retrain on new corpus. We also adopted the compression length-adjusted log probability to avoid the tendency of selecting very short compressions. Figure 1 shows a list of compressions for the sentence "*In Louisiana, the hurricane landed with wind speeds of about 120 miles per hour and caused severe damage in small coastal centres such as Morgan City, Franklin and New Iberia*." ranked according to their length-adjusted log-probability.

3 NeATS – a Multi-Document Summarization System

NeATS (Lin and Hovy, 2002) is an extraction-based multi-document summarization system. It is among the top two performers in DUC 2001 and 2002 (Over and Liggett, 2002). It consists of three main components:

- Content Selection – The goal of content selection is to identify important concepts mentioned in a document collection. NeATS computes the likelihood ratio λ (Dunning, 1993) to identify key concepts in unigrams, bigrams, and trigrams, and clusters these concepts in order to identify major subtopics within the main topic. Each sentence in the document set is then ranked, using the key concept structures. These n-gram key concepts are called topic signatures (Lin and Hovy 2000). We used key n-grams to rerank compressions in our experiments.
- Content Filtering – NeATS uses three different filters: sentence position, stigma words, and maximum marginal relevancy. Sentence position has been used as a good content filter since the late 60s (Edmundson, 1969). We apply a simple sentence filter that only retains the 10 lead sentences. Some sentences start with stigma words such as conjunctions, quotation marks, pronouns, and the verb "say" and its derivatives usually cause discontinuity in summaries. We simply reduce the scores of these sentences to demote their ranks and avoid including them in summaries of small sizes. To



Number of Words	Adjusted Log-Prob	Raw Log-Prob	Sentence
14	-9.212	-128.967	In Louisiana, the hurricane landed with wind speeds of about 120 miles per hour.
14	-9.216	-129.022	The hurricane landed and caused severe damage in small centres such as Morgan City.
12	-9.252	-111.020	In Louisiana, the hurricane landed with wind speeds and caused severe damage.
14	-9.315	-130.406	In Louisiana the hurricane landed with wind speeds of about 120 miles per hour.
12	-9.372	-112.459	In Louisiana the hurricane landed with wind speeds and caused severe damage.
12	-9.680	-116.158	The hurricane landed with wind speeds of about 120 miles per hour.
10	-9.821	-98.210	The hurricane landed with wind speeds and caused severe damage.
13	-9.986	-129.824	The hurricane landed and caused damage in small centres such as Morgan City.
13	-10.023	-130.299	In Louisiana hurricane landed with wind speeds of about 120 miles per hour.
13	-10.048	-130.620	The hurricane landed and caused severe damage in centres such as Morgan City.
9	-10.053	-90.477	In Louisiana, the hurricane landed and caused severe damage.
13	-10.091	-131.183	In Louisiana, hurricane landed with wind speeds of about 120 miles per hour.
13	-10.104	-131.356	In Louisiana, the hurricane landed and caused severe damage in small coastal centres.
9	-10.213	-91.915	In Louisiana the hurricane landed and caused severe damage.
11	-10.214	-112.351	In Louisiana hurricane landed with wind speeds and caused severe damage.

Figure 1. Top 15 compressions ranked by their adjusted log-probability for sentence "In Louisiana, the hurricane landed with wind speeds of about 120 miles per hour and caused severe damage in small coastal centres such as Morgan City, Franklin and New Iberia."

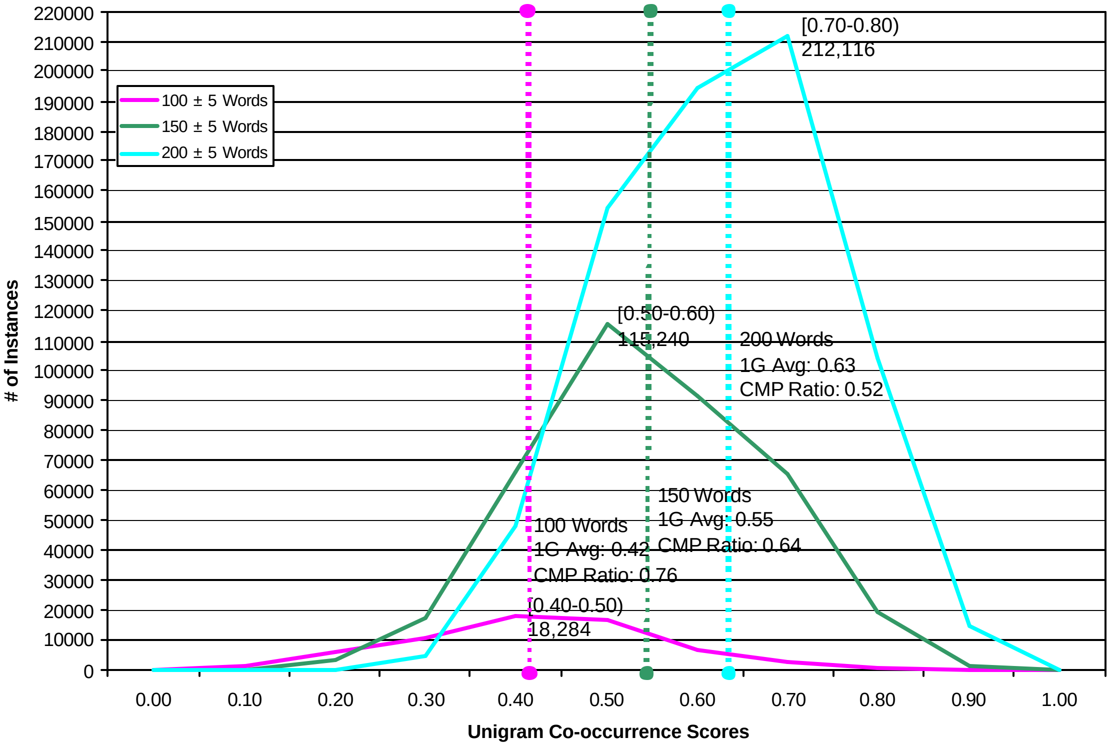


Figure 2. AP900424-0035 100, 150, and 200 words oracle extract instance distributions.

address the redundancy problem, we use a simplified version of CMU's MMR (Goldstein et al., 1999) algorithm. A sentence is added to the summary if and only if its content has less than *X* percent overlap with the summary.

- Content Presentation – To ensure coherence of the summary, NeATS pairs each sentence with an introduction sentence. It then outputs the final sentences in their chronological order.

We ran NeATS to generate summaries of different sizes that were used as our test bed. The topic signatures created in the process were used to rerank compressions. We describe the automatic evaluation metric used in our experiments in the next section.

4 Unigram Co-Occurrence Metric

In a recent study (Lin and Hovy, 2003a), we showed that the recall-based unigram co-occurrence automatic scoring metric correlates highly with human evaluation and has high recall and precision in predicting the statistical significance of results comparing with its human counterpart. The idea is to measure the content similarity between a system extract and a manual summary using simple n-gram overlap. A similar idea called IBM BLEU score has proved successful in automatic machine translation evaluation (NIST, 2002; Papineni et al., 2001). For summarization, we can express the degree of content overlap in terms of n-gram matches as the following equation:

$$C_n = \frac{\sum_{C \in \{Model\ Units\}} \sum_{n\text{-}gram \in C} Count_{match}(n\text{-}gram)}{\sum_{C \in \{Model\ Units\}} \sum_{n\text{-}gram \in C} Count(n\text{-}gram)} \qquad (1)$$

Model units are segments of manual summaries. They are typically either sentences or elementary discourse units as defined by Marcu (1999). $Count_{match}(n\text{-}gram)$ is the maximum number of *n-grams* co-occurring in a system extract and a model unit. *Count*(*n-gram*) is the number of *n-grams* in the model unit. Notice that the average n-gram coverage score, C_n, as shown in equation 1, is a recall-based metric, since the denominator of equation 1

```
<multi size="225" docset="d19d" org-size="227" comp-size="227">
Lawmakers clashed on 06/23/1988 over the question of counting illegal aliens in the 1990 Census, debating whether following the letter of the Constitution results in a system that is unfair to citizens. The forum was a Census subcommittee hearing on bills which would require the Census Bureau to figure out whether people are in the country legally and, if not, to delete them from the counts used in reapportioning seats in the House of Representatives. Simply put, the question was who should be counted as a person and who, if anybody, should not. The point at issue in Senate debate on a new immigration bill was whether illegal aliens should be counted in the process that will reallocate House seats among states after the 1990 census. The national head count will be taken April 1, 1990. In a blow to California and other states with large immigrant populations, the Senate voted on 09/29/1989 to bar the Census Bureau from counting illegal aliens in the 1990 population count. At stake are the number of seats in Congress for California, Florida, New York, Illinois, Pennsylvania and other states that will be reapportioned on the basis of next year's census. Federal aid to states also is frequently based on population counts, so millions of dollars in grants and other funds made available on a per capita basis would be affected.
</multi>
```

Figure 3. 227-word summary for topic D19 (“Aliens”).

```
<multi size="225" docset="d19d" org-size="227" comp-size="98">
Lawmakers clashed over question of counting illegal aliens Census debating whether results. Forum was a Census hearing, to delete them from the counts. Simply put question was who should be counted and who, if anybody, should not. Point at issue in debate on an immigration bill was whether illegal aliens should be counted. National count will be taken April 1, 1990. Senate voted to bar Census Bureau from counting illegal aliens. At stake are number of seats for California New York. Aid to states is frequently based on population counts, so millions would be affected.
</multi>
```

Figure 4. Compressed summary for topic D19 ("Aliens"), 98 words.

```
<DOC>
<TEXT>
<S SNTNO="1">Elizabeth Taylor battled pneumonia at her hospital, assisted by a ventilator, doctors say.</S>
<S SNTNO="2">Hospital officials described her condition late Monday as stabilizing after a lung biopsy to determine the cause of the pneumonia.</S>
<S SNTNO="3">Analysis of the tissue sample was expected to be complete by Thursday.</S>
<S SNTNO="4">Ms. Sam, spokeswoman said "it is serious, but they are really pleased with her progress.</S>
<S SNTNO="5">She's not well.</S>
<S SNTNO="6">She's not on her deathbed or anything.</S>
<S SNTNO="7">Another spokeswoman, Lisa Del Favaro, said Miss Taylor's family was at her bedside.</S>
<S SNTNO="8">During a nearly fatal bout with pneumonia in 1961, Miss Taylor underwent a tracheotomy to help her breathe.</S>
</TEXT>
```

Figure 5. A manual summary for document AP900424-0035.

is the sum total of the number of *n-gram*s occurring in the model summary instead of the system summary and only one model summary is used for each evaluation. In summary, the unigram co-occurrence statistics we use in the following sections are based on the following formula:

$$Ngram(i, j) = \exp\left(\sum_{n=i}^{j} w_n \log C_n\right) \quad (2)$$

Where $j \geq i$, i and j range from 1 to 4, and w_n is $1/(j-i+1)$. $Ngram(1, 4)$ is a weighted variable length *n*-gram match score similar to the IBM BLEU score; while $Ngram(k, k)$, i.e. $i = j = k$, is simply the average *k*-gram co-occurrence score C_k. In this study, we set $i = j = 1$, i.e. unigram co-occurrence score.

With an automatic scoring metric defined, we describe the experimental setup in the next section.

5 Experimental Designs

As stated in the introduction, we aim to investigate the effectiveness of sentence compression on overall system performance. If we can have a lossless compression function that compresses a given sentence to a minimal length and still retains the most important content of the sentence then we would be able to pack more information content into a fixed size summary. Figure 2 illustrates this effect

	Avg	Var	Std	AvgCR	VarCR	StdCR
KM	0.227	0.005	0.068	0.412	0.016	0.125
ORACLE	0.287	0.006	0.078	0.471	0.009	0.092
ORG	0.253	0.006	0.075	0.000	0.000	0.000
SIG	0.244	0.006	0.078	0.537	0.007	0.085
SIGKMa	0.242	0.006	0.077	0.370	0.015	0.123
SIGKMb	0.248	0.006	0.079	0.372	0.014	0.119

Table 1. Result table for six runs. Avg: mean unigram co-occurrence scores of 30 topics, Var: variance, Std: standard deviation, AvgCR: mean compression ratio, VarCR: variance of compression ratio, and StdCR: standard deviation of compression ratio.

Sentence Compression Z-Test (30 instances) Pairwise Observed Z-Score 95% (Size: 100)						
	KM	**ORACLE**	**ORG**	**SIG**	**SIGKMa**	**SIGKMb**
KM	-	-17.123	-7.681	-4.975	-4.474	-6.199
ORACLE		-	9.237	11.39	11.98	10.181
ORG			-	2.411	2.949	1.208
SIG				-	0.508	-1.168
SIGKMa					-	-1.682
SIGKMb						-

Table 2. Pairwise Z-test for six runs shown in Table 1 ($\alpha = 5\%$). Light gray (green) indicates runs on the column that are significantly better than runs on the row; dark gray indicates significantly worse.

graphically. For document AP900424-0035, which consists of 23 sentences or 417 words, we generate the full permutation set of sentence extracts, i.e., all possible 100±5, 150±5, and 200±5 words extracts. The 100±5 words extract at average compression ratio of 0.76 has most of its unigram co-occurrence score instances (18,284/61,762 ˜ 30%) falling within the interval between 0.40 and 0.50, i.e., the expected performance of an extraction-based system would be between 0.40 and 0.50. The 150±5 words extract at lower compression ratio of 064 has most of its instances between 0.50 and 0.60 (115,240/377,933 ˜ 30%) and the 200±5 words extract at compression ratio of 0.52 has most of its instances between 0.70 and 0.80 (212,116/731,819 ˜ 29%). If we can compress 150 or 200-word summaries into 100 words and retain their important content, we would be to achieve an average 30% to 50% increase in performance.

The question is: can an off-the-shelf sentence compression algorithm such as K&M's noisy-channel model achieve this? If the answer is yes, then how much performance gain can be achieved? If not, are there other ways to use sentence compression to improve system performance? To improve system performance? To answer these questions, we conduct the following experiments over 30 DUC 2001 topic sets:

(1) Run NeATS through the 30 DUC 2001 topic sets and generate summaries of size: 100, 120, 125, 130, 140, 150, 160, 175, 200, 225, 250, 275, 300, 325, 350, 375, and 400.

(2) Run K&M's sentence compression algorithm over all summary sentences (run KM). For each summary sentence, we have a set of candidate compressions. See Figure 1 for example.

(3) Rerank each candidate compression set using different scoring methods:

   a. Rerank each candidate compression set using topic signatures (run SIG).
   b. Rerank each candidate compression set using combination of KM and SIG scores using linear interpolation of topic signature score (SIG) and K&M's log-probability score (KM). We use the following formula in this experiment:

SIGKM=?·SIG + (1-?)·KM

? is set to 2/3 (run SIGKMa).

c. Rerank each candidate compression set using SIG score first and then KM is used to break ties (run SIGKMb).

d. Rerank each candidate compression set using unigram co-occurrence score against manual references. This gives the upper bound for the K&M's algorithm applied to the output generated by NeATS (run ORACLE).

(4) Select the best compression combination. For a given length constraint, for example 100 words, we produce the final result by selecting a compressed summary across different summary sizes for each topic that fits the length limit (<= 100±5 words), and output them as the final summary. For example, we found that a 227-word summary for topic D19 could be compressed to 98 words using the topic signature reranking method. The compressed summary would then be selected as the final summary for topic D19. Figure 3 shows the original 227-word summary and Figure 4 shows its compressed version.

There were 30 test topics in DUC 2001 and each topic contained about 10 documents. For each topic, four summaries of approximately 50, 100, 200, and 400 words were created manually as the 'ideal' model summaries. We used the set of 100-word manual summaries as our references in our experiments. An example manual summary is shown in Figure 5. We report results of these experiments in the next section.

6 Results

Tables 1 and 2 summarize the results. Analyzing all runs according to these two tables, we made the following observations.

(1) Selecting compressed sentences using length-adjusted scores (K&M) without any modification performed significantly worse (at $\alpha = 5\%$, table cells marked in dark gray in Table 2) than all other runs. This indicates we cannot rely on pure syntactic-based compression to improve overall system performance although the compression algorithm performed well in the individual sentence level.

(2) The original run (ORG) achieved an average unigram co-occurrence score of 0.253 and was significantly better than all other runs except the ORACLE and SIGKMb runs. This result was a little bit discouraging; it means that no/most reranking is not useful, and indicates that we need to invest more time in finding a better way to rank the compressed sentences. Pure syntactic (noisy-channel model), shallow semantic (by topic signatures), or simple combinations of them did not improve system performance and in some cases even degraded it.

(3) Comparing the ORACLE (0.287) run with the average human performance of 0.270 (not shown in the Tables), we should remain optimistic about finding a better ranking algorithm to select the best compression. However, the low human performance posts a challenge for machine learning algorithms to learn this function. We provided more in-depth discussion of this issue in other papers (Lin and Hovy, 2002; Lin and Hovy 2003b).

(4) That the ORACLE run did not achieve higher score also implied the following:

a. The sentence compression algorithm that we used might drop some important content. Therefore the compressed summaries did not achieve 20% increase in performance as Figure 1 might suggest when systems were allowed to output 100% longer summary than the given constraint (i.e. if a 100-word summary is requested, a system can provide a 200-word summary in response.)

b. The way we generated our compressed summaries was not effective. We might need to optimize and select compressions according to a global optimization function. For example, if some important content is mentioned in sentences already included in a summary, we would want to take this into account and to add compressions with new information to the final summary.

7 Conclusions

In this paper we presented an empirical study of the effectiveness of applying sentence compression to improve summarization performance. We used a good sentence compression algorithm, compared the performance of five different ranking algorithms, and found that pure a-sentence-at-a-time syntactic or shallow semantic-based reranking was not enough to boost system performance. However, the significant difference between the ORACLE run and the original run (ORG) indicated there is potential in sentence compression but we need to find a better compression selection function that should take into account global cross-sentence optimization. This indicated local optimization at the sentence level such as Knight and Marcu's (2000) noisy-channel model is not enough when our goal is to find the best compressed summaries not the best compressed sentences. In the future, we would like to apply a similar methodology to different text units, for example, sub-sentence units such as elementary discourse unit (Marcu, 1999) and a larger corpus, for example, DUC 2002 and DUC 2003. We want to explore compression techniques to go beyond simple sentence extraction.

References

O. Buyukkokten, H. Garcia-Molina, A. Paepcke. 2001. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. The 10th International WWW Conference (WWW10). Hong Kong, China.

J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait. 1998. Practical Simplification of English Newspaper Text to Assist Aphasic Readers. In Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology, Madison, WI, USA.

H.H. Chen, J.J. Kuo, and T.C. Su 2003. Clustering and Visualization in a Multi-Lingual Multi-Document Summarization System. In Proceedings of 25th European Conference on Information Retrieval Research, Lecture Note in Computer Science, April 14-16, Pisa, Italy.

S. Corston-Oliver. 2001. Text Compaction for Display on Very Small Screens. In Proceedings of the Workshop on Automatic Summarization (WAS 2001), Pittsburgh, PA, USA.

DUC. 2002. The Document Understanding Conference. http://duc.nist.gov.

T. Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19, 61–74.

H.P. Edmundson. 1969. New Methods in Automatic Abstracting. Journal of the Association for Computing Machinery. 16(2).

T. Fukusima and M. Okumura. 2001. Text Summarization Challenge Text Summarization Evaluation in Japan. In Proceedings of the Workshop on Automatic Summarization (WAS 2001), Pittsburgh, PA, USA.

J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. 1999. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval (SIGIR-99), Berkeley, CA, USA, 121–128.

G. Grefenstette. 1998. Producing Intelligent Telegraphic Text Reduction to Provide an Audio Scanning Service for the Blind. In Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization, Stanford University, CA, USA, 111–118.

E. Hovy and C.-Y. Lin. 1999. Automatic Text Summarization in SUMMARIST. In I. Mani and M. Maybury (eds), Advances in Automatic Text Summarization, 81–94. MIT Press.

H. Jing. 2000. Sentence simplification in automatic text summarization. In the Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'00). Seattle, Washington, USA.

K. Knight and D. Marcu. 2000. Statistics-Based Summarization – Step One: Sentence Compression. In Proceedings of AAAI-2000, Austin, TX, USA.

J. Kupiec, J. Pederson, and F. Chen. 1995. A Trainable Document Summarizer. In Proceedings of the 18th International ACM Conference on Research and Development in Information Retrieval (SIGIR-95), Seattle, WA, USA, 68–73.

C.-Y. Lin and E. Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), Saarbrücken, Germany.

C.-Y. Lin and E. Hovy. 2002. From Single to Multi-document Summarization: A Prototype System and its Evaluation. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-2002), Philadelphia, PA, U.S.A.

C.-Y. Lin and E. Hovy. 2002. Manual and Automatic Evaluations of Summaries. In Proceedings of the Workshop on Automatic Summarization, post-conference workshop of ACL-2002, pp. 45-51, Philadelphia, PA, USA.

C.-Y. Lin and E. Hovy. 2003a. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Human Language Technology Conference* (HLT-NAACL 2003), Edmonton, Canada.

C.-Y. Lin and E. Hovy 2003b. The Potential and Limitations of Sentence Extraction for Summarization. In Proceedings of the Workshop on Automatic Summarization post-conference workshop of HLT-NAACL-2003, Edmonton, Canada.

H.P. Luhn. 1969. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development. 2(2).

D. Marcu. 1999. Discourse trees are good indicators of importance in text. In I. Mani and M. Maybury (eds), Advances in Automatic Text Summarization, 123–136. MIT Press.

K. McKeown, Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, A. Nenkova, C. Sable, B. Schiffman, S. Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In Proceedings of Human Language Technology Conference 2002 (HLT 2002). San Diego, CA, USA.

NIST. 2002. Automatic Evaluation of Machine Translation Quality using N-gram Co-Occurrence Statistics.

P. Over and W. Liggett. 2002. Introduction to DUC-2002: an Intrinsic Evaluation of Generic News Text Summarization Systems. In Proceedings of Workshop on Automatic Summarization (DUC 2002), Philadelphia, PA, USA. http://www-nlpir.nist.gov/projects/duc/pubs/2002slides/overview.02.pdf

K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. IBM Research Report RC22176 (W0109-022).

D.R. Radev and K.R. McKeown. 1998. Generating Natural Language Summaries from Multiple On-line Sources. Computational Linguistics, 24(3):469–500.

T. Strzalkowski, G. Stein, J. Wang, and B, Wise. A Robust Practical Text Summarizer. 1999. In I. Mani and M. Maybury (eds), Advances in Automatic Text Summarization, 137–154. MIT Press.

M. White, T. Korelsky, C. Cardie, V. Ng, D. Pierce, and K. Wagstaff. 2001. Multidocument Summarization via Information Extraction. In Proceedings of Human Language Technology Conference 2001 (HLT 2001), San Diego, CA, USA.