Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation

Bonnie Dorr, David Zajic
University of Maryland
bonnie, dmzajic@umiacs.umd.edu

Richard Schwartz
BBN
schwartz@bbn.com

Abstract

This paper presents Hedge Trimmer, a HEaDline GEneration system that creates a headline for a newspaper story using linguistically-motivated heuristics to guide the choice of a potential headline. We present feasibility tests used to establish the validity of an approach that constructs a headline by selecting words in order from a story. In addition, we describe experimental results that demonstrate the effectiveness of our linguistically-motivated approach over a HMM-based model, using both human evaluation and automatic metrics for comparing the two approaches.

1 Introduction

In this paper we present Hedge Trimmer, a HEaDline GEneration system that creates a headline for a newspaper story by removing constituents from a parse tree of the first sentence until a length threshold has been reached. Linguistically-motivated heuristics guide the choice of which constituents of a story should be preserved, and which ones should be deleted. Our focus is on headline generation for English newspaper texts, with an eye toward the production of document surrogates—for cross-language information retrieval—and the eventual generation of readable headlines from speech broadcasts.

In contrast to original newspaper headlines, which are often intended only to catch the eye, our approach produces *informative* abstracts describing the main theme or event of the newspaper article. We claim that the construction of informative abstracts requires access to deeper linguistic knowledge, in order to make substantial improvements over purely statistical approaches.

In this paper, we present our technique for producing headlines using a parse-and-trim approach based on the BBN Parser. As described in Miller et al. (1998), the BBN parser builds augmented parse trees according to a process similar to that described in Collins (1997). The BBN parser has been used successfully for the task of information extraction in the SIFT system (Miller et al., 2000).

The next section presents previous work in the area of automatic generation of abstracts. Following this, we present feasibility tests used to establish the validity of an approach that constructs headlines from words in a story, taken in order and focusing on the earlier part of the story. Next, we describe the application of the parse-and-trim approach to the problem of headline generation. We discuss the linguistically-motivated heuristics we use to produce results that are headline-like. Finally, we evaluate Hedge Trimmer by comparing it to our earlier work on headline generation, a probabilistic model for automatic headline generation (Zajic et al, 2002). In this paper we will refer to this statistical system as HMM Hedge We demonstrate the effectiveness of our linguistically-motivated approach, Hedge Trimmer, over the probabilistic model, HMM Hedge, using both human evaluation and automatic metrics.

2 Previous Work

Other researchers have investigated the topic of automatic generation of abstracts, but the focus has been different, e.g., sentence extraction (Edmundson, 1969; Johnson et al, 1993; Kupiec et al., 1995; Mann et al., 1992; Teufel and Moens, 1997; Zechner, 1995), processing of structured templates (Paice and Jones, 1993), sentence compression (Hori et al., 2002; Knight and Marcu, 2001; Grefenstette, 1998, Luhn, 1958), and generation of abstracts from multiple sources (Radev and McKeown, 1998). We focus instead on the construction of headline-style abstracts from a single *story*.

Headline generation can be viewed as analogous to statistical machine translation, where a concise document is generated from a verbose one using a Noisy Channel Model and the Viterbi search to select the most likely summarization. This approach has been explored in (Zajic et al., 2002) and (Banko et al., 2000).

The approach we use in Hedge is most similar to that of (Knight and Marcu, 2001), where a single sentence is shortened using statistical compression. As in this work, we select headline words from story words in the order that they appear in the story—in particular, the first sentence of the story. However, we use linguisti-

cally motivated heuristics for shortening the sentence; there is no statistical model, which means we do not require any prior training on a large corpus of story/headline pairs.

Linguistically motivated heuristics have been used by (McKeown et al, 2002) to distinguish constituents of parse trees which can be removed without affecting grammaticality or correctness. GLEANS (Daumé et al, 2002) uses parsing and named entity tagging to fill values in headline templates.

Consider the following excerpt from a news story:

(1) Story Words: **Kurdish guerilla forces** *moving with lightning speed* **poured into Kirkuk** *today immediately* **after Iraqi troops**, *fleeing relentless U.S. airstrikes,* **abandoned** *the hub of Iraq's rich northern* **oil fields**.

Generated Headline: Kurdish guerilla forces poured into Kirkuk after Iraqi troops abandoned oil fields.

In this case, the words in bold form a fluent and accurate headline for the story. Italicized words are deleted based on information provided in a parse-tree representation of the sentence.

3   Feasibility Testing

Our approach is based on the selection of words from the original story, in the order that they appear in the story, and allowing for morphological variation. To determine the feasibility of our headline-generation approach, we first attempted to apply our "select-words-in-order" technique by hand. We asked two subjects to write headline headlines for 73 AP stories from the TIPSTER corpus for January 1, 1989, by selecting words in order from the story. Of the 146 headlines, 2 did not meet the "select-words-in-order" criteria because of accidental word reordering. We found that at least one fluent and accurate headline meeting the criteria was created for each of the stories. The average length of the headlines was 10.76 words.

Later we examined the distribution of the headline words among the sentences of the stories, i.e. how many came from the first sentence of a story, how many from the second sentence, etc. The results of this study are shown in Figure 1. We observe that 86.8% of the headline words were chosen from the first sentence of their stories. We performed a subsequent study in which two subjects created 100 headlines for 100 AP stories from August 6, 1990. 51.4% of the headline words in the second set were chosen from the first sentence. The distribution of headline words for the second set shown in Figure 2.

Although humans do not always select headline words from the first sentence, we observe that a large percentage of headline words are often found in the first sentence.
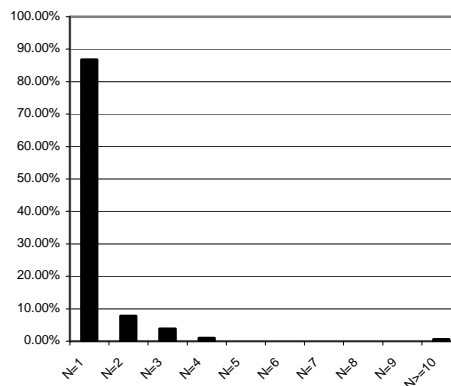

Figure 1: Percentage of words from human-generated headlines drawn from Nth sentence of story (Set 1)
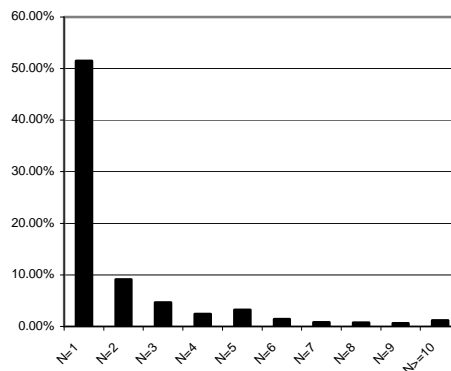

Figure 2: Percentage of words from human-generated headlines drawn from Nth sentence of story (Set 2)

4   Approach

The input to Hedge is a story, whose first sentence is immediately passed through the BBN parser. The parse-tree result serves as input to a linguistically-motivated module that selects story words to form headlines based on key insights gained from our observations of human-constructed headlines. That is, we conducted a human inspection of the 73 TIPSTER stories mentioned in Section 3 for the purpose of developing the Hedge Trimmer algorithm.

Based on our observations of human-produced headlines, we developed the following algorithm for parse-tree trimming:

1. Choose lowest leftmost S with NP,VP
2. Remove low content units

○　some determiners
　　　　○　time expressions
　　3.　Iterative shortening:
　　　　○　XP Reduction
　　　　○　Remove preposed adjuncts
　　　　○　Remove trailing PPs
　　　　○　Remove trailing SBARs

More recently, we conducted an automatic analysis of the human-generated headlines that supports several of the insights gleaned from this initial study. We parsed 218 human-produced headlines using the BBN parser and analyzed the results. For this analysis, we used 72 headlines produced by a third participant.[1] The parsing results included 957 noun phrases (NP) and 315 clauses (S).

We calculated percentages based on headline-level, NP-level, and Sentence-level structures in the parsing results. That is, we counted:

- The percentage of the 957 NPs containing determiners and relative clauses
- The percentage of the 218 headlines containing preposed adjuncts and conjoined S or VPs
- The percentage of the 315 S nodes containing trailing time expressions, SBARs, and PPs

Figure 3 summarizes the results of this automatic analysis. In our initial human inspection, we considered each of these categories to be reasonable candidates for deletion in our parse tree and this automatic analysis indicates that we have made reasonable choices for deletion, with the possible exception of trailing PPs, which show up in over half of the human-generated headlines. This suggests that we should proceed with caution with respect to the deletion of trailing PPs; thus we consider this to be an option only if no other is available.

HEADLINE-LEVEL PERCENTAGES
preposed adjuncts = 0/218 (0%)
conjoined S = 1/218 (.5%)
conjoined VP = 7/218 (3%)
NP-LEVEL PERCENTAGES
relative clauses = 3/957 (.3%)
determiners = 31/957 (3%); of these, only 16 were "a" or "the" (1.6% overall)
S-LEVEL PERCENTAGES[2]
time expressions = 5/315 (1.5%)
trailing PPs = 165/315 (52%)
trailing SBARs = 24/315 (8%)

Figure 3: Percentages found in human-generated headlines

---

[1] No response was given for one of the 73 stories.
[2] Trailing constituents (SBARs and PPs) are computed by counting the number of SBARs (or PPs) not designated as an argument of (contained in) a verb phrase.

For a comparison, we conducted a second analysis in which we used the same parser on just the first sentence of each of the 73 stories. In this second analysis, the parsing results included 817 noun phrases (NP) and 316 clauses (S). A summary of these results is shown in Figure 4. Note that, across the board, the percentages are higher in this analysis than in the results shown in Figure 3 (ranging from 12% higher—in the case of trailing PPs—to 1500% higher in the case of time expressions), indicating that our choices of deletion in the Hedge Trimmer algorithm are well-grounded.

HEADLINE-LEVEL PERCENTAGES
preposed adjuncts = 2/73 (2.7%)
conjoined S = 3/73 (4%)
conjoined VP = 20/73 (27%)
NP-LEVEL PERCENTAGES
relative clauses = 29/817 (3.5%)
determiners = 205/817 (25%); of these, only 171 were "a" or "the" (21% overall)
S-LEVEL PERCENTAGES
time expressions = 77/316 (24%)
trailing PPs = 184/316 (58%)
trailing SBARs = 49/316 (16%)

Figure 4: Percentages found in first sentence of each story.

4.1　Choose the Correct S Node

The first step relies on what is referred to as the *Projection Principle* in linguistic theory (Chomsky, 1981): Predicates project a subject (both dominated by S) in the surface structure. Our human-generated headlines always conformed to this rule; thus, we adopted it as a constraint in our algorithm.

An example of the application of step 1 above is the following, where boldfaced material from the parse tree representation is retained and italicized material is eliminated:

(2)　Input: Rebels agree to talks with government officials said Tuesday.

　　　Parse: *[S* **[S [NP Rebels] [VP agree to talks with government]]** *officials said Tuesday.]*

　　　Output of step 1: Rebels agree to talks with government.

When the parser produces a correct tree, this step provides a grammatical headline. However, the parser often produces an incorrect output. Human inspection of our 624-sentence DUC-2003 evaluation set revealed that there were two such scenarios, illustrated by the following cases:

(3) **[S [SBAR What started as a local controversy] [VP has evolved into an international scandal.]]**

(4) **[NP [NP Bangladesh] [CC and] [NP [NP India] [VP signed a water sharing accord.]]]**

In the first case, an S exists, but it does not conform to the requirements of step 1. This occurred in 2.6% of the sentences in the DUC-2003 evaluation data. We resolve this by selecting the lowest leftmost S, i.e., the entire string "What started as a local controversy has evolved into an international scandal" in the example above.

In the second case, there is no S available. This occurred in 3.4% of the sentences in the evaluation data. We resolve this by selecting the root of the parse tree; this would be the entire string "Bangladesh and India signed a water sharing accord" above. No other parser errors were encountered in the DUC-2003 evaluation data.

4.2 Removal of Low Content Nodes

Step 2 of our algorithm eliminates low-content units. We start with the simplest low-content units: the determiners *a* and *the*. Other determiners were not considered for deletion because our analysis of the human-constructed headlines revealed that most of the other determiners provide important information, e.g., negation (not), quantifiers (each, many, several), and deictics (this, that).

Beyond these, we found that the human-generated headlines contained very few time expressions which, although certainly not content-free, do not contribute toward conveying the overall "who/what content" of the story. Since our goal is to provide an informative headline (i.e., the action and its participants), the identification and elimination of time expressions provided a significant boost in the performance of our automatic headline generator.

We identified time expressions in the stories using BBN's IdentiFinder™ (Bikel et al, 1999). We implemented the elimination of time expressions as a two-step process:

- Use IdentiFinder to mark time expressions
- Remove [PP … [NP [X] …] …] and [NP [X]] where X is tagged as part of a time expression

The following examples illustrate the application of this step:

(5) Input: The State Department on Friday lifted the ban it had imposed on foreign fliers.

Parse: *[Det The]* **State Department** *[PP [IN on] [NP [NNP Friday]]]* **lifted** *[Det the]* **ban it had imposed on foreign fliers.**

Output of step 2: State Department lifted ban it has imposed on foreign fliers.

(6) Input: An international relief agency announced Wednesday that it is withdrawing from North Korea.

Parse: *[Det An]* **international relief agency announced** *[NP [NNP Wednesday]]* **that it is withdrawing from North Korea.**

Output of step 2: International relief agency announced that it is withdrawing from North Korea.

We found that 53.2% of the stories we examined contained at least one time expression which could be deleted. Human inspection of the 50 deleted time expressions showed that 38 were desirable deletions, 10 were locally undesirable because they introduced an ungrammatical fragment,[3] and 2 were undesirable because they removed a potentially relevant constituent. However, even an undesirable deletion often pans out for two reasons: (1) the ungrammatical fragment is frequently deleted later by some other rule; and (2) every time a constituent is removed it makes room under the threshold for some other, possibly more relevant constituent. Consider the following examples.

(7) At least two people were killed Sunday.

(8) At least two people were killed when single-engine airplane crashed.

Example (7) was produced by a system which did not remove time expressions. Example (8) shows that if the time expression Sunday were removed, it would make room below the 10-word threshold for another important piece of information.

4.3 Iterative Shortening

The final step, iterative shortening, removes linguistically peripheral material—through successive deletions—until the sentence is shorter than a given threshold. We took the threshold to be 10 for the DUC task, but it is a configurable parameter. Also, given that the human-generated headlines tended to retain earlier material more often than later material, much of our

---

[3] Two examples of genuinely undesirable time expression deletion are:
- The attack came on the heels of [New Year's Day].
- [New Year's Day] brought a foot of snow to the region.

iterative shortening is focused on deleting the rightmost phrasal categories until the length is below threshold.

There are four types of iterative shortening rules. The first type is a rule we call "XP-over-XP," which is implemented as follows:

> In constructions of the form [XP [XP …] …] remove the other children of the higher XP, where XP is NP, VP or S.

This is a linguistic generalization that allowed us apply a single rule to capture three different phenomena (relative clauses, verb-phrase conjunction, and sentential conjunction). The rule is applied iteratively, from the deepest rightmost applicable node backwards, until the length threshold is reached.

The impact of XP-over-XP can be seen in these examples of NP-over-NP (relative clauses), VP-over-VP (verb-phrase conjunction), and S-over-S (sentential conjunction), respectively:

(9) Input: A fire killed a firefighter who was fatally injured as he searched the house.

Parse: **[S** *[Det A]* **fire killed** *[Det a]* **[NP [NP firefighter]** *[SBAR who was fatally injured as he searched the house]* **]]**

Output of NP-over-NP: fire killed firefighter

(10) Input: Illegal fireworks injured hundreds of people and started six fires.

Parse: **[S Illegal fireworks [VP [VP injured hundreds of people]** *[CC and] [VP started six fires]* **]]**

Output of VP-over-VP: Illegal fireworks injured hundreds of people

(11) Input: A company offering blood cholesterol tests in grocery stores says medical technology has outpaced state laws, but the state says the company doesn't have the proper licenses.

Parse: **[S** *[Det A]* **company offering blood cholesterol tests in grocery stores says [S [S medical technology has outpaced state laws],** *[CC but] [S [Det the] state stays [Det the] company doesn't have [Det the] proper licenses.]]* **]**

Output of S-over-S: Company offering blood cholesterol tests in grocery store says medical technology has outpaced state laws

The second type of iterative shortening is the removal of preposed adjuncts. The motivation for this type of shortening is that **all** of the human-generated headlines ignored what we refer to as the *preamble* of the story. Assuming the Projection principle has been satisfied, the preamble is viewed as the phrasal material occurring before the subject of the sentence. Thus, adjuncts are identified linguistically as any XP unit preceding the first NP (the subject) under the S chosen by step 1. This type of phrasal modifier is invisible to the XP-over-XP rule, which deletes material under a node only if it dominates another node of the same phrasal category.

The impact of this type of shortening can be seen in the following example:

(12) Input: According to a now finalized blueprint described by U.S. officials and other sources, the Bush administration plans to take complete, unilateral control of a post-Saddam Hussein Iraq

Parse: **[S** *[PP According to a now-finalized blueprint described by U.S. officials and other sources]* *[Det the]* **Bush administration plans to take complete, unilateral control of** *[Det a]* **post-Saddam Hussein Iraq]**

Output of Preposed Adjunct Removal: Bush administration plans to take complete unilateral control of post-Saddam Hussein Iraq

The third and fourth types of iterative shortening are the removal of trailing PPs and SBARs, respectively:

- Remove PPs from deepest rightmost node backward until length is below threshold.
- Remove SBARs from deepest rightmost node backward until length is below threshold.

These are the riskiest of the iterative shortening rules, as indicated in our analysis of the human-generated headlines. Thus, we apply these conservatively, only when there are no other categories of rules to apply. Moreover, these rules are applied with a backoff option to avoid over-trimming the parse tree. First the PP shortening rule is applied. If the threshold has been reached, no more shortening is done. However, if the threshold has not been reached, the system reverts to the parse tree as it was before any PPs were removed, and applies the SBAR shortening rule. If the threshold still has not been reached, the PP rule is applied to the result of the SBAR rule.

Other sequences of shortening rules are possible. The one above was observed to produce the best results on a 73-sentence development set of stories from the

TIPSTER corpus. The intuition is that, when removing constituents from a parse tree, it's best to remove smaller portions during each iteration, to avoid producing trees with undesirably few words. PPs tend to represent small parts of the tree while SBARs represent large parts of the tree. Thus we try to reach the threshold by removing small constituents, but if we can't reach the threshold that way, we restore the small constituents, remove a large constituent and resume the deletion of small constituents.

The impact of these two types of shortening can be seen in the following examples:

(13) Input: More oil-covered sea birds were found over the weekend.

Parse: [S **More oil-covered sea birds were found** *[PP over the weekend]*]

Output of PP Removal: More oil-covered sea birds were found.

(14) Input: Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition while assuring closer cooperation after Hong Kong returns.

Parse: [S **Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition** *[SBAR while assuring closer cooperation after Hong Kong returns]*]

Output of SBAR Removal: Visiting China Interpol chief expressed confidence in Hong Kong's smooth transition

5   Evaluation

We conducted two evaluations. One was an informal human assessment and one was a formal automatic evaluation.

5.1   HMM Hedge

We compared our current system to a statistical headline generation system we presented at the 2001 DUC Summarization Workshop (Zajic et al., 2002), which we will refer to as HMM Hedge. HMM Hedge treats the summarization problem as analogous to statistical machine translation. The verbose language, articles, is treated as the result of a concise language, headlines, being transmitted through a noisy channel. The result of the transmission is that extra words are added and some morphological variations occur. The Viterbi algorithm is used to calculate the most likely unseen headline to have generated the seen article. The Viterbi algorithm is biased to favor headline-like characteristics gleaned from observation of human performance of the headline-construction task. Since the 2002 Workshop, HMM Hedge has been enhanced by incorporating part of speech of information into the decoding process, rejecting headlines that do not contain a word that was used as a verb in the story, and allowing morphological variation only on words that were used as verbs in the story. HMM Hedge was trained on 700,000 news articles and headlines from the TIPSTER corpus.

5.2   Bleu: Automatic Evaluation

BLEU (Papineni et al, 2002) is a system for automatic evaluation of machine translation. BLEU uses a modified n-gram precision measure to compare machine translations to reference human translations. We treat summarization as a type of translation from a verbose language to a concise one, and compare automatically generated headlines to human generated headlines.

For this evaluation we used 100 headlines created for 100 AP stories from the TIPSTER collection for August 6, 1990 as reference summarizations for those stories. These 100 stories had never been run through either system or evaluated by the authors prior to this evaluation. We also used the 2496 manual abstracts for the DUC2003 10-word summarization task as reference translations for the 624 test documents of that task. We used two variants of HMM Hedge, one which selects headline words from the first 60 words of the story, and one which selects words from the first sentence of the story. Table 1 shows the BLEU score using trigrams, and the 95% confidence interval for the score.

	AP900806	DUC2003
HMM60	0.0997 ± 0.0322 avg len: 8.62	0.1050 ± 0.0154 avg len: 8.54
HMM1Sent	0.0998 ± 0.0354 avg len: 8.78	0.1115 ± 0.0173 avg len: 8.95
HedgeTr	0.1067 ± 0.0301 avg len: 8.27	0.1341 ± 0.0181 avg len: 8.50

Table 1

These results show that although Hedge Trimmer scores slightly higher than HMM Hedge on both data sets, the results are not statistically significant. However, we believe that the difference in the quality of the systems is not adequately reflected by this automatic evaluation.

5.3   Human Evaluation

Human evaluation indicates significantly higher scores than might be guessed from the automatic evaluation. For the 100 AP stories from the TIPSTER corpus for August 6, 1990, the output of Hedge Trim-

mer and HMM Hedge was evaluated by one human. Each headline was given a subjective score from 1 to 5, with 1 being the worst and 5 being the best. The average score of HMM Hedge was 3.01 with standard deviation of 1.11. The average score of Hedge Trimmer was 3.72 with standard deviation of 1.26. Using a t-score, the difference is significant with greater than 99.9% confidence.

The types of problems exhibited by the two systems are qualitatively different. The probabilistic system is more likely to produce an ungrammatical result or omit a necessary argument, as in the examples below.

(15) HMM60: Nearly drowns in satisfactory condition satisfactory condition.

(16) HMM60: A county jail inmate who noticed.

In contrast, the parser-based system is more likely to fail by producing a grammatical but semantically useless headline.

(17) HedgeTr: It may not be everyone's idea especially coming on heels.

Finally, even when both systems produce acceptable output, Hedge Trimmer usually produces headlines which are more fluent or include more useful information.

(18) a. HMM60: New Year's eve capsizing
b. HedgeTr: Sightseeing cruise boat capsized and sank.

(19) a. HMM60: hundreds of Tibetan students demonstrate in Lhasa.
b. HedgeTr: Hundreds demonstrated in Lhasa demanding that Chinese authorities respect culture.

6   Conclusions and Future Work

We have shown the effectiveness of constructing headlines by selecting words in order from a newspaper story. The practice of selecting words from the early part of the document has been justified by analyzing the behavior of humans doing the task, and by automatic evaluation of a system operating on a similar principle.

We have compared two systems that use this basic technique, one taking a statistical approach and the other a linguistic approach. The results of the linguistically motivated approach show that we can build a working system with minimal linguistic knowledge and circumvent the need for large amounts of training data. We should be able to quickly produce a comparable system for other languages, especially in light of current multi-lingual initiatives that include automatic parser induction for new languages, e.g. the TIDES initiative.

We plan to enhance Hedge Trimmer by using a language model of Headlinese, the language of newspaper headlines (Mårdh 1980) to guide the system in which constituents to remove. We Also we plan to allow for morphological variation in verbs to produce the present tense headlines typical of Headlinese.

Hedge Trimmer will be installed in a translingual detection system for enhanced display of document surrogates for cross-language question answering. This system will be evaluated in upcoming iCLEF conferences.

7   Acknowledgements

References

Banko, M., Mittal, V., Witbrock, M. (2000). Headline Generation Based on Statistical Translation. In *Proceedings of 38th Meeting of Association for Computation Linguistics*, Hong Kong, pp. 218-325.

Bikel, D., Schwartz, R., and Weischedel, R. (1999). An algorithm that learns what's in a name. *Machine Learning*, 34(1/3), February

Chomsky, Noam A. (1981). *Lectures on Government and Binding*, Foris Publications, Dordrecht, Holland.

Collins, M. (1997). Three generative lexicalised models for statistical parsing. In *Proceedings of the 35th ACL*, 1997.

Daumé, H., Echihabi, A., Marcu, D., Munteanu, D., Soricut, R. (2002). GLEANS: A Generator of Logical Extracts and Abstracts for Nice Summaries, In *Workshop on Automatic Summarization*, Philadelphia, PA, pp. 9-14.

Edmundson, H. (1969). "New methods in automatic extracting." *Journal of the ACM*, 16(2).

Grefenstett, G. (1998). Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Working Notes of the AIII Spring Symposium on Intelligent Text Summarization,* Stanford University, CA, pp. 111-118.

Hori, C., Furui, S., Malkin, R., Yu, H., Waibel, A. (2002). Automatic Speech Summarization Applied to English Broadcast News Speech. In *Proceedings of*

2002 International Conference on Acoustics, Speech and Signal Processing, Istanbul, pp. 9-12.

Johnson, F. C., Paice, C. D., Black, W. J., and Neal, A. P. (1993). "The application of linguistic processing to automatic abstract generation." *Journal of Document and Text Management*, 1(3):215-42.

Knight, K. and Marcu, D. (2001). "Statistics-Based Summarization Step One: Sentence Compression," In *Proceedings of AAAI-2001*.

Kupiec, J., Pedersen, J., and Chen, F. (1995). "A trainable document summarizer." In *Proceedings of the 18th ACM-SIGIR Conference*.

Luhn, H. P. (1958). "The automatic creation of literature abstracts." *IBM Journal of Research and Development*, 2(2).

Mann, W.C., Matthiesen, C.M.I.M., and Thomspson, S.A. (1992). Rhetorical structure theory and text analysis. In Mann, W.C. and Thompson, S.A., editors, Discourse Description. J. Benjamins Pub. Co., Amsterdam.

Mårdh, I. (1980). Headlinese: On the Grammar of English Front Page Headlines, Malmo.

McKeown, K., Barzilay, R., Blair-Goldensohn, S., Evans, D., Hatzivassiloglou, V., Klavans, J., Nenkova, A., Schiffman, B., and Sigelman, S. (2002). "The Columbia Multi-Document Summarizer for DUC 2002," In *Workshop on Automatic Summarization*, Philadelphia, PA, pp. 1-8.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R. and Annotation Group, the (1998). Algorithms that Learn to Extract Information; BBN: Description of the SIFT System as Used for MUC-7. In Proceedings of the MUC-7.

Miller, S., Ramshaw, L., Fox, H., and Weischedel, R. (2000). "A Novel Use of Statistical Parsing to Extract Information from Text," In *Proceedings of 1st Meeting of the North American Chapter of the ACL*, Seattle, WA, pp.226-233.

Paice, C. D. and Jones, A. P. (1993). "The identification of important concepts in highly structured technical papers." In *Proceedings of the Sixteenth Annual International ACM SIGIR conference on research and development in IR*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation," In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, PA, pp. 331-318

Radev, Dragomir R. and Kathleen R. McKeown (1998). "Generating Natural Language Summaries from Multiple On-Line Sources." *Computational Linguistics*, 24(3):469--500, September 1998.

Teufel, Simone and Marc Moens (1997). "Sentence extraction as a classification task," In *Proceedings of the Workshop on Intelligent and scalable Text summarization*, ACL/EACL-1997, Madrid, Spain.

Zajic, D., Dorr, B., Schwartz, R. (2002) "Automatic Headline Generation for Newspaper Stories," In *Workshop on Automatic Summarization*, Philadelphia, PA, pp. 78-85.

Zechner, K. (1995). "Automatic text abstracting by selecting relevant passages." Master's thesis, Centre for Cognitive Science, University of Edinburgh.