

# Multiple Alternative Sentence Compressions and Word-Pair Antonymy for Automatic Text Summarization and Recognizing Textual Entailment

Saif Mohammad, Bonnie J. Dorr, Melissa Egan, Nitin Madnani, David Zajic, & Jimmy Lin

Institute for Advanced Computer Studies

University of Maryland

College Park, MD, 20742

{saif,bonnie,mkegan,nmadnani,dmzajic,jimmylin}@umiacs.umd.edu

## Abstract

The University of Maryland participated in three tasks organized by the Text Analysis Conference 2008 (TAC 2008): (1) the update task of text summarization; (2) the opinion task of text summarization; and (3) recognizing textual entailment (RTE). At the heart of our summarization system is Trimmer, which generates multiple alternative compressed versions of the source sentences that act as candidate sentences for inclusion in the summary. For the first time, we investigated the use of automatically generated antonym pairs for both text summarization and recognizing textual entailment. The UMD summaries for the opinion task were especially effective in providing non-redundant information (rank 3 out of a total 19 submissions). More coherent summaries resulted when using the antonymy feature as compared to when not using it. On the RTE task, even when using only automatically generated antonyms the system performed as well as when using a manually compiled list of antonyms.

## 1 Introduction

Automatically capturing the most significant information, pertaining to a user's query, from multiple documents (also known as query-focused summarization) saves the user from having to read vast

amounts of text and yet fulfills his/her information needs. A particular kind of query-focused summarization is one in which we assume that the user has already read certain documents pertaining to a topic and the system must now summarize additional documents in a way that the summary contains mostly new information that was not present in the first set of documents. This task has come to be known as the **update task** and this paper presents University of Maryland's submission to the Text Analysis Conference 2008's update task.

With the explosion of user-generated information on the web, such as blogs and wikis, more and more people are getting useful information from these informal, less-structured, and non-homogeneous sources of text. Blogs are especially interesting because they capture the sentiment and opinion of people towards events and people. Thus, summarizing blog data is beginning to receive much attention and in this paper we present University of Maryland's opinion summarization system, as well as some of the unique challenges of summarizing blog data. For the first time we investigate the usefulness of automatically generated antonym pairs in detecting the object of opinion, contention, or dispute, and include sentences that describe such items in the summary. The UMD system was especially good at producing non-redundant summaries and we show that more coherent summaries can be obtained by using antonymy features.

The third entry by the Maryland team in TAC 2008 was a joint submission with Stanford University in the Recognizing Textual Entailment (RTE)

task.<sup>1</sup> The object of this task was to determine if an automatic system can infer the information conveyed by one piece of text (the **hypothesis**) from another piece of text (referred to simply as **text**). The Stanford RTE system uses WordNet antonyms as one of the features in this task. Our joint submission explores the usefulness of automatically generated antonym pairs. We used the Mohammad et al. (2008) method for computing word-pair antonymy. We show that using this method we can get good results even in languages that do not have a wordnet.

The next section describes related work in text summarization and for determining word-pair antonymy. Section 3 gives an overview of the UMD summarization system and how it was adapted for the two TAC 2008 summarization tasks. Section 4 gives an overview of the Stanford RTE system and how it was adapted for the TAC 2008 RTE task. Finally, in Section 5, we present the performance of the UMD submissions on the three tasks we participated in, and conclude with a discussion of future work in Section 6.

## 2 Background

### 2.1 Multidocument summarization

Extractive multi-document summarization systems typically rank candidate sentences according to a set of factors. Redundancy within the summary is minimized by iteratively re-ranking the candidates as they are selected for inclusion in the summary. For example, MEAD (Radev et al., 2004; Erkan and Radev, 2004) ranks sentences using a linear combination of features. The summary is constructed from the highest scoring sentences, then all sentences are rescored with a redundancy penalty, and a new summary is constructed based on the new ranking. This process is repeated until the summary stabilizes. Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998; Goldstein et al., 2000) balances relevance and anti-redundancy by selecting one sentence at a time for inclusion in the summary and re-scoring for redundancy after each selection. Our system takes the latter approach to summary construction, but differs in that the candidate pool is

enlarged by making multiple sentence compressions derived from the source sentences.

There are several automatic summarization systems that make use of sentence compression (Jing, 2000; Daumé and Marcu, 2005; Knight and Marcu, 2002; Banko et al., 2000; Turner and Charniak, 2005; Conroy et al., 2006; Melli et al., 2006; Hassel and Sjöbergh, 2006). All such approaches perform sentence compression through removal of non-essential sentential components (e.g., relative clauses or redundant constituents), either as pre-processing before selection or post-processing after selection. Our approach differs in that multiple trimmed versions of source sentences are generated and the selection process determines which compressed candidate, if any, of a sentence to use. The potential of multiple alternative compressions has also been explored by (Vanderwende et al., 2006).

### 2.2 Computing word-pair antonymy

Knowing that two words express some degree of contrast in meaning is useful for detecting and generating paraphrases and detecting contradictions (Marneffe et al., 2008; Voorhees, 2008). Manually created lexicons of antonym pairs have limited coverage and do not include most semantically contrasting word pairs. Thus, tens of thousands of contrasting word pairs remain unrecorded. To further complicate matters, many definitions of antonymy have been proposed by linguists (Cruse, 1986; Lehrer and Lehrer, 1982), cognitive scientists (Kagan, 1984), psycholinguists (Deese, 1965), and lexicographers (Egan, 1984), which differ from each other in small and large respects. In its strictest sense, antonymy applies to gradable adjectives, such as *hot-cold* and *tall-short*, where the two words represent the two ends of a semantic dimension. In a broader sense, it includes other adjectives, nouns, and verbs as well (*life-death*, *ascend-descend*, *shout-whisper*). In its broadest sense, it applies to any two words that represent contrasting meanings (*life-lifeless*). From an applications perspective, a broad definition of antonymy is beneficial, simply because it covers a larger range of word pairs. For example, in order to determine that sentence (1) entails sentence (2), it is useful to know that the noun *life* conveys a contrasting meaning to the adjective *lifeless*.

<sup>1</sup>Stanford University team members include Marie Catherine de Marneffe, Sebastian Pado, and Christopher Manning. We thank them for their efforts in this joint submission.

- (1) The Mars expedition began with much excitement, but all they found was a lifeless planet.
- (2) Mars has no life.

Despite the many challenges, certain automatic methods of detecting antonym pairs have been proposed. Lin et al. (2003) used patterns such as “from  $X$  to  $Y$ ” and “either  $X$  or  $Y$ ” to separate antonym word pairs from distributionally similar pairs. Turney (2008) proposed a uniform method to solve word analogy problems that requires identifying synonyms, antonyms, hypernyms, and other lexical-semantic relations between word pairs. However, the Turney method is supervised. Harabagiu et al. (2006) detected antonyms for the purpose of identifying contradictions by using WordNet chains—synsets connected by the hypernymy–hyponymy links and exactly one antonymy link. Mohammad et al. (2008) proposed an empirical measure of antonymy that combined corpus statistics with the structure of a published thesaurus. The approach was evaluated on a set of closest-opposite questions, obtaining a precision of over 80%. Notably, this method captures not only strict antonyms but also those that exhibit a degree of contrast. We use this method to detect antonyms and then use it as an additional feature in text summarization and for recognizing textual entailment.

### 3 The UMD summarization system

The UMD summarization system consists of three stages: tagging, compression, and candidate selection. (See Figure 1.) This framework has been applied to single and multi-document summarization tasks.

In the first stage, the source document sentences are part-of-speech tagged and parsed. We use the Stanford Parser (Klein and Manning 2003), a constituency parser that uses the Penn Treebank conventions. The named entities in the sentences are also identified.

In the second stage (sentence compression), multiple alternative compressed versions of the source sentences are generated, including a version with no compression, i.e., the original sentence. The module we use for this purpose is called Trimmer (Zajic, 2007). Trimmer compressions are generated by

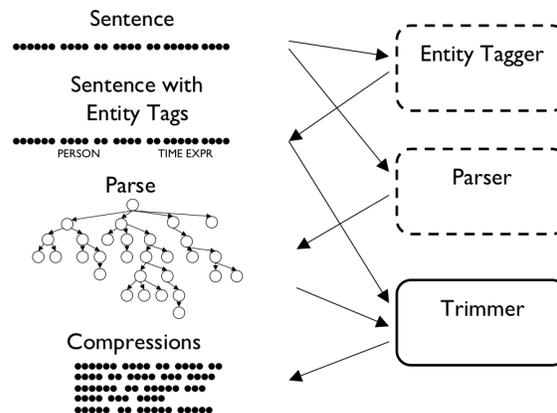


Figure 1: UMD summarization system.

applying linguistically motivated rules to mask syntactic components of the parse of a source sentence. The rules are applied iteratively, and in many combinations, to compress sentences below a configurable length threshold for generating multiple such compressions. Trimmer generates multiple compressions by treating the output of each Trimmer rule application as a distinct compression. The output of a Trimmer rule is a single parse tree and an associated surface string. Trimmer rules produce multiple outputs. These Multi-Candidate Rules (MCRs) increase the pool of candidates by having Trimmer processing continue along each MCR output. For example, a MCR for conjunctions generates three outputs: one in which the conjunction and the left child are trimmed, one in which the conjunction and the right child are trimmed, and one in which neither are trimmed. The three outputs of this particular conjunction MCR on the sentence *The program promotes education and fosters community* are: (1) *The program promotes education*, (2) *The program fosters community*, and (3) the original sentence itself. Other MCRs deal with the selection of the root node for the compression and removal of preambles. For a more detailed description of MCRs, see (Zajic, 2007). The Trimmer system used for our submission employed a number of MCRs.

Trimmer assigns compression-specific feature values to the candidate compressions that can be used in candidate selection. It uses the number of rule applications and parse tree depth of various rule applications as features.

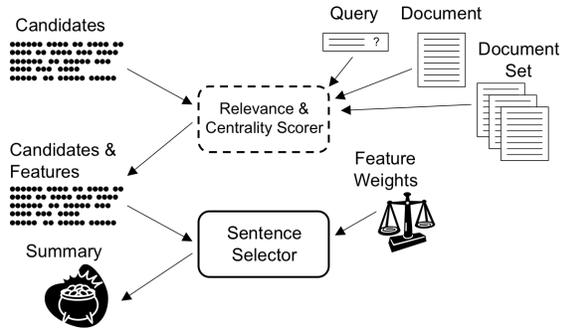


Figure 2: Selection of candidates in the UMD summarization system.

The final stage of the summarizer is the selection of candidates from the pool created by filtering and compression. (See Figure 2.) We use a linear combination of static and dynamic candidate features to select the highest scoring candidate for inclusion in the summary. Static features include position of the original sentence in the document, length, compression-specific features, and relevance scores. These are calculated prior to the candidate selection process and do not change. Dynamic features include redundancy with respect to the current summary state, and the number of candidates already in the summary from a candidate’s source document. The dynamic features have to be recalculated after every candidate selection. In addition, when a candidate is selected, all other candidates derived from the same source sentence are removed from the candidate pool. Candidates are selected for inclusion until either the summary reaches the prescribed word limit or the pool is exhausted.

### 3.1 Redundancy

One of the candidate features used in the MASC framework is the redundancy feature which measures how similar a candidate is to the current summary. If a candidate contains words that occur much more frequently in the summary than in the general language, the candidate is considered to be redundant with respect to the summary. This feature is based on the assumption that the summary is produced by a word distribution which is separate from the word distribution underlying the general language. We use an interpolated probability formulation to measure whether a candidate word  $w$  is more

likely to have been generated by the summary word distribution than by the word distribution representing the general language:

$$P(w) = \lambda P(w|S) + (1 - \lambda)P(w|L) \quad (1)$$

where  $S$  is text representing the summary and  $L$  is text representing the general language.<sup>2</sup> As a general estimate of the portion of words in a text that are specific to the text’s topic,  $\lambda$  was set to 0.3. The conditional probabilities are estimated using maximum likelihood:

$$P(w|S) = \frac{\text{Count of } w \text{ in } S}{\text{number of words in } S}$$

$$P(w|L) = \frac{\text{Count of } w \text{ in } L}{\text{number of words in } L}$$

Assuming the words in a candidate to be independently and identically distributed, the probability of the entire candidate  $C$ , i.e., the value of the redundancy feature, is given by:

$$\begin{aligned} P(C) &= \prod_{i=1}^N P(w_i) \\ &= \prod_{i=1}^N \lambda P(w_i|S) + (1 - \lambda)P(w_i|L) \end{aligned}$$

We use log probabilities for ease of computation:

$$= \sum_{i=1}^N \log(\lambda P(w_i|S) + (1 - \lambda)P(w_i|L))$$

Note that redundancy is a *dynamic* feature because the word distribution underlying the summary changes each time a new candidate is added to it.

### 3.2 Adaptation for Update Task

The interpolated redundancy feature can quantitatively indicate whether the current candidate is more like the sentences in other documents (non-redundant) or more like sentences in the current summary state (redundant). We adapted this feature for the update task to indicate whether a candidate was more like text from novel documents

<sup>2</sup>The documents in the cluster being summarized are used to estimate the general language model.

(update information) or from previously-read documents (not update information). This adaption was straightforward—for each of the three given document clusters, we added the documents that are assumed to have been previously read by the user to  $S$  from equation 1. Since  $S$  represents content already included in the summary, any candidate with content from the already-read documents is automatically considered redundant by our system.

### 3.3 Incorporating word-pair antonymy

This year we added a new feature to the summarizer that relies on word pair antonymy. Antonym pairs can help identify differing sentiment, new information, non-coreferent entities, and genuine contradictions. Below are examples of each case—the antonyms are underlined. Also note that in many of these instances, the underlined words are not strict antonyms of each other but rather contrasting word pairs.

#### **Differing sentiment:**

*The Da Vinci Code* was an exhilarating read.

*The Da Vinci Code* was rather flat.

#### **New information:**

Gravity will cause the universe to shrink. Scientists now say that the universe will continue to expand forever.

#### **Genuine contradictions:**

*The Da Vinci Code* has an original story line.

*The Da Vinci Code* is inspired heavily from *Angels and Demons*.

Antonyms are also used to compare and contrast an object of interest. For example:

Pierce Brosnan shined in the role of a happy-go-lucky private-eye impersonator, but came off rather drab in the role of James Bond.

We believe that such sentences are likely to capture the topic of discussion and are worthy of being included in the summary.

We compiled a list of antonyms, along with scores indicating the degree of antonymy for each, using

the Mohammad et al. (2008) method. The summarization system examines each of the words in a sentence to determine if it has an antonym in the same sentence or in any other sentence in the same document. If no, then the antonymy score contributed by this word is 0. If yes, then the antonymy score contributed by this word is its degree of antonymy with the word it is most antonymous to. The antonymy score of a sentence is the sum of the scores of all the words in it. This way, if a sentence has one or more words that are antonymous to other words in the document, it gets a high antonymy score and is likely to be picked to be part of the summary.

## 4 The Stanford RTE system

The details of the Stanford RTE system can be found in (MacCartney et al., 2006; Marie-Catherine de Marneffe and Manning, 2007). We briefly summarize its central elements here. The Stanford RTE system has three stages. The first stage involves linguistic preprocessing: tagging, parsing, named entity resolution, coreference resolution, and such. Dependency graphs are created for both the source text and the hypothesis. In the second stage, the system attempts to align the hypothesis words with the words in the source. Different lexical resources, including WordNet, are used to obtain similarity scores between words. A score is calculated for each candidate alignment and the best alignment is chosen. In the third stage, various features are generated to be used in a machine learning setup to determine if the hypothesis is entailed, contradicted, or neither contradicted nor entailed by the source.

A set of features generated in the third stage correspond to word-pair antonymy. The following boolean features were used which were triggered if there is an antonym pair across the source and hypothesis: (1) whether the antonymous words occur in matching polarity context; (2) whether the source text member of the antonym pair is in a negative polarity context and the hypothesis text member of the antonym pair is in a positive polarity context; and (3) whether the hypothesis text member of the antonym pair is in a negative polarity context and the source text member is in a positive context.

The polarity of the context of a word is determined by the presence of linguistic markers for

EVALUATION METRIC	Score	Rank
Pyramid	0.206/1.0	43/57
Linguistic Quality	1.938/5	49/57
Responsiveness	1.917/5	50/57
ROUGE-2	0.05624	61/71
ROUGE-SU4	0.08827	64/71

Table 1: Performance of the UMD summarizer on the update task.

negation such as *not*, downward-monotone quantifiers such as *no*, *few*, restricting prepositions such as *without*, *except*, and superlatives such as *tallest*. Antonym pairs are generated either from manually created resources such as WordNet and/or an automatically generated list using the Mohammad et al. (2008) method.

## 5 Evaluation

In TAC 2008, we participated in two summarization tasks and the recognizing textual entailment task. The following three sections describe the tasks and present the performance of the UMD submissions.

### 5.1 Summarization: Update Task

The TAC 2008 summary update task was to create short 100-word multi-document summaries under the assumption that the reader has already read some number of previous documents. There were 48 topics in the test data, with 20 documents to each topic. For each topic, the documents were sorted in chronological order and then partitioned into two sets *A* and *B*. The participants were then required to generate (a) a summary for cluster *A*, (b) an update summary for cluster *B* assuming documents in *A* have already been read. We summarized documents in set *A* using the traditional UMD summarizer. We summarized documents in set *B* using the adaptation of the redundancy feature as described in Section 3.2. (Antonymy features were NOT used in these runs.)

Table 1 lists the performance of the UMD summarizer on the update task.

Observe that the UMD summarizer performs poorly on this task. This shows that a simple adaptation of the redundancy feature for the update task is largely insufficient. More sophisticated means must be employed to create better update summaries.

### 5.2 Summarization: Opinion Task

The summarization opinion task was to generate well-organized, fluent summaries of opinions about specified targets, as found in a set of blog documents. Similar to past query-focused summarization tasks, each summary is to be focused by a number of complex questions about the target, where the question cannot be answered simply with a named entity (or even a list of named entities).

The opinion data consisted of 609 blog documents covering a total of 25 topics, with each topic covered by 9–39 individual blog pages. A single blog page contained an average of 599 lines (3,292 words / 38,095 characters) of HTML-formatted web content. This typically included many lines of header, style, and formatting information irrelevant to the content of the blog entry itself. A substantial amount of cleaning was therefore necessary to prepare the blog data for summarization.

To clean a given blog page, we first extracted the content from within the HTML `<BODY>` tags of the document. This eliminated the extraneous formatting information typically included in the `<HEAD>` section of the document, as well as other metadata. We then decoded HTML-encoded characters such as “&nbsp;” (for space), “&quot;” for (double quote), and “&amp;” for (ampersand). We converted common HTML separator tags such as “<BR>”, “<HR>”, “<TD>”, and “<P>” into newlines, and stripped all remaining HTML tags. This left us with substantially English text content. However, even this included a great amount of text irrelevant to the blog entry itself. We devised hand-crafted rules to filter out common non-content phrases such as “Posted by...”, “Published by...”, “Related Stories...”, “Copyright...”, and others. However, the extraneous content differed greatly between blog sites, making it difficult to handcraft rules that would work in the general case. We therefore made the decision to filter out any line of text that did not contain at least 6 words. Although it is likely that we filtered out some amount of valid content in doing so, this strategy helped greatly in eliminating irrelevant text. The average length of a cleaned blog file was 58 lines (1,535 words / 8,874 characters), down from the original average of 599 lines (3,292 words / 38,095 characters).

EVALUATION METRIC	NO antonymy		WITH antonymy	
	Score	Rank	Score	Rank
Pyramid	0.136/1	11/19	0.130/1	13/19
Grammaticality	4.409/10	16/19	4.318/10	17/19
Non-redundancy	6.682/10	3/19	6.455/10	5/19
Coherence	2.364/5	13/19	2.409/5	11/19
Fluency	3.545/10	9/19	3.318/10	15/19

Table 2: Performance of the UMD summarizer on the opinion task.

Table 2 shows the performance of the UMD summarizer on the opinion task with and without using the word-pair antonymy feature. The ranks provided are for only the automatic systems and those that do not make use of the answer snippet. This is because our system is automatic and does not use answer snippets provided for optional use by the organizers. Observe that the UMD summarizer stands roughly middle of the pack among the other systems that took part in this task. However, it is especially strong on non-redundancy (rank 3). Also note that adding the antonymy feature improved the coherence of resulting summaries (rank jumped from 13 to 11). This is because the feature encourages inclusion of more sentences that describe the topic of discussion. However, performance on other metrics dropped with the inclusion of the antonymy feature.

### 5.3 Recognizing Textual Entailment Task

The objective of the Recognizing Textual Entailment task was to determine if one piece of text (the source) entail another piece of text (the hypothesis), whether it contradicts it, or whether neither inference can be drawn. The TAC 2008 RTE data consisted of 1,000 source and hypothesis pairs.

The Stanford–UMD joint submission consisted of three submissions differing in which antonym pairs were used: submission 1 used antonym pairs from WordNet; submission 2 used automatically generated antonym pairs (Mohammad, Dorr, and Hirst 2008 method); and submission 3 used automatically generated antonyms as well as manually compiled lists from WordNet. Table 3 shows the performance of the three RTE submissions. Our system stood 7th among the 33 participating systems. Observe that using the automatic method of generating antonym pairs, the system performs as well, if not slightly better, than when using antonyms from a manually

SOURCE OF ANTONYMS	2-WAY ACCURACY	3-WAY ACCURACY	AVERAGE PRECISION
WordNet	61.7	55.4	44.08
Automatic	61.7	55.6	44.26
Both	61.7	55.6	44.27

Table 3: Performance of the Stanford–UMD RTE submissions.

created resource such as WordNet. This is an especially nice result for resource-poor languages, where the automatic method can be used in place of high-quality linguistic resources. Using both automatically and manually generated antonyms did not increase performance by much.

## 6 Future Work

The use of word-pair antonymy in our entries for both opinion summarization and for recognizing textual entailment, although intuitive, is rather simple. We intend to further explore and more fully utilize this phenomenon through many other features. For example, if a word in one sentence is antonymous to a word in a sentence immediately following it, then it is likely that we would want to include both sentences in the summary or neither. The idea is to include or exclude pairs of sentences connected by a contrast discourse relation. Further, our present TAC entries only look for antonyms within a document. Antonyms can be used to identify contentious or contradictory sentences across documents that may be good candidates for inclusion in the summary. In the textual entailment task, antonyms were used only in the third stage of the Stanford RTE system. Antonyms are likely to be useful in determining better alignments between the source and hypothesis text (stage two), and this is something we hope to explore more in the future.

## Acknowledgments

We thank Marie Catherine de Marneffe, Sebastian Pado, and Christopher Manning for working with us on the joint submission to the recognizing textual entailment task. This work was supported, in part, by the National Science Foundation under Grant No. IIS-0705832 and, in part, by the Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations ex-

pressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

## References

- Michele Banko, Vibhu Mittal, and Michael Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of ACL*, pages 318–325, Hong Kong.
- Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pages 335–336, Melbourne, Australia.
- John M. Conroy, Judith D. Schlesinger, Dianne P. O’Leary, and J. Goldstein. 2006. Back to basics: CLASSY 2006. In *Proceedings of the NAACL-2006 Document Understanding Conference Workshop*, New York, New York.
- David A. Cruse. 1986. *Lexical semantics*. Cambridge University Press.
- Hal Daumé and Daniel Marcu. 2005. Bayesian multi-document summarization at MSE. In *Proceedings of ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.
- James Deese. 1965. *The structure of associations in language and thought*. The Johns Hopkins Press.
- Rose F. Egan. 1984. Survey of the history of English synonymy. *Webster’s New Dictionary of Synonyms*, pages 5a–25a.
- Güneş Erkan and Dragomir R. Radev. 2004. The university of michigan at duc2004. In *Proceedings of Document Understanding Conference Workshop*, pages 120–127.
- Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, pages 40–48.
- Sanda M. Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Lacatusu: Negation, contrast and contradiction in text processing. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-06)*, Boston, MA.
- Martin Hassel and Jonas Sjöbergh. 2006. Towards holistic summarization – selecting summaries, not sentences. In *Proceedings of LREC*, Genoa, Italy.
- Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of ANLP*.
- Jerome Kagan. 1984. *The Nature of the Child*. Basic Books.
- Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.
- Adrienne Lehrer and K. Lehrer. 1982. Antonymy. *Linguistics and Philosophy*, 5:483–501.
- Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1492–1493, Acapulco, Mexico.
- B. MacCartney, T. Grenager, M.C. de Marneffe, D. Cer, and C. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the North American Association for Computational Linguistics*.
- Bill MacCartney Daniel Cer Daniel Ramage Chlo Kiddon Marie-Catherine de Marneffe, Trond Grenager and Christopher D. Manning. 2007. Aligning semantic graphs for textual inference and machine reading. In *In AAAI Spring Symposium at Stanford*. 2007.
- Gabor Melli, Zhongmin Shi, Yang Wang, Yudong Liu, Annop Sarkar, and Fred Popowich. 2006. Description of SQUASH, the SFU question answering summary handler for the DUC-2006 summarization task. In *Proceedings of DUC*.
- Saif Mohammad, Bonnie Dorr, and Graeme Hirst. 2008. Computing word-pair antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-2008)*, Waikiki, Hawaii.
- Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. In *Information Processing and Management*, volume 40, pages 919–938.
- Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of ACL 2005*, pages 290–297, Ann Arbor, Michigan.
- Peter Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, pages 905–912, Manchester, UK.
- Lucy Vanderwende, Hisami Suzuki, and Chris Brockett. 2006. Microsoft Research at DUC2006: Task-focused summarization with sentence simplification and lexical expansion. In *Proceedings of DUC*, pages 70–77.
- David M. Zajic. 2007. *Multiple Alternative Sentence Compressions (MASC) as a Tool for Automatic Summarization Tasks*. Ph.D. thesis, University of Maryland, College Park.