Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression

Florian Boudin and **Emmanuel Morin**
LINA - UMR CNRS 6241, Université de Nantes, France
{florian.boudin,emmanuel.morin}@univ-nantes.fr

Abstract

Multi-Sentence Compression (MSC) is the task of generating a short single sentence summary from a cluster of related sentences. This paper presents an N-best reranking method based on keyphrase extraction. Compression candidates generated by a word graph-based MSC approach are reranked according to the number and relevance of keyphrases they contain. Both manual and automatic evaluations were performed using a dataset made of clusters of newswire sentences. Results show that the proposed method significantly improves the informativity of the generated compressions.

1 Introduction

Multi-Sentence Compression (MSC) can be broadly described as the task of generating a short single sentence summary from a cluster of related sentences. It has recently attracted much attention, mostly because of its relevance to single or multi-document extractive summarization. A standard way to generate summaries consists in ranking sentences by importance, cluster them by similarity and select a sentence from the top ranked clusters (Wang et al., 2008). One difficulty is then to generate concise, non-redundant summaries. Selected sentences almost always contain additional information specific to the documents from which they came, leading to readability issues in the summary.

Sentence Compression (SC), i.e. the task of summarizing a sentence while retaining most of the informational content and remaining grammatical (Jing, 2000), is a straightforward solution to this problem. Another solution would be to create, for each cluster of related sentences, a concise and fluent fusion of information, reflecting facts common to all sentences. Originally defined as sentence fusion (Barzilay and McKeown, 2005), MSC is a text-to-text generation process in which a novel sentence is produced as a result of summarizing common information across a set of similar sentences.

Most of the previous MSC approaches rely on syntactic parsers for producing grammatical compressions, e.g. (Filippova and Strube, 2008; Elsner and Santhanam, 2011). Recently, (Filippova, 2010) proposed a word graph-based approach which only requires a Part-Of-Speech (POS) tagger and a list of stopwords. The key assumption behind her approach is that redundancy within the set of related sentences provides a reliable way of generating informative and grammatical sentences. Although this approach seemingly works well, 48% to 60% of the generated sentences are missing important information about the set of related sentences. In this study, we aim at producing more informative sentences by maximizing the range of topics they cover.

Keyphrases are words that capture the main topics of a document. Extracting keyphrases can benefit various Natural Language Processing tasks such as summarization, information retrieval and question-answering (Kim et al., 2010). In summarization, keyphrases provide semantic metadata that represent the content of a document. Sentences containing the most relevant keyphrases are used to generate the summary (D'Avanzo and Magnini, 2005). In the same way, we hypothesize that keyphrases can be used to better generate sentences that convey the gist

of the set of related sentences.

In this paper, we present a reranking method of N-best multi-sentence compressions based on keyphrase extraction and describe a series of experiments conducted on a manually constructed evaluation corpus. More precisely, the main contributions of our work are as follows:

- We extend Filippova (2010)'s word graph-based MSC approach to produce well-punctuated and more informative compressions.
- We investigate the use of automatic Machine Translation (MT) and summarization evaluation metrics to evaluate MSC performance.
- We introduce a French evaluation dataset made of 40 sets of related sentences along with reference compressions composed by humans.

The rest of this paper is organized as follows. We first briefly review the previous work, followed by a description of the method we propose. Next, we give the details of the evaluation dataset we have constructed and present our experiments and results. Lastly, we conclude with a discussion and directions for further work.

2 Related work

2.1 Multi-sentence compression

MSC have received much attention recently and many different approaches have been proposed. The pioneering work of (Barzilay and McKeown, 2005) introduced the framework used by many subsequent works: input sentences are represented by dependency trees, some words are aligned to merge the trees into a lattice, and the lattice is linearized using tree traversal to produce fusion sentences. (Filippova and Strube, 2008) cast MSC as an integer linear program, and show promising results for German. Later, (Elsner and Santhanam, 2011) proposed a supervised approach trained on examples of manually fused sentences.

Previously described approaches require the use of a syntactic parser to control the grammaticality of the output. As an alternative, several word graph-based approaches that only require a POS tagger were proposed. The key assumption is that redundancy provides a reliable way of generating grammatical sentences. First, a directed word graph is constructed from the set of input sentences in which nodes represent unique words, defined as word and POS tuples, and edges express the original structure of sentences (i.e. word ordering). Sentence compressions are obtained by finding commonly used paths in the graph. Word graph-based MSC approaches were used in different tasks, such as guided microblog summarization (Sharifi et al., 2010), opinion summarization (Ganesan et al., 2010) and newswire summarization (Filippova, 2010).

2.2 Keyphrase extraction

Keyphrases are words that are representative of the main content of documents. Extracting keyphrases can benefit various Natural Language Processing tasks such as summarization, information retrieval and question-answering (Kim et al., 2010). Previous works fall into two categories: supervised and unsupervised methods. The idea behind supervised methods is to recast keyphrase extraction as a binary classification task. A model is trained using annotated data to determine whether a given phrase is a keyphrase or not (Frank et al., 1999; Turney, 2000).

Unsupervised approaches proposed so far have involved a number of techniques, including language modeling (Tomokiyo and Hurst, 2003), graph-based ranking (Mihalcea and Tarau, 2004; Wan and Xiao, 2008) and clustering (Liu et al., 2009). While supervised approaches have generally proven more successful, the need for training data and the bias towards the domain on which they are trained remain two critical issues.

3 Method

In this section, we first describe Filippova (2010)'s word graph-based MSC approach. Then, we present the keyphrase extraction approach we use and our method for reranking generated compressions.

3.1 Description of Filippova's approach

Let $G = (V, E)$ be a directed graph with the set of vertices (nodes) V and a set of directed edges E, where E is a subset of $V \times V$. Given a set of related sentences $S = \{s_1, s_2, ..., s_n\}$, a word graph is constructed by iteratively adding sentences to it.

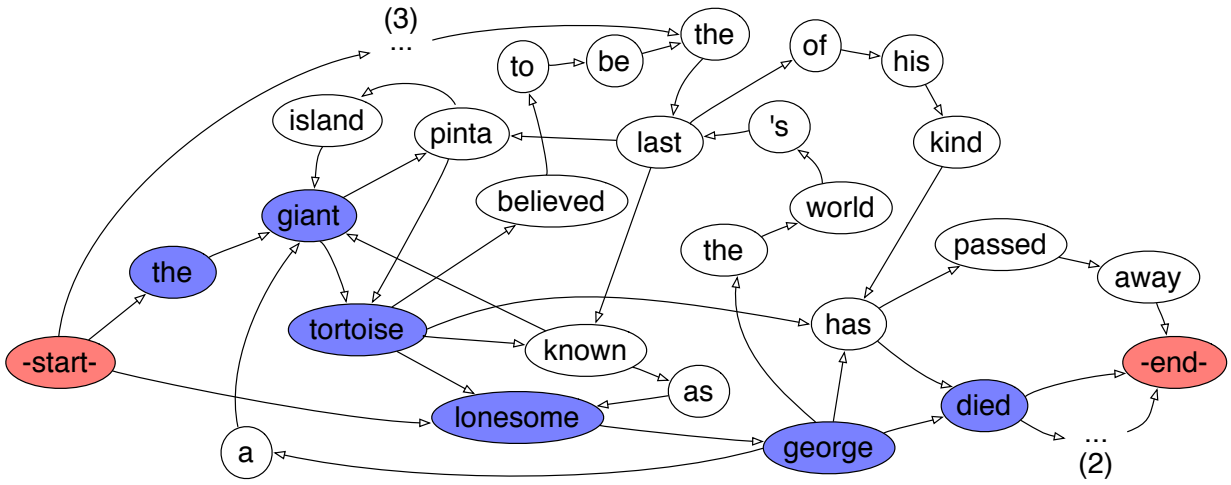

Figure 1: Word graph constructed from the set of related sentences, a possible compression path is also given.

Figure 1 is an illustration of the word graph constructed from the following sentences. For clarity, edge weights are omitted and italicized fragments from the sentences are replaced with dots.

1. Lonesome George, the world's last Pinta Island giant tortoise, has passed away.
2. The giant tortoise known as Lonesome George died *Sunday at the Galapagos National Park in Ecuador*.
3. *He was only about a hundred years old, but* the last known giant Pinta tortoise, Lonesome George, has passed away.
4. Lonesome George, a giant tortoise believed to be the last of his kind, has died.

At the first step, the graph simply represents one sentence plus the start and end symbols (–start– and –end– in Figure 1). A node is added to G for each word in the sentence, and words adjacent in the sentence are connected with directed edges. A word from the following sentences is mapped onto an existing node in the graph if they have the same lowercased word form and POS and that no word from this sentence has already been mapped onto this node. A new node is created if there is no suitable candidate in the graph.

Words are added to the graph in the following order:

i. non-stopwords for which no candidate exists in the graph or for which an unambiguous mapping is possible;

ii. non-stopwords for which there are either several possible candidates in the graph or which occur more than once in the sentence;

iii. stopwords.

For the last two groups of words where mapping is ambiguous (i.e. there are two or more nodes in the graph that refer to the same word/POS tuple), the immediate context (the preceding and following words in the sentence and the neighboring nodes in the graph) or the frequency (i.e. the node which has words mapped onto it) are used to select the candidate node. We use the stopword list included in nltk[1] extended with temporal nouns (e.g. monday, yesterday).

In Filippova's approach, punctuation marks are excluded. To generate well-punctuated compressions, we simply added a fourth step for adding punctuation marks in the graph. When mapping is ambiguous, we select the candidate which has the same immediate context.

Once the words from a sentence are added to the graph, words adjacent in the sentence are connected with directed edges. Edge weights are calculated using the weighting function defined in Equation 1.

[1] http://nltk.org/

$$w(i,j) = \frac{cohesion(i,j)}{freq(i) \times freq(j)} \quad (1)$$

$$cohesion(i,j) = \frac{freq(i) + freq(j)}{\sum_{s \in S} d(s,i,j)^{-1}} \quad (2)$$

where $freq(i)$ is the number of words mapped to the node i. The function $d(s,i,j)$ refers to the distance between the offset positions of words i and j in sentence s.

The purpose of this function is two fold: i. to generate a grammatical compression, links between words which appear often in this order are favored (see Equation 2); ii. to generate an informative compression, the weight of edges connecting salient nodes is decreased.

A K-shortest paths algorithm is then used to find the 50 shortest paths from start to end nodes in the graph. Paths shorter than eight words or that do not contain a verb are filtered. The remaining paths are reranked by normalizing the total path weight over its length. The path which has the lightest average edge weight is then considered as the best compression.

3.2 Reranking paths using keyphrases

The main difficulty of MSC is to generate sentences that are both informative and grammatically correct. Here, redundancy within the set of input sentences is used to identify important words and salient links between words. Although this approach seemingly works well, important information is missing in 48% to 60% of the generated sentences (Filippova, 2010). One of the reasons for this is that node salience is estimated only with the frequency measure. To tackle this issue, we propose to rerank the N-best list of compressions using keyphrases extracted from the set of related sentences. Intuitively, an informative sentence should contain the most relevant keyphrases. We propose to rerank generated compressions according to the number and relevance of keyphrases they contain.

An unsupervised method based on (Wan and Xiao, 2008) is used to extract keyphrases from each set of related sentences. This method is based on the assumption that a word recommends other co-occurring words, and the strength of the recommendation is recursively computed based on the importance of the words making the recommendation. Keyphrase extraction can be divided into two steps. First, a weighted graph is constructed from the set of related sentences, in which nodes represent words defined as word and POS tuples. Two nodes (words) are connected if their corresponding lexical units co-occur within a sentence. Edge weights are the number of times two words co-occur. TextRank (Mihalcea and Tarau, 2004), a graph-based ranking algorithm that takes into account edge weights, is applied for computing a salience score for each node. The score for node V_i is initialized with a default value and is computed in an iterative manner until convergence using this equation:

$$S(V_i) = (1-d) + d \times \sum_{V_j \in adj(V_i)} \frac{w_{ji}}{\sum_{V_k \in adj(V_i)} w_{jk}} S(V_i)$$

where $adj(V_i)$ denotes the neighbors of V_i and d is the damping factor set to 0.85.

The second step consists in generating and scoring keyphrase candidates. Sequences of adjacent words satisfying a specific syntactic pattern are collapsed into multi-word phrases. We use $(ADJ)\star(NPP|NC)+(ADJ)\star$ for French, in which ADJ are adjectives, NPP are proper nouns and NC are common nouns.

The score of a candidate keyphrase k is computed by summing the salience scores of the words it contains normalized by its length + 1 to favor longer n-grams (see equation 3).

$$score(k) = \frac{\sum_{w \in k} TextRank(w)}{length(k) + 1} \quad (3)$$

The small vocabulary size as well as the high redundancy within the set of related sentences are two factors that make keyphrase extraction easier to achieve. On the other hand, a large number of the generated keyphrases are redundant. Some keyphrases may be contained within larger ones, e.g. *giant tortoise* and *Pinta Island giant tortoise*. To solve this problem, generated keyphrases are clustered using word overlap. For each cluster, we then select the keyphrase with the highest score. This filtering process enables the generation of a smaller subset of keyphrases while having a better coverage of the cluster content.

Reranking techniques can suffer from the limited scope of the N-best list, which may rule out many potentially good candidates. For this reason, we use a larger number of paths than the one in (Filippova, 2010). Accordingly, the K-shortest paths algorithm is used to find the 200 shortest paths. We rerank the paths by normalizing the total path weight over its length multiplied by the sum of keyphrase scores it contains. The score of a sentence compression c is given by:

$$score(c) = \frac{\sum_{i,j \in path(c)} w_{(i,j)}}{length(c) \times \sum_{k \in c} score(k)} \quad (4)$$

4 Experimental settings

4.1 Construction of the evaluation dataset

To our knowledge, there is no dataset available to evaluate MSC in an automatic way. The performance of the previously described approaches was assessed by human judges. In this work, we introduce a new evaluation dataset made of 40 sets of related sentences along with reference compressions composed by human assessors. The purpose of this dataset is to investigate the use of existing automatic evaluation metrics for the MSC task.

Similar to (Filippova, 2010), we collected news articles presented in clusters on the French edition of Google News[2] over a period of three months. Clusters composed of at least 20 news articles and containing one single prevailing event were manually selected. To obtain the sets of related sentences, we extracted the first sentences from each article in the cluster, removing duplicates. Leading sentences in news articles are known to provide a good summary of the article content and are used as a baseline in summarization (Dang, 2005).

The resulting dataset contains 618 sentences (33 tokens on average) spread over 40 clusters. The number of sentences within each cluster is on average 15, with a minimum of 7 and a maximum of 36. The word redundancy rate within the dataset, computed as the number of unique words over the number of words for each cluster, is 38.8%.

Three reference compressions were manually composed for each set of sentences. Human annotators, all native French speakers, were asked to carefully read the set of sentences, extract the most salient facts and generate a sentence (compression) that summarize the set of sentences. Annotators were also told to introduce as little new vocabulary as possible in their compressions. The purpose of this guideline is to reduce the number of possible mismatches, as existing evaluation metrics are based on n-gram comparison. Reference compressions have a compression rate of 60%.

4.2 Automatic evaluation

The use of automatic methods for evaluating machine-generated text has gradually become the mainstream in Computational Linguistics. Well known examples are the ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) evaluation metrics used in the summarization and MT communities. These metrics assess the quality of a system output by computing its similarity to one or more human-generated references.

Prior work in sentence compression use the F1 measure over grammatical relations to evaluate candidate compressions (Riezler et al., 2003). It was shown to correlate significantly with human judgments (Clarke and Lapata, 2006) and behave similarly to BLEU (Unno et al., 2006). However, this metric is not entirely reliable as it depends on parser accuracy and the type of dependency relations used (Napoles et al., 2011). In this work, the following evaluation measures are considered relevant: BLEU[3], ROUGE-1 (unigrams), ROUGE-2 (bigrams) and ROUGE-SU4 (bigrams with skip distance up to 4 words)[4]. ROUGE measures are computed using stopword removal and French stemming [5].

4.3 Manual evaluation

The quality of the generated compressions was assessed in an experiment with human raters. Two aspects were considered: grammaticality and informativity. Following previous work (Barzilay and McKeown, 2005), we asked raters to assess grammaticality on a 3-points scale: *perfect* (2 pts), if the compression is a complete grammatical sentence; *almost*

[2] http://news.google.fr

[3] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

[4] We use the version 1.5.5 of the ROUGE package available from http://www.berouge.com

[5] http://snowball.tartarus.org/

(1 pt), if it requires minor editing, e.g. one mistake in articles, agreement or punctuation; *ungrammatical* (0 pts), if it is none of the above. Raters were explicitly asked to ignore lack of capitalization while evaluating grammaticality.

Informativity is evaluated according to the 3-points scale defined in (Filippova, 2010): *perfect* (2 pts), if the compression conveys the gist of the main event and is more or less like the summary the person would produce himself; *related* (1 pt), if it is related to the the main theme but misses something important; *unrelated* (0 pts), if the compression is not related to the main theme.

Three raters, all native French speakers, were hired to assess the generated compressions.

5 Results

To evaluate the effectiveness of our method, we compare the compressions generated with Filippova's approach (denoted as baseline) against the ones obtained by reranking paths using keyphrases (denoted as KeyRank). We evaluated the agreement between the three raters using Fleiss's kappa (Artstein and Poesio, 2008). The κ value is 0.56 which denotes a moderate agreement.

Table 1 presents the average grammaticality and informativity scores. Results achieved by the baseline are consistent with the ones presented in (Filippova, 2010). We observe a significant improvement in informativity for KeyRank. Grammaticality scores are, however, slightly decreased. One reason for that is the reranking we added to the shortest path method that outputs longer compressions. The average length for our method is nevertheless drastically shorter than the average length of the input sentences (19 vs. 33 tokens). This corresponds to a compression rate (58%) that is close to the one observed on reference compressions (60%).

Table 2 shows the distributions over the three scores for both grammaticality and informativity. We observe that 97.5% of the compressions generated with KeyRank are related to the main theme of the cluster, and 62.5% convey the very gist of it without missing any important information. This represents an absolute increase of 19.2% over the baseline. Although our reranking method has lower grammaticality scores, 65% of the generated sentences are perfectly grammatical.

Method	**Gram.**	**Info.**	**Length**		**CompR**
			Avg.	Std.Dev.	
Baseline	1.63	1.33	16.3	4.8	50%
KeyRank	1.53	1.60†	19	6.1	58%

Table 1: Average ratings over all clusters and raters along with average compression length (in tokens), standard deviation and corresponding compression rate († indicates significance at the 0.01 level using Student's t-test).

Method	**Gram.**			**Info.**		
	0	1	2	0	1	2
Baseline	9.2%	18.3%	72.5%	10.0%	46.7%	43.3%
KeyRank	11.7%	23.3%	65.0%	2.5%	35.0%	62.5%

Table 2: Distribution over possible manual ratings for grammaticality and informativity. Ratings are expressed on a scale of 0 to 2.

Table 3 shows the performance of the baseline and our reranking method in terms of ROUGE and BLEU scores. KeyRank significantly outperforms the baseline according to the different ROUGE metrics. This indicates an improvement in informativity for the compressions generated using our method. We observe a large but not significant increase in BLEU scores. The slightly decreased grammaticality scores could be a reason for this. BLEU is essentially a precision metric, and it measures how well a compression candidate overlaps with multiple references. Longer n-grams used by BLEU[6] tend to score for grammaticality rather than content.

Metric	**Baseline**	**KeyRank**
ROUGE-1	0.57441	0.65677‡
ROUGE-2	0.39212	0.44140†
ROUGE-SU4	0.37004	0.43443‡
BLEU	0.61560	0.65770

Table 3: Automatic evaluation scores († and ‡ indicate significance at the 0.01 and 0.001 levels respectively using Student's t-test)

To assess the effectiveness of automatic evalua-

[6] BLEU measures are computed using 4-grams.

tion metrics, we compute the Pearson's correlation coefficient between ROUGE and BLEU scores and averaged manual ratings. According to Table 4, results show medium to strong correlation between ROUGE scores and informativity ratings. On the other hand, BLEU scores better correlate with grammaticality ratings. Overall, automatic evaluation metrics are not highly correlated with manual ratings. One reason for that may be that the manual score assignments are arbitrary (i.e. 0, 1, 2), and that a score of one is in fact closer to two than to zero. Results suggest that automatic metrics do give an indication of the compression quality, but can not replace manual evaluation.

Metric	Gram.	Info.
ROUGE-1	0.402	0.591
ROUGE-2	0.432	0.494
ROUGE-SU4	0.386	0.542
BLEU	0.444	0.401

Table 4: Pearson correlation coefficients for automatic metrics vs. average human ratings.

6 Conclusion

This paper presented a multi-sentence compression approach that uses keyphrases to generate more informative compressions. We extended Filippova (2010)'s word graph-based MSC approach by adding a re-reranking step that favors compressions that contain the most relevant keyphrases of the input sentence set. An implementation of the proposed multi-sentence compression approach is available for download[7]. We constructed an evaluation dataset made of 40 sets of related sentences along with reference compressions composed by humans. This dataset is freely available for download[8]. We performed both manual and automatic evaluations and showed that our method significantly improves the informativity of the generated compressions. We also investigated the correlation between manual and automatic evaluation metrics and found that ROUGE and BLEU have a medium correlation with manual ratings.

[7] https://github.com/boudinfl/takahe

[8] https://github.com/boudinfl/lina-msc

In future work, we intend to examine how grammaticality of the generated compressions can be enhanced. Similar to the work of Hasan et al. (2006) in the Machine Translation field, we plan to experiment with high order POS language models reranking.

Acknowledgments

The authors would like to thank Sebastián Peña Saldarriaga and Ophélie Lacroix for helpful comments on this work. We thank the anonymous reviewers for their useful comments. This work was supported by the French Agence Nationale de la Recherche under grant ANR-12-CORD-0027 and by the French Region Pays de Loire in the context of the DEPART project (http://www.projetdepart.org/).

References

R. Artstein and M. Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 377–384, Sydney, Australia, July. Association for Computational Linguistics.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the Document Understanding Conference*.

Ernesto D'Avanzo and Bernardo Magnini. 2005. A keyphrase-based approach to summarization: the lake system at duc-2005. In *Proceedings of the Document Understanding Conference*.

Micha Elsner and Deepak Santhanam. 2011. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 54–63, Portland, Oregon, June. Association for Computational Linguistics.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 177–185, Honolulu, Hawaii, October. Association for Computational Linguistics.

Katja Filippova. 2010. Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. In *Proceed-*

ings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 322–330, Beijing, China, August. Coling 2010 Organizing Committee.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China, August. Coling 2010 Organizing Committee.

S. Hasan, O. Bender, and H. Ney. 2006. Reranking translation hypotheses using structural properties. In *Proceedings of the EACL Workshop on Learning Structured Information in Natural Language Applications*, pages 41–48.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 310–315, Seattle, Washington, USA, April. Association for Computational Linguistics.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden, July. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266, Singapore, August. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97, Portland, Oregon, June. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 118–125. Association for Computational Linguistics.

Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. 2010. Summarizing Microblogs Automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 685–688, Los Angeles, California, June. Association for Computational Linguistics.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40, Sapporo, Japan, July. Association for Computational Linguistics.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

Yuya Unno, Takashi Ninomiya, Yusuke Miyao, and Jun'ichi Tsujii. 2006. Trimming cfg parse trees for sentence compression using machine learning approaches. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 850–857, Sydney, Australia, July. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976, Manchester, UK, August. Coling 2008 Organizing Committee.

Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. 2008. Multi-document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 307–314, New York, NY, USA. ACM.