Extractive Summarization by Aggregating Multiple Similarities

Olof Mogren, Mikael Kågebäck, Devdatt Dubhashi
Department of Computer Science and Engineering
Chalmers University of Technology,
412 96 Göteborg, Sweden
mogren@chalmers.se

Abstract

News reports, social media streams, blogs, digitized archives and books are part of a plethora of reading sources that people face every day. This raises the question of how to best generate automatic summaries. Many existing methods for extracting summaries rely on comparing the similarity of two sentences in some way. We present new ways of measuring this similarity, based on sentiment analysis and continuous vector space representations, and show that combining these together with similarity measures from existing methods, helps to create better summaries. The finding is demonstrated with MULTSUM, a novel summarization method that uses ideas from kernel methods to combine sentence similarity measures. Submodular optimization is then used to produce summaries that take several different similarity measures into account. Our method improves over the state-of-the-art on standard benchmark datasets; it is also fast and scale to large document collections, and the results are statistically significant.

1 Introduction

Extractive summarization, the process of selecting a subset of sentences from a set of documents, is an important component of modern information retrieval systems (Baeza-Yates et al., 1999). A good summarization system needs to balance two complementary aspects: finding a summary that captures all the important topics of the documents (*coverage*), yet does not contain too many similar sentences (*non-redundancy*). It follows that it is essential to have a good way of measuring the similarity of sentences, in no way a trivial task. Consequently, several measures for sentence similarity have been explored for extractive summarization.

In this work, two sets of novel similarity measures capturing deeper semantic features are presented, and evaluated in combination with existing methods of measuring sentence similarity. The new methods are based on sentiment analysis, and continuous vector space representations of phrases, respectively.

We show that summary quality is improved by combining multiple similarities at the same time using kernel techniques. This is demonstrated using MULTSUM, an ensemble-approach to generic extractive multi-document summarization based on the existing, state-of-the-art method of Lin and Bilmes (2011). Our method obtains state-of-the-art results that are statistically significant on the de-facto standard benchmark dataset DUC 04. The experimental evaluation also confirm that the method generalizes well to other datasets.

2 MULTSUM

MULTSUM, our approach for extractive summarization, finds representative summaries taking multiple sentence similarity measures into account. As Lin and Bilmes (2011), we formulate the problem as the optimization of monotone non-decreasing submodular set functions. This results in a fast, greedy optimization step that provides a $(1 - \frac{1}{e})$ factor approximation. In the original version, the optimization objective is a function scoring a candidate summary by coverage and diversity, expressed using cosine similarity between sentences represented as bag-of-terms vectors. We extend this method by using several sentence similarity measures M^l (as described in Section 3) at the same time, combined by multiplying them together element-wise:

$$M_{\mathbf{s}_i,\mathbf{s}_j} = \prod M^l_{\mathbf{s}_i,\mathbf{s}_j}.$$

In the literature of kernel methods, this is the standard way of combining kernels as a conjunction (Duvenaud, 2014; Schölkopf et al., 2004, Ch 1).

3 Sentence Similarity Measures

Many existing systems rely on measuring the similarity of sentences to balance the coverage with the amount of redundancy of the summary. This is also true for MULTSUM which is based on the existing submodular optimization method. Similarity measures that capture general aspects lets the summarization system pick sentences that are representative and diverse in general. Similarity measures capturing more specific aspects allow the summarization system to take these aspects into account.

We list some existing measures in Table 3 (that mainly relies on counting word overlaps) and in Sections 3.1 and 3.2, we present sentence similarity measures that capture more specific aspects of the text. MULTSUM is designed to work with all measures mentioned below; this will be evaluated in Section 4. Interested readers are referred to a survey of existing similarity measures from the litterature in (Bengtsson and Skeppstedt, 2012). All these similarity measures require sentence splitting, tokenization, part-of-speech tagging and stemming of words. The Filtered Word, and TextRank comparers are set similarity measures where each sentence is represented by the set of all their terms. The KeyWord comparer and LinTFIDF represent each sentence as a word vector and uses the vectors for measuring similarity.

DepGraph first computes the dependency parse trees of the two sentences using Maltparser (Nivre, 2003). The length of their longest common path is then used to derive the similarity score.

The similarity measure used in TextRank (Mihalcea and Tarau, 2004) will be referred to as TRComparer. The measure used in submodular optimization (Lin and Bilmes, 2011) will be referred to as LinTFIDF. All measures used in this work are normalized, $M_{\mathbf{s}_i,\mathbf{s}_j} \in [0,1]$.

3.1 Sentiment Similarity

Sentiment analysis has previously been used for document summarization, with the aim of capturing an average sentiment of the input corpus (Lerman et al., 2009), or to score emotionally charged sentences (Nishikawa et al., 2010). Other research has shown that negative emotion words appear at a relative higher rate in summaries written by humans (Hong and Nenkova, 2014). We propose a different way of making summaries sentiment aware by comparing the level of sentiment in sentences. This allows for summaries that are both representative and diverse in sentiment.

Name	Formula
Filtered	$M_{\mathbf{s}_i,\mathbf{s}_j} = \lvert(\mathbf{s}_i \cap \mathbf{s}_j)\rvert / \sqrt{\lvert(\mathbf{s}_i)\rvert + \lvert(\mathbf{s}_j)\rvert}$
TRCmp.	$M_{\mathbf{s}_i,\mathbf{s}_j} = \lvert\mathbf{s}_i \cap \mathbf{s}_j\rvert / (log\lvert\mathbf{s}_i\rvert + log\lvert\mathbf{s}_j\rvert)$
LinTFIDF	$M_{\mathbf{s}_i,\mathbf{s}_j} = \frac{\sum_{w \in \mathbf{s}_i} tf_{w,i} \cdot tf_{w,j} \cdot idf_w^2}{\sqrt{\sum_{w \in \mathbf{s}_i} tf_{w,\mathbf{s}_i} idf_w^2} \sqrt{\sum_{w \in \mathbf{s}_j} tf_{w,\mathbf{s}_j} idf_w^2}}$
KeyWord	$M_{\mathbf{s}_i,\mathbf{s}_j} = \frac{\sum_{w \in \{\{\mathbf{s}_i \cap \mathbf{s}_j\} \cap K\}} tf_w \cdot idf_w}{\lvert\mathbf{s}_i\rvert + \lvert\mathbf{s}_j\rvert}$
DepGraph	*See text description.*

Table 1: Similarity measures from previous works.

Two lists, of positive and of negative sentiment words respectively, were manually created[1] and used. Firstly, each sentence $\mathbf{s}_i$ is given two sentiment scores, $positive(\mathbf{s}_i)$ and $negative(\mathbf{s}_i)$, defined as the fraction of words in $\mathbf{s}_i$ that is found in the positive and the negative list, respectively. The similarity score for positive sentiment are computed as follows:

$$M_{\mathbf{s}_i,\mathbf{s}_j} = 1 - |positive(\mathbf{s}_i) - positive(\mathbf{s}_j)|$$

The similarity score for negative sentiment are computed as follows:

$$M_{\mathbf{s}_i,\mathbf{s}_j} = 1 - |negative(\mathbf{s}_i) - negative(\mathbf{s}_j)|$$

3.2 Continuous Vector Space Representations

Continuous vector space representations of words has a long history. Recently, the use of deep learning methods has given rise to a new class of continuous vector space models. Bengio et al. (2006) presented vector space representations for words that capture semantic and syntactic properties. These vectors can be employed not only to find similar words, but also to relate words using multiple dimensions of similarity. This means that words sharing some sense can be related using

[1] To download the sentiment word lists used, please see `http://www.mogren.one/`

translations in vector space, e.g. $v_{king} - v_{man} + v_{woman} \approx v_{queen}$.

Early work on extractive summarization using vector space models was presented in (Kågebäck et al., 2014). In this work we use a similar approach, with two different methods of deriving word embeddings. The first model (CW) was introduced by Collobert and Weston (2008). The second ($W2V$) is the skip-gram model by Mikolov et al. (2013).

The Collobert and Weston vectors were trained on the RCV1 corpus, containing one year of Reuters news wire; the skip-gram vectors were trained on 300 billion words from Google News.

The word embeddings are subsequently used as building blocks for sentence level phrase embeddings by summing the word vectors of each sentence. Finally, the sentence similarity is defined as the cosine similarity between the sentence vectors.

With MULTSUM, these similarity measures can be combined with the traditional sentence similarity measures.

4 Experiments

Our version of the submodular optimization code follows the description by Lin and Bilmes (2011), with the exception that we use multiplicative combinations of the sentence similarity scores described in Section 3. The source code of our system can be downloaded from `http://www.mogren.one/`. Where nothing else is stated, MULTSUM was evaluated with a multiplicative combination of TRComparer and FilteredWordComparer.

4.1 Datasets

In the evaluation, three different datasets were used. DUC 02 and DUC 04 are from the Document Understanding Conferences, both with the settings of task 2 (short multi-document summarization), and each consisting of around 50 document sets. Each document set is comprised of around ten news articles (between 111 and 660 sentences) and accompanied with four gold-standard summaries created by manual summarizers. The summaries are at most 665 characters long. DUC 04 is the de-facto standard benchmark dataset for generic multi-document summarization.

Experiments were also carried out on Opinosis (Ganesan et al., 2010), a collection of short user reviews in 51 different topics. Each topic consists of between 50 and 575 one-sentence user reviews by different authors about a certain characteristic of a hotel, a car, or a product. The dataset includes 4 to 5 gold-standard summaries created by human authors for each topic. The the gold-standard summaries is around 2 sentences.

4.2 Baseline Methods

Our baseline methods are Submodular optimization (Lin and Bilmes, 2011), DPP (Kulesza and Taskar, 2012), and ICSI (Gillick et al., 2008). The baseline scores are calculated on precomputed summary outputs (Hong et al., 2014).

4.3 Evaluation Method

Following standard procedure, we use ROUGE (version 1.5.5) for evaluation (Lin, 2004). ROUGE counts n-gram overlaps between generated summaries and the gold standard. We have concentrated on recall as this is the measure with highest correlation to human judgement (Lin and Hovy, 2003), on ROUGE-1, ROUGE-2, and ROUGE-SU4, representing matches in unigrams, bigrams, and skip-bigrams, respectively.

The Opinosis experiments were aligned with those of Bonzanini et al. (2013) and Ganesan et al. (2010)[2]. Summary length was 2 sentences. In the DUC experiments, summary length is 100 words[3].

5 Results

Our experimental results show significant improvements by aggregating several sentence similarity measures, and our results for ROUGE-2 and ROUGE-SU4 recall beats state–of–the–art.

5.1 Integrating Different Similarity Measures

Table 2 shows ROUGE recall on DUC 04. MULTSUM[4] obtains ROUGE scores beating state-of-the-art systems, in particular on ROUGE-2 and ROUGE-SU4, suggesting that MULTSUM produce summaries with excellent fluency. We also note, that using combined similarities, we beat original submodular optimization.

Figure 5.1 shows, for each $n \in [1..9]$, the highest ROUGE-1 recall score obtained by MULTSUM, determined by exhaustive search

[2]ROUGE options on Opinosis: -a - m -s -x -n 2 -2 4 -u.

[3]ROUGE options on DUC: -a -n 2 -m -l 100 -x -c 95 -r 1000 -f A -p 0.5 -t 0 -2 4 -u.

[4]Here, MULTSUM is using TRComparer and FilteredWordComparer in multiplicative conjunction.

	ROUGE-1	ROUGE-2	ROUGE-SU4
MULTSUM	39.35	**9.94**	**14.01**
ICSISumm	38.41	9.77	13.62
DPP	**39.83**	9.62	13.86
SUBMOD	39.18	9.35	13.75

Table 2: ROUGE recall scores on DUC 04. Our system MULTSUM obtains the best result yet for ROUGE-2 and ROUGE-SU4. DPP has a higher ROUGE-1 score, but the difference is not statistically significant (Hong et al., 2014).

1	**2**	**3**	**4**
1.0	0.00038	0.00016	0.00016

Table 3: p-values from the Mann-Whitney U-test for combinations of similarity measures of size $n \in [1..4]$, compared to using just one similarity measure. Using 2, 3, or 4 similarity measures at the same time with MULTSUM, gives a statistically significant improvement of the ROUGE-1 scores. Dataset: DUC 04.

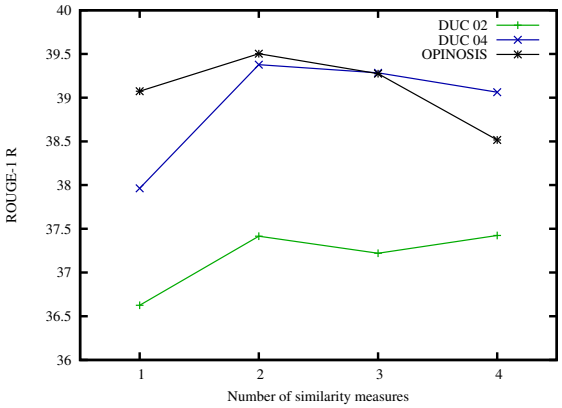

Figure 1: MULTSUM ROUGE-1 recall performance for each top-performing combination of up to four similarity measures. On all datasets, using combinations of two, three, and four similarity measures is better than using only one.

among all possible combinations of size n. The performance increases from using only one sentence similarity measure, reaching a high, stable level when $n \in [2..4]$. The behaviour is consistent over three datasets: DUC 02, DUC 04 and OPINOSIS. Based on ROUGE-1 recall, on DUC 02, a combination of four similarity measures provided the best results, while on DUC 04 and Opinosis, a combination of two similarity scores provided a slightly better score.

Table 3 shows p-values obtained using the Mann-Whitney U-test (Mann et al., 1947) on the ROUGE-1 scores when using a combination of n similarities with MULTSUM, compared to using only one measure. The Mann-Whitney U-test compares two ranked lists A and B, and decides whether they are from the same population. Here, A is the list of scores from using only one measure, and B is the top-10 ranked combinations of n combined similarity measures, $n \in [1..4]$). One can see that for each $n \in [1..4]$, using n sentence similarity measures at the same time, is significantly better than using only one.

On DUC 02, the best combination of similarity measures is using CW, LinTFIDF, NegativeSentiment, and TRComparer. Each point in Figure 5.1 represents a combination of some of these four similarity measures. Let n be the number of measures in such a combination. When $n = 1$, the "combinations" are just single similarity measures. When $n = 2$, there are 6 different ways to choose, and when $n = 3$, there are four. A line goes from each measure point through all combinations the measure is part of. One can clearly see the benefits of each of the combination steps, as n increases.

5.2 Evaluation with Single Similarity Measures

In order to understand the effect of different similarity measures, MULTSUM was first evaluated using only one similarity measure at a time. Table 4 shows the ROUGE recall scores of these experiments, using the similarity measures presented in Section 3, on DUC 04.

We note that MULTSUM provides summaries of high quality already with one similarity measure (e.g. with TRComparer), with a ROUGE-1 recall of 37.95 Using only sentiment analysis as the single similarity measure does not capture enough information to produce state-of-the-art summaries.

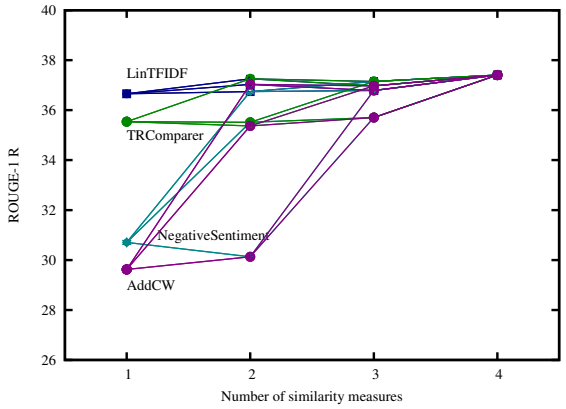

Figure 2: ROUGE-1 recall for the top-performing four-combination on DUC 2002 (CW, LinTFIDF, NegativeSentiment, and TRComparer), and all possible subsets of these four similarity measures. (When the number of similarity measures is one, only a single measure is used).

	ROUGE-1	ROUGE-2	ROUGE-SU4
TRComparer	**37.95**	**8.94**	**13.19**
Filtered	37.51	8.26	12.73
LinTFIDF	35.74	6.50	11.51
KeyWord	35.40	7.13	11.80
DepGraph	32.81	5.43	10.12
NegativeSent.	32.65	6.35	10.29
PositiveSent.	31.19	4.87	9.27
W2V	32.12	4.94	9.92
CW	31.59	4.74	9.51

Table 4: ROUGE recall of MULTSUM using different similarity measures, one at a time. Dataset: DUC 04. The traditional word-overlap measures are the best scoring when used on their own; the proposed measures with more semantical comparisons provide the best improvements when used in conjunctions.

6 Discussion

Empirical evaluation of the method proposed in this paper shows that using several sentence similarity measures at the same time produces significantly better summaries.

When using one single similarity at a time, using sentiment similarity and vector space models does not give the best summaries. However, we found that when combining several similarity measures, our proposed sentiment and continuous vector space measures often rank among the top ones, together with the TRComparer.

MULTSUM, our novel summarization method, based on submodular optimization, multiplies several sentence similarity measures, to be able to make summaries that are good with regards to several aspects at the same time. Our experimental results show significant improvements when using multiplicative combinations of several sentence similarity measures. In particular, the results of MULTSUM surpasses that of the original submodular optimization method.

In our experiments we found that using between two and four similarity measures lead to significant improvements compared to using a single measure. This verifies the validity of commonly used measures like TextRank and LinTFIDF as well as new directions like phrase embeddings and sentiment analysis.

There are several ideas worth pursuing that could further improve our methods. We will explore methods of incorporating more semantic information in our sentence similarity measures. This could come from systems for Information Extraction (Ji et al., 2013), or incorporating external sources such as WordNet, Freebase and DBpedia (Nenkova and McKeown, 2012).

7 Related Work

Ever since (Luhn, 1958), the field of automatic document summarization has attracted a lot of attention, and has been the focus of a steady flow of research. Luhn was concerned with the importance of words and their representativeness for the input text, an idea that's still central to many current approaches. The development of new techniques for document summarization has since taken many different paths. Some approaches concentrate on what words should appear in summaries, some focus on sentences in part or in whole, and some consider more abstract concepts.

In the 1990's we witnessed the dawn of the data explosion known as the world wide web, and research on multi document summarization took off. Some ten years later, the Document Understanding Conferences (DUC) started providing researchers with datasets and spurred interest with a venue for competition.

Luhn's idea of a frequency threshold measure for selecting topic words in a document has lived on. It was later superseded by $tf \times idf$, which measures the specificity of a word to a document,

The two bombers who carried out Friday's attack, which led the Israeli Cabinet to suspend deliberations on the land-for-security accord signed with the Palestinians last month, were identified as members of Islamic Holy War from West Bank villages under Israeli security control. The radical group Islamic Jihad claimed responsibility Saturday for the market bombing and vowed more attacks to try to block the new peace accord. Israel radio said the 18-member Cabinet debate on the Wye River accord would resume only after Yasser Arafat's Palestinian Authority fulfilled all of its commitments under the agreement, including arresting Islamic militants.

Table 5: Example output from MULTSUM. Input document: d30010t from DUC 04. Similarity Measures: W2V, TRComparer, and FilteredWordComparer.

something that has been used extensively in document summarization efforts. RegSum (Hong and Nenkova, 2014) trained a classifier on what kinds of words that human experts include in summaries. (Lin and Bilmes, 2011) represented sentences as a $tf \times idf$ weighted bag-of-words vector, defined a sentence graph with weights according to cosine similarity, and used submodular optimization to decide on sentences for a summary that is both representative and diverse.

Several other methods use similar sentence-based formulations but with different sentence similarities and summarization objectives (Radev et al., 2004; Mihalcea and Tarau, 2004).

(Bonzanini et al., 2013) introduced an iterative sentence removal procedure that proved good in summarizing short online user reviews. CLASSY04 (Conroy et al., 2004) was the best system in the official DUC 04 evaluation. After some linguistic preprocessing, it uses a Hidden Markov Model for sentence selection where the decision on inclusion of a sentence depends on its number of signature tokens. The following systems have also showed state–of–the–art results on the same data set. ICSI (Gillick et al., 2008) posed the summarization problem as a global integer linear program (ILP) maximizing the summary's coverage of key n-grams. OCCAMS_V (Davis et al., 2012) uses latent semantic analysis to determine the importance of words before the sentence selection. (Kulesza and Taskar, 2012) presents the use of Determinantal point processes (DPPs) for summarization, a probabilistic formulation that allows for a balance between diversity and coverage. An extensive description and comparison of these state–of–the–art systems can be found in (Hong et al., 2014), along with a repository of summary outputs on DUC 04.

Besides the aforementioned work, interested readers are referred to an extensive survey (Nenkova and McKeown, 2012). In particular, they discuss different approaches to sentence representation, scoring and summary selection and their effects on the performance of a summarization system.

8 Conclusions

We have demonstrated that extractive summarization benefits from using several sentence similarity measures at the same time. The proposed system, MULTSUM works by using standard kernel techniques to combine the similarities. Our experimental evaluation shows that the summaries produced by MULTSUM outperforms state-of-the-art systems on standard benchmark datasets. In particular, it beats the original submodublar optimization approach on all three variants of ROUGE scores. It attains state-of-the-art results on both ROUGE-2 and ROUGE-SU4, showing that the resulting summaries have high fluency. The results are statistically significant and consistent over all three tested datasets: DUC 02, DUC 04, and Opinosis.

We have also seen that sentence similarity measures based on sentiment analysis and continuous vector space representations can improve the results of multi-document summarization. In our experiments, these sentence similarity measures used separately are not enough to create a good summary, but when combining them with traditional sentence similarity measures, we improve on previous methods.

Acknowledgments

This work has been done within "Data-driven secure business intelligence", grant IIS11-0089 from the Swedish Foundation for Strategic Research (SSF). We would like to thank Gabriele Capannini for discussion and help and Jonatan Bengtsson for his work on sentence similarity measures.

References

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*, volume 463. ACM press New York.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*. Springer.

Jonatan Bengtsson and Christoffer Skeppstedt. 2012. Automatic extractive single document summarization. Master's thesis, Chalmers University of Technology and University of Gothenburg.

Marco Bonzanini, Miguel Martinez-Alvarez, and Thomas Roelleke. 2013. Extractive summarisation via sentence removal: condensing relevant sentences into a short summary. In *SIGIR*, pages 893–896.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167.

John M Conroy, Judith D Schlesinger, Jade Goldstein, and Dianne P O'leary. 2004. Left-brain/right-brain multi-document summarization. In *DUC 2004*.

Sashka T Davis, John M Conroy, and Judith D Schlesinger. 2012. Occams–an optimal combinatorial covering algorithm for multi-document summarization. In *ICDMW*. IEEE.

David Duvenaud. 2014. *Automatic model construction with Gaussian processes*. Ph.D. thesis, University of Cambridge.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of COLING*, pages 340–348. ACL.

Dan Gillick, Benoit Favre, and Dilek Hakkani-Tur. 2008. The icsi summarization system at tac 2008. In *Proceedings of TAC*.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of EACL*.

Kai Hong, John M Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. *LREC*.

Heng Ji, Benoit Favre, Wen-Pin Lin, Dan Gillick, Dilek Hakkani-Tur, and Ralph Grishman. 2013. Open-domain multi-document summarization via information extraction: Challenges and prospects. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 177–201. Springer.

Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. *Proceedings of (CVSC)@ EACL*, pages 31–39.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *arXiv:1207.6083*.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of EACL*, pages 514–522. ACL.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *ACL*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of NAACL/HLT*, pages 71–78.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proc. of the ACL-04 Workshop*, pages 74–81.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal*, 2(2):159–165.

Henry B Mann, Donald R Whitney, et al. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1):50–60.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 43–76. Springer.

Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo, and Genichiro Kikui. 2010. Optimizing informativeness and readability for sentiment summarization. In *Proceedings of ACL*, pages 325–330. ACL.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT*. Citeseer.

Dragomir R. Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drábek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. 2004. Mead - a platform for multidocument multilingual text summarization. In *LREC*.

Bernhard. Schölkopf, Koji. Tsuda, and Jean-Philippe. Vert. 2004. *Kernel methods in computational biology*. MIT Press, Cambridge, Mass.