Analyzing the Use of Word Graphs for Abstractive Text Summarization

Elena Lloret
Dept. of Software and Computing Systems
University of Alicante
Apdo. de correos, 99
E-03080, Alicante, Spain
Email: elloret@dlsi.ua.es

Manuel Palomar
Dept. of Software and Computing Systems
University of Alicante
Apdo. de correos, 99
E-03080, Alicante, Spain
Email: mpalomar@dlsi.ua.es

***Abstract*—This paper focuses on abstractive text summarization. Our aim is to explore to what extent new sentences generated employing a word graph-based method (which either compress or merge information) are suitable for producing abstracts. Moreover, in order to decide which of the new sentences should be included in the abstractive summary, an extractive text summarization approach is developed (i.e., COMPENDIUM), so that the most relevant abstractive sentences can be selected and extracted. As shown by the results obtained, this task is very challenging. However, preliminary experiments carried out prove that the combination of extractive and abstractive information is a more suitable strategy to adopt towards the generation of abstracts.**

***Keywords*-Human Language Technologies, automated retrieval and mining, automated content summarization, abstractive techniques, graph-based algorithms.**

I. Introduction

Currently, the necessity of having good systems and tools capable of dealing with all the information available in an efficient and effective manner is crucial to provide users with the specific information they are interested in. In light of this, Text Summarization (TS) is of great help since its main aim is to produce a condensed new text containing a significant portion of the information in the original text(s) [1].

The process of summarization can be divided into three stages [2]: *topic identification*, *topic interpretation* and *summary generation*. Extractive summarization relies on the selection of the most important sentences in order to produce the summary. As a consequence, only carry out the *topic identification* step is carried out. In contrast, abstractive approaches require a more elaborate process, involving sentence compression, information fusion, and/or language generation. In these cases, all the stages of the summarization process are taken into account.

Due to the difficulty associated to the generation of abstracts, most approaches only focus on the first stage (i.e., topic identification), producing extracts as a result [3], [4], [5], [6]. The main problem of extractive summarization, though, concerns the coherence of the resulting summaries, since the sentences contained may not be properly linked, and most of them will suffer from the well-known *dangling anaphora* phenomenon, i.e., when the pronouns in a summary do not refer to their correct antecedent. Consequently, in order to solve these limitations, research into abstractive methods is needed [7], [8], [9].

The aim of this paper is to conduct an analysis of the potentials and limitations of word graphs for generating abstractive summaries. We first propose a method for compressing and merging information based on word graphs, and then we generate summaries from the resulting sentences. This allows us to quantify how feasible it is to produce abstracts directly. The results obtained give clear proof of the difficulty of the task, and the challenges it presents. However, in a preliminary experiment, we show that a more appropriate strategy would be to combine extractive and abstractive information, improving the performance of the resulting summaries considerably.

The remaining of the paper is structured as follows: Section II introduces previous work in abstractive techniques. Section III describes the word graph-based method for compressing and merging sentences. Further on, how abstractive summaries are produced is explained in Section IV. Section V provides all issues concerning the experiments and evaluation. Additionally, Section VI shows a preliminary analysis of two proposed strategies in an attempt to solve the limitations found in the approach. Finally, the conclusions of the paper together with the future work are outlined in Section VII.

II. Related Work

In this section, we explain previous work on recent abstractive summarization, and we stress our novelty with respect to other similar approaches.

An approach for combining different fragments of information that have been extracted from one or more documents is suggested in [10]. From a predefined vocabulary (e.g., *to address*), the algorithm is able to decide which of these expressions is more appropriate for a sentence, depending on the content and the partial abstract generated. Using machine learning techniques and experimenting with different types of classifiers, results showed that the best classifier, based

 ISBN: 978-1-61208-162-5

on summarization features was able to correctly predict 60% of the cases.

Furthermore, sentence compression [11], [12], and sentence fusion [13], [14] are techniques that have also been applied to abstractive summarization. In particular, graph-based algorithms used for such purpose have been proven to be very successful for producing multi-document summaries [15], [16]. On the one hand, regarding sentence fusion, in [15], related sentences are represented by means of dependency graphs, and then the nodes of such graphs are aligned taking into account their structure. Then, Integer Linear Programming [17] is used to generate a new sentence, where irrelevant edges of the graphs are removed, and an optimal sub-tree is found employing structural, syntactic and semantic constraints. On the other hand, for sentence compression, Filippova [16] suggests a method based on word graphs, where the shortest path is computed to obtain a very short summary (only one sentence) from a set of related sentences belonging to different documents.

Liu and Liu [18] attempt to transform an extractive summarization into an abstractive one in the context of meeting summarization by performing sentence compression. Different compression algorithms, such as Integer Programming, Markov Grammars [19] or even human compression were evaluated, with the result that there are certain limitations when using only sentence compression for generating abstracts. With the same idea, Steinberger et al. [20] explore different ways to generate summaries from their representations through their most important sentences. Their aim is to remove unnecessary words from the original sentences, and then use a probabilistic approach to try to reconstruct them. This approach was found to obtain similar results to extractive summarization.

Our research focuses on studying the applicability of a compression and fusion strategy for producing abstractive summaries. We rely on word graphs for representing documents, and we use them to produce single-document abstracts, allowing the algorithm to compress and merge information.

III. Using Word Graphs for Generating New Sentences

In this section, we explain the proposed algorithm based on word graphs for generating new sentences. Such sentences can be either a compressed version of the original one, or a longer sentence containing information from several.

A. Building the Word Graph

A document is represented as a directed weighted graph $DG = (V, E)$, where $V = v_i, v_{i+1}, \ldots, v_{i+n}$ is the set of nodes corresponding to document's words, and $E = e_{i,i+1}, e_{i+1,i+2}, \ldots, e_{i-n,i+n}$ is the set of edges, which consists of adjacency relations between the words. For the implementation we used Python-graph library [21]. Two words are mapped into the same node only if they have the same part of speech by using TreeTagger [22]. It is important to stress that stop words are not mapped together; otherwise, the real meaning of the sentence could be changed when generating the new sentence. In the future, we plan to use semantic knowledge in order to be able to map concepts instead of words.

In addition, we have to define a weighting function $W(e_{i,i+1})$ for each edge, in order to determine how relevant the edge is. The proposed weight takes into account the frequency of occurrence (*FreqRel*) of two words together in the document, as well as the importance of the words themselves, which is determined through their PageRank value (*PR*) [23]. Therefore, the weighting function can be computed according to Formula 1.

$$W(e_{i,i+1}) = \frac{1}{FreqRel(v_i, v_{i+1}) * (PR(v_i) + PR(v_{i+1}))} \quad (1)$$

In Figure 1, a fragment of a graph is shown.

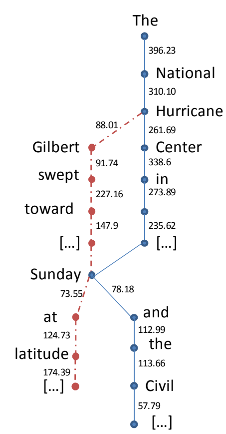

Figure 1. Example of a word graph representation

B. Obtaining New Sentences

In order to produce a new sentence, we employ Dijkstra's algorithm [24] to find the shortest paths between an initial node and the remaining ones that are directly or indirectly connected with it. We chose the shortest path algorithm because, on the one hand, it has been shown to be appropriate for compressing sentences in previous work [16], and on the other hand, the shortest path will also look for minimal-length sentences that contains information from several ones, thus allowing them to include more information.

Two strategies are proposed for defining a starting node to apply the searching algorithm over the document's graph representation:

- The initial node corresponds to the **first word in each sentence**. This manner, we ensure that for each sentence in the source document, we have at least one

derived sentence, so the whole content of the document is covered.
- The initial node corresponds to the **10 words with highest *tf-idf***. Term frequency-inverse document frequency (tf-idf) accounts for frequent terms in the document, but not very frequent in the whole collection of documents. With this strategy, we keep the most important terms and the information related to them.

C. Ensuring Sentences' Correctness

By applying the Dijkstra's algorithm over the graph we obtain all possible shortest paths between one node and the remaining ones. This leads to a high number of resulting sentences, which are not equally good. In fact, some of the sentences might be completely incomprehensible and not correctly formed. In order to guarantee the completeness and correctness of a new sentence, we define three basic constraints in order to discard those sentences, which do not satisfied all of them:

- The minimal length for a sentence must be 3 words (i.e., subject+verb+object).
- Every sentence must contain a verb.
- The sentence should not end in an article (e.g., a, the), a preposition (e.g., of), an interrogative word (e.g., who), nor a conjunction (e.g., and).

The remaining sentences after applying the aforementioned constraints will be used for building the abstractive summaries.

IV. Producing Abstractive Summaries

In order to use the new generated sentences (Section III) for building abstractive summaries, it is necessary to identify, which of them carry the most relevant information, since the length of the summary is restricted (in our case, to 100 words), thus not being possible to include all of them. Therefore, for determining important content we employed COMPENDIUM TS approach [25].

With the purpose of analysing whether or not it is better to generate new information before selecting the most important one, or the opposite (i.e., to extract relevant information first, and then generate new sentences from it), we apply COMPENDIUM in two different ways:

1) The set of new sentences obtained from the word graph-based method is the input for COMPENDIUM (***Graphs*+COMPENDIUM**).
2) The important content of the document is first selected, and then the word graph-based method is applied for generating new sentences derived from the extract (**COMPENDIUM+*Graphs***).

In both cases, the resulting summaries will be abstracts, since they do not reproduce verbatim the sentences of the source document.

V. Experimental Setup

In this section, we explain the dataset used, the experiments carried out as well as the results obtained together with an in-depth discussion.

A. Dataset

As dataset we randomly selected 50 documents of the DUC 2002 newswire corpus [26], each document having 500 words on average. Additionally, two model summaries written by humans are also provided for each document. These summaries have a length of approximately 100 words, which corresponds to a 20% compression rate with respect to the source documents.

B. Experiments and Results

In order to test the appropriateness of our suggested method for generating abstractive summaries, we followed the same guidelines as in DUC 2002 [27] (i.e., we produce generic single-document summaries of 100 words each) and we compare our abstractive summaries to the existing model summaries.

In addition to the two approaches explained in Section IV: ***Graphs*+COMPENDIUM** and **COMPENDIUM+*Graphs***, we define a ***baseline***, in which we generate the new sentences from the source document and select the first ones to build the abstractive summary, until the length of 100 words is reached.

Moreover, in order to broaden this analysis, we experiment with three heuristics concerning the length of the generated sentences:

- **ALL:** all generated sentences;
- **LONG:** only those sentences that are longer (in number of words) than the average length, and
- **SHORT:** only those sentences, which are shorter than the average length.

In total we analyze 18 types of abstracts: 2 strategies for generating new sentences, 3 summarization approaches, and 3 heuristics for selecting sentences with regard to their length, resulting in 900 different summaries (50 documents x 18 types).

For assessing the appropriateness of the generated abstractive summaries, we compare them to the model summaries employing the evaluation tool ROUGE [28]. In particular, we use the following metrics: ROUGE-1, ROUGE-2 and ROUGE-SU4, which account for the number of common unigrams, bigrams, and skip-bigrams with four words in-between at most, respectively. Tables I and II show the F-measure results for the abstractive summaries.

C. Discussion

As can be seen from both tables, results are not very high, though they are promising for further research, since they quantify how far we are from producing abstracts. They also help us to identify the limitations and the main challenges

 ISBN: 978-1-61208-162-5

Table I
RESULTS (F-MEASURE) OF THE ABSTRACTIVE SUMMARIES WHEN THE FIRST WORD OF EACH SENTENCE IS USED FOR GENERATING NEW SENTENCES.

Abstractive Approach	**R-1**	**R-2**	**R-SU4**
baseline-ALL	0.18726	0.04908	0.05967
baseline-LONG	0.19625	0.05029	0.06277
baseline-SHORT	0.20793	0.04877	0.06312
Graphs+COMPENDIUM-ALL	0.21609	0.05719	0.06951
Graphs+COMPENDIUM-LONG	0.22829	0.06187	0.07446
Graphs+COMPENDIUM-SHORT	0.21252	0.04808	0.06448
COMPENDIUM+Graphs-ALL	**0.29788**	**0.09663**	**0.11110**
COMPENDIUM+Graphs-LONG	0.29022	0.09660	0.10942
COMPENDIUM+Graphs-SHORT	0.16984	0.04633	0.05565

Table II
RESULTS (F-MEASURE) OF THE ABSTRACTIVE SUMMARIES WHEN THE TOP 10 WORDS WITH HIGHEST TF-IDF OF EACH SENTENCE ARE USED FOR GENERATING NEW SENTENCES.

Abstractive Approach	**R-1**	**R-2**	**R-SU4**
baseline-ALL	0.13058	0.03436	0.03957
baseline-LONG	0.14590	0.03729	0.04362
baseline-SHORT	0.15916	0.03604	0.04605
Graphs+COMPENDIUM-ALL	0.15668	0.04135	0.05042
Graphs+COMPENDIUM-LONG	0.17754	0.04554	0.05490
Graphs+COMPENDIUM-SHORT	0.17512	0.04228	0.05234
COMPENDIUM+Graphs-ALL	0.20850	0.06210	0.07048
COMPENDIUM+Graphs-LONG	**0.22323**	**0.06688**	**0.07647**
COMPENDIUM+Graphs-SHORT	0.18057	0.05186	0.05770

we need to face. A clear tendency is observed in the majority of the cases that the best results are obtained when the important information is first identified and extracted, and then the new sentences are generated (COMPENDIUM+*Graphs*). ROUGE results for COMPENDIUM+*Graphs-ALL* improve on average 55% with respect to the *Graphs*+COMPENDIUM-*ALL* approach when the first words of a sentence are used to generate the new sentences. The same approach but in the case of the top 10 words with highest tf-idf are used, leads to an improvement of 40% compared to the results obtained for *Graphs*+COMPENDIUM-*ALL*. Concerning the COMPENDIUM+*Graphs-ALL* and *baseline-ALL* approaches, the results of the former increase by 80% and 72%, for the first words or the top 10 words with highest tf-idf, respectively. In general, results are lower when the new sentences are generated from the words with highest tf-idf values. This is due to the fact that the summarization guidelines we followed together with the model summaries we had, focused on generic summarization, whereas our proposed strategy for generating new sentences using the top 10 words with highest tf-idf may be more appropriate to query-focused summarization, since this type of summary contains the most important information with regard to a specific topic, and consequently, the tf-idf method can provide some clues about the relevant topics of a document.

Now, by examining the content of the generated abstracts, we mainly focus on two types of problems. On the one hand, we try to elucidate the reasons why the *Graph*+COMPENDIUM approach performs worse than the COMPENDIUM+*Graph*, and on the other hand, we want to analyze the reasons of the low overall performance of the abstractive approaches.

Regarding the first type of analysis carried out, if we use the word graph-based method for generating new sentences first, and use all of them as input for COMPENDIUM, this TS tool can have difficulties in selecting important content. This occurs because many of the sentences will start with the same words (e.g., if we take the top 10 words with highest tf-idf), so once COMPENDIUM detects a specific fragment of information as relevant, sentences containing the same portion of information that have not been detected as redundant will be also selected, leading to summaries that have not much variation in content. In order to solve this limitation, besides checking for the correctness of the sentences once they have been generated and filtering out those ones, which do not satisfy the proposed constraints, we would also need to apply some constraints based on the information sentences contain, optimizing the set of generated sentences, so that only the best ones with respect to their content are used.

With respect to the general results of the abstractive approaches, since the length of the summaries is restricted to only 100 words, when selecting the most important sentences before or after generating new sentences, some of the concepts may not be included. Consequently, this affects the performance of the summaries, leading to low ROUGE results. Contrary to what was expected, longer sentences do not necessarily lead to better summaries, nor shorter sentences lead to more informative summaries. It happens the same problem as before: the concepts in the sentences may not present a great variation, focusing on a few topics, rather than providing an overview of the topics covered in the document. Finally, it is worth mentioning that producing pure abstracts is a challenging task, as it is shown also in previous research [18], where F-measure values for ROUGE-1 ranged from 13% to 18%.

VI. ADDRESSING THE LIMITATIONS OF THE APPROACH

The aim of this section is to conduct a preliminary analysis of the potential solutions to the problems previously identified.

A. Optimizing the Set of Generated Sentences

As it was previously stated, one possible solution for improving the selection the new generated sentences for taking part in the summary would be to find an optimization function that could provide us with the best generated sentences. In order to analyze if this could improve the final abstractive summaries, we carry out a preliminary experiment assuming an ideal case. We selected the 20% of the documents we used for our experiments, and we manually select the best sentences resulting from the word graph-based method. As before, we used such sentences either as input for COMPENDIUM, or we first extracted the relevant content and then we generated the sentences, from which we manually selected the best ones. Table III shows the results of this pilot experiment. We perform a t-test to account for the significance of the results for a 95% confidence interval (results which are statistically significant are marked with a star).

Table III
ROUGE-1 RESULTS FOR THE ABSTRACTIVE SUMMARIES ASSUMING AN IDEAL CASE.

Abstractive Approach	Recall	Precision	$F_\beta = 1$
Graphs+COMPENDIUM_firstWords	0.207	0.209	0.208
Graphs+COMPENDIUM_firstWords_ideal	0.279*	0.287*	0.283
COMPENDIUM+Graphs_firstWords	0.283	0.291	0.287
COMPENDIUM+Graphs_firstWords_ideal	**0.293**	**0.301**	**0.296**
Graphs+COMPENDIUM_top10tfidf	0.197	0.199	0.198
Graphs+COMPENDIUM_top10tfidf_ideal	0.255*	0.263*	0.259*
COMPENDIUM+Graphs_top10tfidf	0.271	0.220	0.218
COMPENDIUM+Graphs_top10tfidf_ideal	0.283*	0.292*	0.287*

Assuming this ideal case, the results are improved by 25% on average, with respect to the original approaches. Furthermore, the improvement is higher for the *Graphs*+COMPENDIUM approach (36% and 31%, for rows 1-2 and 5-6, respectively). As it was previously shown, it is more appropriate to determine relevant information first by means of an extractive TS approach, and then try to compress and combine such information. In an ideal case, results for COMPENDIUM+*Graphs* improve by 5% and 10% with respect to *Graphs*+COMPENDIUM when the first words or the 10 words with highest tf-idf values are used for generating new sentences, respectively.

B. Combining Extractive and Abstractive Information

Here we want to analyze to what extent the generated sentences can be used in combination with extracts corresponding to the same documents. Therefore, we again experimented with the 20% of the documents and we took as a basis the extractive summaries for each of them generated by COMPENDIUM. Further on, taking also into consideration the abstractive summaries produced, we combined both types of summaries, according to these rules: i) if the sentence in the extract has one or more equivalent sentences in the abstract, we substitute the former for the latter; ii) if the sentence in the extract does not correspond to any sentence in the abstract, we keep the sentence in the extract, and iii) if the abstract contains some sentences that are not present in the extract, we enrich the extract with these sentences. In this manner, the new summary produced contains both extractive and abstractive information. Table IV shows the preliminary results of this experiment. Statistical differences according to a t-test are indicated with a star.

As it can be seen from the results obtained, we can confirm that for generic summaries, it is better to generate the new sentences from the first words of each original sentence. Consequently, the summary will cover a wide range of topics. Moreover, results have improved considerably with respect to the ones obtained for the abstracts shown in Table III (62% on average). It is worth mentioning that when we take as a basis an extractive summary and we enrich it with abstractive information generated from the source document, F-measure results improve significantly compared to the initial extract. This is very positive result, since it indicates that we can carry out research into this type of summaries, improving the quality of them, as well as going beyond the simple selection of sentences.

Table IV
ROUGE-1 RESULTS FOR THE EXTRACTIVE+ABSTRACTIVE SUMMARIES.

Approach	Recall	Precision	$F_\beta = 1$
Extractive summary (ES)	**0.491**	0.456	0.472
ES+Graphs+COMPENDIUM_firstWords_ideal	0.471	**0.492**	**0.480***
ES+COMPENDIUM+Graphs_firstWords_ideal	0.426	0.458	0.441
ES+Graphs+COMPENDIUM_top10tfidf_ideal	0.458	0.456	0.457
ES+COMPENDIUM+Graphs_top10tfidf_ideal	0.405	0.436	0.419

VII. CONCLUSION AND FUTURE WORK

In this paper, we analyzed a method based on word graphs for generating abstractive summaries. The purpose of the method was to compress and merge information from sentences. In order to decide which of the new sentences should be included in the abstractive summary, we employed an extractive TS approach (i.e., COMPENDIUM), so that the most relevant sentences could be selected and extracted. We analyzed different strategies for generating abstracts, including the most appropriate way to generate new sentences, the order to select important information, and the length of the sentences. The results obtained, although encouraging, showed the difficulty of the task itself, and brought some insights of the problems with the resulting abstracts. In light of this, we conducted two additional preliminary experiments to analyze how to improve the resulting summaries. The main conclusion we can draw from this research is that the word graph-based method proposed is appropriate to generate abstractive information that can be later used to enrich extractive information, influencing positively in the resulting summaries.

Nevertheless, there is still a lot of room for improvement, so several actions have to be taken for further work. In the short-term, we plan to increase the corpus size and carry out the same experimentation with more documents, improving also the word-graph method. Moreover, we want to verify if the proposed strategy for generating new sentences taking into account the words with highest tf-idf could be appropriate for query-focused summarization. In the long-term, we are interested in analyzing other methods for representing information and how it can be generalized (e.g., concept graphs).

ACKNOWLEDGMENT

This research has been supported by the FPI grant (BES-2007-16268) from the Spanish Ministry of Science and Innovation, under the project TEXT-MESS (TIN2006-15265-C06-01) and project grant no. TIN2009-13391-C04-01, both funded by the Spanish Government. It has been also funded by the Valencian Government (grant no. PROMETEO/2009/119 and ACOMP/2011/001).

REFERENCES

[1] K. Spärck Jones, "Automatic summarising: The state of the art," *Information Processing & Management*, vol. 43, no. 6, pp. 1449–1481, 2007.

[2] E. Hovy, *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2005, ch. Text Summarization, pp. 583–598.

[3] D. S. Leite, L. H. M. Rino, T. A. S. Pardo, and M. d. G. V. Nunes, "Extractive automatic summarization: Does more linguistic knowledge make a difference?" in *Proceedings of the 2nd Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*. ACL, 2007, pp. 17–24.

[4] P. Lal and S. Rüger, "Extract-based summarization with simplification," in *Workshop on Text Summarization in conjunction with the ACL*, 2002.

[5] M. Liu, W. Li, and Q. Wu, M.ingli andLu, "Extractive summarization based on event term clustering," in *Proceedings of the 45th Annual Meeting of the ACL*, 2007, pp. 185–188.

[6] K.-F. Wong, M. Wu, and W. Li, "Extractive summarization using supervised and semi-supervised learning," in *Proceedings of the 22nd International Conference on Computational Linguistics*, 2008, pp. 985–992.

[7] G. Carenini and J. C. K. Cheung, "Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality," in *Proceedings of the 5th International Natural Language Generation Conference, ACL*, 2008, pp. 33–40.

[8] C. Sauper and R. Barzilay, "Automatically generating wikipedia articles: A structure-aware approach," in *Proceedings of the 47th Association of Computational Linguistics*, 2009, pp. 208–216.

[9] H. Saggion, "Learning predicate insertion rules for document abstracting," in *Computational Linguistics and Intelligent Text Processing*, ser. LNCS, 2011, vol. 6609, pp. 301–312.

[10] ——, "A classification algorithm for predicting the structure of summaries," in *Proceedings of the Workshop on Language Generation and Summarisation*. ACL, 2009, pp. 31–38.

[11] D. Zajic, B. J. Dorr, J. Lin, and R. Schwartz, "Multi-candidate reduction: Sentence compression as a tool for document summarization tasks," *Information Processing & Management*, vol. 43, no. 6, pp. 1549–1570, 2007.

[12] J. Clarke and M. Lapata, "Models for sentence compression: a comparison across domains, training requirements and evaluation measures," in *Proceedings of the 44th Annual Meeting of the ACL*, 2006, pp. 377–384.

[13] R. Barzilay and K. R. McKeown, "Sentence fusion for multidocument news summarization," *Computational Linguistics*, vol. 31, pp. 297–328, 2005.

[14] E. Marsi and E. Krahmer, "Explorations in sentence fusion," in *Proceedings of the 10th European Workshop on Natural Language Generation*, 2005.

[15] K. Filippova and M. Strube, "Sentence fusion via dependency graph compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 177–185.

[16] K. Filippova, "Multi-sentence compression: Finding shortest paths in word graphs," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 322–330.

[17] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, Inc., 1986.

[18] F. Liu and Y. Liu, "From extractive to abstractive meeting summaries: can it be done by sentence compression?" in *Proceedings of the ACL-IJCNLP Conference Short Papers*. ACL, 2009, pp. 261–264.

[19] M. Galley and K. McKeown, "Lexicalized markov grammars for sentence compression," in *Proceedings of the Human Language Technology Conference of the NAACL*, 2007, pp. 180–187.

[20] J. Steinberger, M. Turchi, M. Kabadjov, R. Steinberger, and N. Cristianini, "Wrapping up a summary: From representation to generation," in *Proceedings of the ACL 2010 Conference*, 2010, pp. 382–386.

[21] Python-graph, "http://code.google.com/p/python-graph/," Last access: July, 2011.

[22] TreeTagger, "http://www.ims.uni-stuttgart.de/projekte/corplex/treetagger/," Last access: July, 2011.

[23] S. Brin and L. Page, "The Anatomy of a Large-scale Hypertextual Web Search Engine," *Computer Networks ISDN Systems*, vol. 30, pp. 107–117, 1998.

[24] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[25] E. Lloret, "Text summarisation based on human language technologies and its applications," Ph.D. dissertation, University of Alicante, Spain, June 2011.

[26] DUC Past Data, "http://www-nlpir.nist.gov/projects/duc/data.html," Last access: July, 2011.

[27] DUC, "http://www-nlpir.nist.gov/projects/duc/guidelines.html," Last access: July, 2011.

[28] C.-Y. Lin, "ROUGE: a package for automatic evaluation of summaries," in *Proceedings of ACL Text Summarization Workshop*, 2004, pp. 74–81.