Using coreference links and sentence compression in graph-based summarization

Iris Hendrickx
CNTS – Language Technology Group,
University of Antwerp,
Antwerp, Belgium
iris.hendrickx@ua.ac.be

Wauter Bosma
CLTL – Comp. Ling. and Term. Lab,
Vrije Universiteit,
Amsterdam, The Netherlands
w.bosma@let.vu.nl

Abstract

Recent years have shown that graphs are an adequate text representation model for summarization. For this years' TAC update summarization challenge, we extended our graph-based summarization system with coreference relations and sentence compression. Our results show that using coreference relations did not result in a significant performance gain; sentence compression had a negative effect on performance. We participated in the opinion summarization task with our base graph-based system. The measured performance of our opinion summarization system was competitive with respect to responsiveness, and poor with respect to linguistic quality.

1 Introduction

The series of document understanding conferences has lead to a solid basis for experimenting (benchmark data, evaluation metrics) and an active community in the field of automatic summarization. In this paper we describe our approach for the TAC 2008 update summary challenge as well as the opinion summarization pilot.

The update task assumes that a user has already read several articles about a certain topic. The aim is to produce small (100 words) update summaries that only contain new information. The organizers provide the participants with two consecutive document sets. The first set of documents is assumed to be read. This information should be left out of the update summary. In practice the task is to create two summaries. The first step is to extract a query-based summary of the first document set. The secons step is to create an update summary that does not overlap in content with the first summary.

The opinion summarization task requires the system to generate a summary, answering a set of (one or two) questions, using a set of source documents. If more than one question is provided, the questions typically ask for opposite opinions. An example of such a question is: *What did American voters admire about Rudy Guiliani?* There are two variants of the task: one with and one without a list of answers provided as input to the summarization system. In the variant in which answers are provided, these short text snippets are extracted by question answering systems in the TAC question answering task. The summarization system can then focus on generating a summary without repeating the search for answers, which is already done by question answering systems.

We present our graph-based summarization system (Bosma, 2008), extended with coreference relations and sentence compression. We consider document structure as an essential clue to create a summary of a set of documents. We model both the internal structure of a document and cross-document relations in a weighted graph model. The types of relations that we model are *relevancy* and *redundancy*. Besides cosine similarity as weighting score, we also use coreference information in our graph model. We construct summaries by extracting a subset of sentences from the original documents.

Coreference resolution (entity detection and tracking) can be a helpful information source to determine what a text is about. Important entities in a text are mentioned multiple times throughout the text and tracking these patterns an point to the topic of the text. Furthermore, the presence of an important entity in a particular sentence may indicate that this might be an important sentence that should be extracted. Also from a linguistic viewpoint, coreference information can help to maintain the readability and coherence in a text. Indeed the literature shows us that coreference information is helpful for the task of automatic summarization (Baldwin and Morton, 1998; Steinberger et al., 2007; Bergler et al., 2003; Witte et al., 2006).

We use sentence compression as a post processing step in our system. The aim of this compression is to remove unimportant phrases or words from sentences and to increase the information density in the summary. As the task is to create small update summaries, we aim to produce summaries consisting of shorter, denser sentences than the sentences in the original documents. The use of sentence compression for automatic summarization has been investigated previously in automatic summarization systems (Jing, 2000; Knight and Marcu, 2002; Lin, 2003; Conroy et al., 2006; Zajic et al., 2007).

2 Graph-based summarization

We decompose the summarization process in a number of sub tasks. This system presents a framework which allocates these sub tasks to different components, which allows for substituting part of the system while leaving everything else unchanged. This allows us to use the same summarization system for different summarization applications with only minor effort. We used the system previously for generic summarization and query-based summarization (Bosma, 2008). Separate modules are responsible for the following sub tasks.

Segmentation. Both source documents and query are segmented into *content units*. A content unit is an undividable text passage which is candidate for inclusion in the summary. In the TAC experiments, we use sentences as content units. The sentence segmenter retains meta information, such as from which document and which paragraph the sentence originates.

Feature extraction. The documents and the query are processed and converted to a feature graph to prepare for content selection. Content units are represented as nodes in the graph. The feature graphs are characterized by their weighted edges expressing the level of relevance and redundancy of individual content units. Multiple modules may be used in parallel with the effect that multiple graphs are generated. These generated graphs are integrated into a single model. In the TAC experiments, we use cosine similarity and coreference relations to build the feature graphs. The feature extraction module is described in more detail in the next section.

Salience estimation. This module consists of an algorithm that computes a salience value for each sentence from the (combined) feature graph. For each content unit, a salience value (in the range of 0 to 1) is calculated iteratively from the feature graphs. We use the 'probabilistic centrality' method, described elaborately in (Bosma, 2008). In short, the salience value is positively affected by relevance edges targeting the content unit. The greater the salience of a content unit, the greater its effect on the salience of neighboring content units. The content units of the query have a salience value of 1, regardless incident edges. Thus, salience propagates through the graph, starting from query content units and using relevance edges. To avoid redundancy in the summary, the salience of a content unit is reduced by incoming redundancy edges coming from a higher salient node. Finally, the most salient content units are selected for inclusion in the summary.

Presentation. A summary is created using the most salient content units. The most basic presentation method we use is to generate an extract consisting of the set of most salient sentences. Our runs with sentence compression include compressed sentences instead of verbatim sentences to fit more sentences within the word limit. If possible, the linear ordering of the sentences in the source text is retained. If the sum-

mary contains sentences from multiple sources documents, sentences from the document containing the largest number of sentences are presented first.

In the update summarization task, there are three classes of input text: the query, the first document set representing old information and the second document set representing the new information. Each content unit is annotated with its input class to enable modules to discriminate sentences, depending on the input class. The query expresses the user's information need; the second document set contains content units which are candidates for inclusion in the summary; the first document set (old information) is used to calculate the level of redundancy of candidate content units.

In the opinion summarization pilot we only have two classes of input: the query and the document set. In this pilot we submitted two runs; one with and one without using text snippets retrieved by question answering systems in the context of the QA track of TAC. These answer snippets are known to contribute to the query. Where the snippets were used, the snippets comprised the query in the summarization process. Where the snippets were not used, the questions comprised the query. In both cases, we used plain query-based summarization.

2.1 Feature extraction

The feature graphs represent relations between content units of different strengths. There are two types of feature graphs: one representing 'relevancy', and one representing 'redundancy'. A feature graph has a node for each content unit, and directed edges between any pair of nodes. The strength of an edge in the relevancy graph roughly corresponds to the probability that the target unit of the edge is relevant, given the source unit is relevant, i.e. $P(\text{relevant}(target)|\text{relevant}(source))$. The strength of an edge in the redundancy graph corresponds to the probability that the target unit is redundant, given the source unit is read, i.e. $P(\text{redundant}(target)|\text{known}(source))$. The strengths of all edges are in the range of 0 (weak) to 1 (strong). We generate the following feature graphs.

Query-relevance: the cosine similarity value between each content unit of the query and each candidate (i.e. source document) content unit (relevance graph).

Relatedness: the cosine similarity value between candidate content units of the same source document (relevance).

Cross-document relatedness: the cosine similarity value between candidate content units of different source documents (redundancy).

Redundancy: the cosine similarity value between content units of read documents (i.e. the first document set in update summarization) and candidate content units (redundancy).

Coreference: derived from the number of shared coreferences between candidate content units of the same document (relevance).

An added feature we implemented for this year's update summarization task is automatically recognized coreferential relations. The coreference resolution system predicts coreference relations between noun phrases (pronouns, proper nouns and common nouns) in the text. Each anaphoric noun phrase (NP) is linked to its antecedent. We use these coreferential links as a kind of discourse information: sentences which share a coreferential link are modeled as connected nodes in the feature graph. For each coreferential link which crosses a sentence boundary, an edge of strength 0.5 is added to the coreference graph. If there are several different coreference relations between two sentences, multiple edges are added. This is possible because the system allows edges to share an origin and a target node.

The coreference module (Hoste, 2005; Hendrickx et al., 2007) takes a standard machine learning approach following (Soon et al., 2001). This approach requires a corpus annotated with coreferential links between NPs. Instances are created between every NP (potential anaphor) and all of its preceding NPs (potential antecedents). The task of the classifier is to label each pair of NPs as coreferential or not. Instances describe the relation between a potential anaphor and its antecedent and are labeled positive when the NPS are coreferential and negative otherwise. The system is trained on the MUC-6 data set (Grishman and Sundheim, 1995)

and for each NP pair we created a feature set encoding morphological-lexical, syntactic, semantic, string matching and positional information.

2.2 Sentence compression

We use sentence compression as a post processing step to fit more information into our summary. Each selected sentence is compressed with a compression rate of 25%. We use a sentence compression module developed in the MUSA project[1]. This module was developed for compressing spoken text for auto-cue subtitling. The MUSA compressor relies on hand crafted phrase deletion rules to simplify sentences (Daelemans et al., 2004).

The compressor works as follows. First sentences are shallow parsed using a cascade of processing modules: after tokenization, each token in the sentence is assigned a part-of-speech tag and its lemma. Next the sentence is split into phrase chunks and a relation finder determines the grammatical relations between verbal chunks and noun chunks (for example 'object' or 'modifier' relations). The tagger, lemmatizer, chunker and relation finder all use the memory-based tagger MBT (Daelemans et al., 1996) trained on the Penn Tree bank (Marcus et al., 1993).

The simplification process has two steps. The deletion rules determine which words or phrases are candidates to be removed. Each candidate gets an importance weight. In a second step the lowest weighted candidates are removed until the desired compression rate is met. The deletion rules aim to remove the least informative phrases such as interjunctions, adverbs, adjectives, first names or prepositional phrases. The importance weight assigned to each candidate phrase expresses the summed log likelihood of frequencies of words in the phrase estimated on a large corpus. In Table 1 we give as an example the compression of a sentence taken from the TAC 2008 data set. For this sentence the compressor aims to remove 8 words (25%). After shallow parsing, the MUSA selection rules mark the deletion candidates and assign weights. The first deletion candidate, the preposition *[of five million Canadian dollars]* gets the lowest weight of 11.5. The second deletion candidate is the phrase *[in South Asia]*. Both are deleted in the output sentence.

When testing the MUSA compression on DUC 2006 newspaper text, it became clear that the module often removed prepositional phrases for which the preposition was actually a verb particle. For example, in a phrase like *[according [to the president]]* the preposition *to+NP* would be removed. To prevent this type of errors we added an extra restriction to MUSA. We used a pp-attachment module that classified each preposition as depending on a verb or a noun, and only noun-prepositions could be selected as candidates for removal.

3 Update summarization

3.1 Experimental setup

We trained our graph-based system on the DUC 2006 data set for query-based summarization. We used genetic algorithms to tune the weights in the multigraph, using Rouge-2 as a fitness function.

The data set consists of 48 topics, and 2 document-sets (old, new) per topic. NIST provided baseline summaries as well as 4 human summaries per document set as gold standard evaluation set and coordinated both the automatic and manual evaluation. The baseline summaries include the leading sentences the most recent documents, up to the word limit of 100 words.

We submitted three different variants of our system. The first variant contains all ingredients and is named *cosim+coref+sc*. This system uses both the cosim similarity and coreference information in the graph-based model and it uses the sentence compression post processing. The second variant (*cosim+coref*) excludes the sentence compression module. The third variant (*cosim*) only uses cosine similarity feature graphs for summarization.

3.2 Results

Tables 2, 3 and 4 list the scores that are computed automatically with the Rouge-2, Rouge-SU4 and Basic Element evaluation metrics respectively. In addition to our runs, the tables show results for the baseline system (leading sentences) and the best values achieved by any system. The bracketed numbers represent its performance rank among 72 submitted runs.

Our runs *cosim* and *cosim+coref* show similar

[1]MUSA project page: http://sifnos.ilsp.gr/musa/

Input sentence:	*A Canadian couple on Monday stunned the Canadian Red Cross by handing over a donation of five million Canadian dollars (4.1 million US) for the tsunami relief effort in South Asia .*
Weighted candicates:	*A [4(12.4) Canadian] couple [3(12.1) on Monday] stunned the Canadian Red Cross by handing over a donation [1(11.5) of five million [5(12.4) Canadian] dollars] (4.1 million US) [6(13.5) for the tsunami relief effort] [2(11.7) in South Asia] .*
Compressed output:	*A Canadian couple on Monday stunned the Canadian Red Cross by handing over a donation (4.1 million US) for the tsunami relief effort .*

Table 1: Example of input, weighting scheme and output of the MUSA sentence compression module.

score	Overall	Initial	Update
best	0.104	0.111	0.101
cosim	0.073 (42)	0.086 (23)	0.060 (52)
cosim+coref	0.073 (44)	0.086 (24)	0.060 (54)
cosim+coref+sc	0.056 (63)	0.066 (61)	0.046 (63)
baseline	0.059	0.058	0.060

Table 2: Average Rouge-2 scores for the three runs of our system in the update summarization task, compared to the baseline and the highest achieved scores.

performance, well above the baseline. From a closer inspection of the generated summaries, we learned that these two system configurations tend to select the same sentences in the extracts in the majority of the cases. The results of the *cosim+coref+sc* run are considerably lower than the other two runs. When we study the columns *Initial* and *Update*, we observe a clear difference in performance. In all the cases, our system runs score better on the initial summaries than on the update summaries.

The results of the manual evaluation are presented in the Tables 5 (Pyramid evaluation), 6 (linguistic quality) and 7 (responsiveness). The bracketed numbers represent the performance ranks among 58 manually evaluated runs. Unfortunately only two of our three variants could be evaluated manually. With respect to the Pyramid evaluation both of our runs scored above the baseline. Regarding linguistic quality, the baseline system outperformed the submitted runs of all participants. The *cosim+coref* run beats the baseline for responsiveness, but the *cosim+coref+sc* (with sentence compression) did not. Interestingly, the run with sentence compression scored consistently lower than the run without sentence compression. The comparison between the initial and update summaries show the same tendencies as the automatically computed scores. Again, the results for the initial summaries are better.

score	Overall	Initial	Update
best	0.136	0.0715	0.0685
cosim	0.110 (45)	0.121 (28)	0.100 (54)
cosim+coref	0.110 (46)	0.122 (27)	0.098 (55)
cosim+coref+sc	0.091 (63)	0.099 (65)	0.083 (63)
baseline	0.093	0.093	0.094

Table 3: Average Rouge-SU4 scores for the three runs of our system in the update summarization task, compared to the baseline and the highest achieved scores.

score	Overall	Initial	Update
best	0.065	0.064	0.076
cosim	0.045 (38)	0.052 (15)	0.037 (50)
cosim+coref	0.045 (40)	0.052 (16)	0.037 (53)
cosim+coref+sc	0.032 (62)	0.036 (58)	0.027 (63)
baseline	0.032	0.030	0.035

Table 4: Average Basic Elements (BE) scores for the three runs of our system in the update summarization task, compared to the baseline and the highest achieved scores.

score	Overall	Initial	Update
best	0.336	0.362	0.344
cosim+coref	0.234 (34)	0.227 (27)	0.191 (38)
cosim+coref+sc	0.194 (45)	0.224 (45)	0.164 (46)
baseline	0.167	0.186	0.147

Table 5: Pyramid scores for two runs of our system, compared to the baseline and the highest achieved scores.

score	Overall	Initial	Update
best	3.073	3.000	3.208
cosim+coref	2.292 (36)	2.417 (25)	2.167 (39)
cosim+coref+sc	1.625 (55)	1.667 (54)	1.583 (53)
baseline	3.334	3.250	3.417

Table 6: Linguistic quality evaluation for two runs of our system, compared to the baseline and the highest achieved scores.

score	Overall	Initial	Update
best	2.667	2.792	2.604
cosim+coref	2.188 (31)	2.354 (30)	2.021 (31)
cosim+coref+sc	1.740 (50)	1.875 (51)	1.604 (48)
baseline	2.073	2.292	1.854

Table 7: Responsiveness rating for two runs of our system, compared to the baseline and the highest achieved scores.

4 Opinion summarization

4.1 Experimental setup

The opinion test set consists of 22 topics[2], and one or more questions per topic. For each topic, one summary is generated with a maximum of 7000 non-whitespace characters per question. The questions are requests for opinions. The following is an example of two questions of the same topic:

Why do people like George Clooney?

Why do people dislike George Clooney?

Furthermore, answer snippets are provided by NIST for each topic. The snippets are answers to the questions and are retrieved by question answering systems in the TAC question answering track. Both form and quality of the snippets varies. The following is an example of two answer snippets related to the questions above.

he's a great actor

January 07, 2006 George Clooney Terrorist-Lover Not one penny of mine will go for any movie featuring George Clooney

We submitted summaries generated by two variants of our system. One of them ignores the answer snippets and uses the questions as the query. The second variant uses answer snippets provided by NIST as the query. Due to time constraints, both systems use the cosine similarity feature graphs, and do not use coreference graphs or sentence compression.

[2]The original test set consisted of 25 topics, but only 22 were evaluated by NIST.

score	Pyramid (rank)	Ling. qual.	Resp.
best	0.534	4.909	5.318
cosim	0.186 (12)	3.409 (16)	4.318 (10)

Table 8: Results for our opinion summarization system, *with* use of answer snippets.

score	Pyramid (rank)	Ling. qual.	Resp.
best	0.251	5.318	3.909
cosim	0.133 (12)	3.182 (17)	3.455 (2)

Table 9: Results for our opinion summarization system, *without* use of answer snippets.

4.2 Results

In the opinion task, each summary was evaluated for content using the nuggets Pyramid method used to evaluate the squishy list questions in the TAC QA task. In addition, an overall responsiveness score was assigned to each summary, as well as several ratings of the linguistic quality of the summary (grammaticality, non-redundancy, coherence). The responsiveness and linguistic quality ratings ranged from 1 to 10 (worst to best).

The results of our system are presented in Table 8 (with snippets) and Table 9 (without the use of snippets). Among the evaluated runs were 17 submissions which made use of snippets and 19 who didn't. The measured performance of our system was competitive with respect to responsiveness, and poor with respect to linguistic quality. Surprisingly, the Pyramid results seem to contradict the responsiveness ratings: the version of our system which did not use answer snippets scored 2nd best for responsiveness, while its Pyramid score was mediocre.

5 Conclusion

In update summarization, adding sentence compression as post-processing hurts the performance both on information content and on linguistic quality. This is in contrast to the related work mentioned in the introduction but it is in line with the work of (Lin, 2003) who reports on a pilot study in which automatic sentence compression did not improve the performance of the summarization system. Our results did not show any effect of adding predicted coreferential relation information.

For the opinion summarization task, we used our basic system without coreference graphs and sentence compression. The system performed well on responsiveness but poorly on linguistic quality.

For future research we plan to train and tune the parameters on a larger data set (DUC 2005, 2006, 2007 data). Another point that we would like to investigate is the sentence compression module. In this study we applied sentence compression as post-processing step. In that case erroneously compressed sentences can not be checked and are just added to the summary. If we use the module as a pre-processing step, the actual (compressed) candidate sentences are used for content selection, rather than the full sentences.

6 Acknowledgments

This work was conducted within the DAESO project funded by the Stevin program (De Nederlandse Taalunie).

References

B. Baldwin and T. S. Morton. 1998. Dynamic coreference-based summarization. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*, pages 1–6.

S. Bergler, R. Witte, M. Khalife, Z. Li, and F. Rudzicz. 2003. Using Knowledge-poor Coreference Resolution for Text Summarization. In *Proceedings of the 2006 Document Understanding Conference (DUC 2003)*.

W. Bosma. 2008. Discourse oriented summarization. Ph.D. thesis, Twente University, Twente.

J. M. Conroy, J. D. Schlesinger, D. OĹeary, and J. Goldstein. 2006. Back to basics: Classy 2006. In *Proceedings of the 2006 Document Understanding Conference (DUC 2006)*.

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In *Proceedings of the 4th ACL/SIGDAT Workshop on Very Large Corpora*, pages 14–27.

W. Daelemans, A. Hothker, and E. Tjong Kim Sang. 2004. Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1045–1048.

I. Hendrickx, V. Hoste, and W. Daelemans. 2007. Evaluating hybrid versus data-driven coreference resolution. *Lecture Notes in Artificial Intelligence. Anaphora: Analysis, Algorithms and Application*, 4410:137–150.

V. Hoste. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. Ph.D. thesis, Antwerp University, Antwerp.

H. Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000)*, pages 310–315.

K. Knight and D. Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

C.-Y. Lin. 2003. Improving summarization performance by sentence compression: a pilot study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages*, pages 1–8.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

R. Grishman and B. Sundheim, 1995. Appendix D, Coreference task definition. version 2.3. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 335–344.

W. M. Soon, H. T. Ng, and D. C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

J. Steinberger, M. Poesio, M. A. Kabadjov, and K. Jeek. 2007. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6):1663–1680.

R. Witte, R. Krestel, and S. Bergler. 2006. Context-based Multi-Document Summarization using Fuzzy Coreference Cluster Graphs. In *Proceedings of Document Understanding Workshop (DUC)*, New York City, NY, USA, June 8–9.

D. Zajic, B. J. Dorr, J. Lin, and R. Schwartz. 2007. Multicandidate reduction: Sentence compression as a tool for document summarization tasks. In *Information Processing and Management*, 43(6):1549–1570