# Cascaded Filtering for Topic-Driven Multi-Document Summarization

**Katja Filippova, Margot Mieskes, Vivi Nastase, Simone Paolo Ponzetto, Michael Strube**
EML Research gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
http://www.eml-research.de/nlp

## Abstract

This paper presents EMLR's NLP group's first participation in the DUC summarization competitions. Our system combines document filtering, ranking sentences using lexical chains and graph matching algorithms with the topic, on top of several annotation layers in the MMAX2 annotation tool. The system ranked 14 out of 30 participating teams in manual annotation, and had particularly good ranking from the linguistic quality point of view.

## 1 Introduction

EMLR has participated in the main task of the DUC 2007 competition, for topic-driven multi-document summarization. As in previous years (2005, 2006), the task was to produce a 250 words summary for each of a number of given topics. Each topic was associated with a preselected set of 20-30 documents, to be used as a source for the produced summary.

EMLR's system (SYSID 5) is an extractive summarizer, which combines several layers of annotation and filtering algorithms. The system is built on top of the MMAX2 annotation tool[1], which facilitates work with data having multiple layers of annotations. Our extractive summarizing technique is based on several filtering steps, both at the document and sentence level.

The first filtering step is at the document level, where from the given set of about 30 documents per topic, we select a small number, based on their degree of matching with the corresponding topic. Reducing the number of documents to be processed in future steps allows us to perform deeper semantic analysis, which otherwise would be very time consuming.

The next filtering stage acts on a sentence level. Sentences are scored based on information from lexical chains, rooted in the nouns and verbs from the topic. The information about lexical chains intersecting a document sentence is used to compute multiple scores, equivalent to n-grams (n = 1..3), where the elements of the n-grams are points of intersection of the chains with the sentence.

The best ranked sentences obtained after lexical chain-based scoring are passed further to the next filtering step. At this stage, both the topic and the sentences are transformed into a dependency graph/tree, using grammatical dependencies information. Each sentence is newly scored based on the match between its dependency tree representation and the topic's graph[2]. This scoring reranks the list of sentences obtained after lexical chain filtering. From this list, our system incrementally generates a summary from the highest ranked sentences, checking for redundancy before each addition.

The MMAX2 system and the annotation layers that are the scaffolding for our summarization system are presented in Section 2. The document filtering stage is described in Section 3. We show how lexical chains are built and used for sentence scoring in Section 4. The last filtering step and generation of the summary is presented in Section 5. In Section 6 we present and discuss the results obtained.

[1] Available at `http://mmax.eml-research.de`

[2] If the topic contains several sentences, they are put together to obtain a single dependency graph representation.

## 2 Annotation layers in MMAX2

Multi-document summarization is a high-level, discourse oriented NLP task: as a result, it requires a significant amount of linguistic information. The preprocessing steps required by our system include:

- tokenization, based on the Penn Treebank tokenization conventions (Marcus et al., 1993);
- lemmatization, as given by a finite-state morphological analyzer (Minnen et al., 2001);
- PoS tagging, using the OpenNLP[3] implementation of a maximum entropy model (Ratnaparkhi, 1996);
- base phrase chunking, using a SVM-based chunker (Kudoh and Matsumoto, 2000);
- full syntactic parsing, using the Charniak parser (Charniak, 2000).

In order to manage the annotation layers produced by the different preprocessing components, we use MMAX2 (Müller and Strube, 2006). MMAX2 is a highly customizable tool for creating, browsing, visualizing and querying linguistic annotations on multiple levels. As such, it provides a unified platform to access information on different linguistic levels in a consistent and efficient manner. Additionally, it provides a *discourse API* which allows for programmatic access of the documents and their linguistic information on different levels at the same time, e.g. one can straightforwardly retrieve the PoS of the last token of a NP chunk for additional phrase boundary checking. This is a highly desirable feature in the present scenario, where different linguistic phenomena have to be taken into account.

## 3 Document filtering

Manual analysis of the document collections for a number of topics in the training data revealed that usually a reduced number of documents suffices to generate an informative summary, especially given the 250 words limit. Indeed, as it has been observed before (Barzilay, 2003), the most important pieces of information reoccurr over several documents. And although it is seldom the case that there is one single document containing all the relevant information, normally there are 3-5 articles which cover the most important points.

Thus, the next step in our summarization process after the annotation was document filtering. For each document $d_i$, its headline and the first sentence, which are known to be highly informative, were extracted ($h^i, s_1^i \in d_i$). Then a vector space model was constructed using Java-Lucene API[4]. We calculated the score of each document by measuring the cosine similarity between its simplified representation and the topic $T$: $Rank(d_i) = \cos(v(h^i, s_1^i), v(T))$, where $v(x)$ is a vector representing a text fragment $x$. The five documents with the highest score were then picked for the further summarization steps.

## 4 Lexical chains

Barzilay and Elhadad (1999) proposed a method using lexical chains for text summarization. Silber and McCoy (2002) developed a more efficient way to determine lexical chains. Our work is based on Silber and McCoy (2002), but we diverted from their computation of lexical chains in several ways. First, we used the topic information as a filter and computed only chains based on the content words in the topic. Second, we did not use the full document set for each topic, but only the previously selected ones. Both filtering steps considerably reduced the number of chains to be computed and therefore reduced the computational effort. Finally, we scored each sentence that had a chain connection to the topic. This resulted in a ranking of sentences, which were used in the final extraction of sentences for the summary.

### 4.1 Topic/Question Analysis

The title and sentences from the topic were stemmed and filtered using a stopword list (Galley et al., 2003). We expanded the remaining words ($WT$) with all their senses, by gathering the corresponding WordNet synsets (Fellbaum, 1998).

### 4.2 Document Analysis

The previously selected documents (see Section 3 for details) were treated as one large document, and filtered in the same way as the words from the topic (see Sec-

[3] `http://opennlp.sourceforge.net`.

[4] `http://lucene.apache.org`

tion 4.1). The remaining words were collected in the set $WD$. Next, we examined whether the considered noun or verb ($wd_i$) in the document shared a synset with a content bearing word ($wt_j$) from the topic and what kind of relationship ($REL(wt_i, wt_j)$) held between them. The strongest relationship found between any combination of $wd_i$'s and $wt_j$'s synsets is considered. Based on this relationship the word received a score ($score_{wd_i}$), based on the scores suggested by Silber and McCoy (2002).

$$score_{wd_i} = \begin{cases} 0.0 & \exists wt_j \in WT, similar(wd_i, wt_j) \\ 0.5 & \exists wt_j \in WT, hyponym(wd_i, wt_j) \\ 0.5 & \exists wt_j \in WT, hypernym(wd_i, wt_j) \\ 1.0 & \exists wt_j \in WT, antonym(wd_i, wt_j) \\ 1.0 & \exists wt_j \in WT, equal(wd_i, wt_j) \end{cases} \quad (1)$$

### 4.3 Sentence Scoring

The results from the word scores alone were not discriminative enough to filter sentences as many received the same score. We have then changed the scoring by using additional scores. One added score was the number of chains passing through a sentence ($score_{chain}$) and the other score was based on n-grams ($score_{bigram}$, $score_{trigram}$). Each occurrence of a chain increased the score by 1.

$$score_{chain} = l \quad (2)$$

where $l$ is the number of chains passing through the sentence. Each occurrence of a bigram increased the score by 1.

$$score_{bigram} = m \quad (3)$$

where $m$ is the number of bigrams found in the sentence. Each occurrence of a trigram increased the score by 2.

$$score_{trigram} = 2 \times n \quad (4)$$

where $n$ is the number of trigrams found in the sentence. The overall score ($score_{sent}$) was calculated for each sentence and then used to rank sentences for the final extraction.

$$score_{sent} = \sum_{g=1}^{k} score_{word_g} + score_{chain} + score_{bigram} + score_{trigram},$$

where $k$ is the total number of words that were assigned a word score. Using the number of chains passing through a sentence gives higher scores to longer sentences as they are likely to have several chains passing through. But longer sentences also have a higher likelihood to contain more information – especially, if several chains pass through them.

The score based on n-grams favours sentences that also have a high score based on identity. It also emphazises sentences where words from the topic not only occur somewhere in the sentences, but also where words occur in the same order as in the topic sentences.

Chali and Kolla (2004) and Kolla and Chali (2005) also presented work on multi-document summarization using lexical chains for sentence selection. In Chali and Kolla (2004) the scoring mechanism only considers the number of occurrences of words within a sentence and within a segment (segment in their work is comparable to single documents within a collection for one topic). No additional information like n-grams is used. The published results on the DUC 2004/2005 tasks (Dang, 2005) seem to corroborate our own observation that sentence scoring based on simple lexical chain counts (comparable to our $score_{chain}$) is not performant enough.

## 5 Summary generation

The top 30 most relevant sentences selected by the lexical chaining method are the input to another ranking algorithm which reranks presumably relevant sentences according to a different criterion. After that we eliminate redundancy and, finally, order the sentences.

### 5.1 Reranking

This is a modified version of the algorithm described in Nastase and Szpakowicz (2006). There, every candidate sentence and the topic are represented as graphs. Open-class words are vertices and an edge between two words stands for a dependency relation which holds between these words. Graph representations allow for distinguishing between sentences which share some words with the

topic and those which not only share words but also dependencies between them. Thus, a sentence which shares two words with the topic is ranked lower than a sentence which has the same two words in common with the topic if these two words are connected both in the topic and in the latter sentence.

Naturally, two sentences can be similar but have only a few words in common. To identify such cases, WordNet is used in the lexical chain part of our system (Section 4.1). For sentence reranking, we applied a different technique to detect similarities. Dekang Lin's list of similar words[5] was taken and each noun in the topic was expanded with its ten most similar nouns, and each verb with its five most similar verbs. Then we did not just count how many words in the topic are mentioned in a candidate sentence, but also how many of the related words can be found there. The same was done for dependencies. Thus, to compute the new score of a sentence, we assigned weighted scores for every common lemma ($com(w_i)$), i.e. for every word whose lemma $w_i$ is among the extended list of topic words ($w_i \in S, w_i \in T^{ext}$). A higher score was assigned in case where there are common dependencies ($dep(w_i, w_j)$). $S$ and $T$ are a sentence from a document and the topic respectively.

$$score(S) = \sum_{w_i \in S, T^{ext}} com(w_i) + \sum_{w_i, w_j \in S, T^{ext}} dep(w_i, w_j)$$

### 5.2 Redundancy Elimination

The sentence with the highest similarity score is added to the summary first. Before we add any other sentence we check whether we have already reached the 250 words limit and whether this sentence would introduce redundancy. Redundancy elimination is a well-recognized problem in summarization, especially for extractive approaches. A well-known formula which measures the appropriateness of a sentence by considering both its relevance for the topic and the degree of redundancy it would bring into the summary is maximal marginal relevance (Carbonell and Goldstein, 1998, MMR). According to this formula, at each iteration the sentence which maximizes MMR is taken:

$$s = \arg\max_{s_i}[\lambda * Sim(s_i, t) - (1 - \lambda) * \max Sim(s_i, s_j)]$$

where $s_i$ is one of the yet unselected sentences and $s_j$ a sentence from the summary.

The value of $\lambda$ depends on what is more important: relevance or non-redundancy. It has been suggested, that $\lambda$ should increase with the length of summary. For example, $\lambda$ = 0.3 for the first third, $\lambda$ = 0.5 and $\lambda$ = 0.7 for the second and the third thirds of the summary, respectively. Decreasing $\lambda$ was suggested by Murray et al. (2005) and used in DUC 2005 by Hachey et al. (2005).

In our system we used a slightly different formula:

$$s = \arg\max_{s_i}[Sim(s_i, t) + Sim(s_i', t) - 2 * Sim(s_i, s_i')]$$

where $s_i' = \arg\max_{s_i} Sim(s_i, s_j)$, $s_i$ is one of the yet unselected sentences, $s_j$ is a sentence from the summary. The reason for using a different formula was that as soon as $\lambda$ = 0.7, the MMR formula may add a sentence already contained in the summary if its similarity to the topic is high. As it is sometimes the case that there are very similar and even identical sentences in different documents, this property is highly unfortunate. Our formula, on the contrary, is highly unlikely to allow duplicates in the summary.

### 5.3 Sentence Ordering

The rule we applied here is very simple. Sentences extracted from the same document are grouped together and arranged in the order they are found in the original document.

## 6 Results

Our system performed well in manual evaluation, from both a responsiveness and linguistic quality point of view. Figure 1 shows the comparative results of our system with the overall average (over the 30 competing systems) and best scoring system in each of the manual evaluation categories: responsiveness, and linguistic quality (grammaticality, non-redundancy, referential clarity, focus and structure and coherence). The EMLR system performs above the average in each category. It ranked 14th on

[5] http://www.cs.ualberta.ca/~lindek/downloads.htm

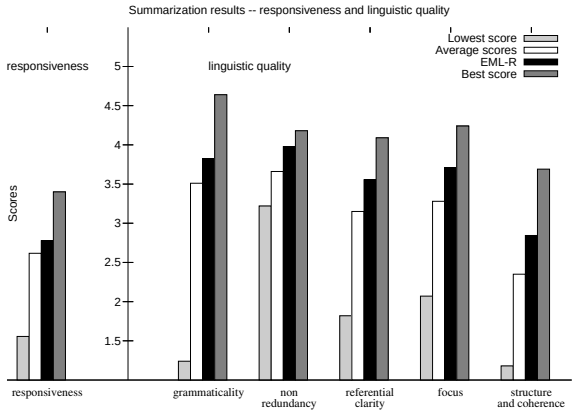


Figure 1: Comparative results for manual evaluation for EMLR, overall average and best system

manually assessed responsiveness, and 4th from a linguistic quality point of view.

The EMLR system's performance is also good in terms of the basic elements (BE) automatic evaluation, where it ranked 18th, but rather poor for ROUGE-2 and ROUGE-SU4 scores. The fact that during prototyping we have relied on manual evaluation of the results, rather than automatic scoring, partly explains the difference in ranking between manual and automatic scoring.

The document filtering step affects positively non-redundancy and coherence of the final summary: From the 30 participants, EMLR is ranked fourth for both of these parameters. Having a smaller set of documents, and therefore a smaller number of sentences to choose from, decreases the number of potential redundancies. A simple heuristics which brings sentences from the same document together and preserves their original order is more useful when applied to a smaller set of documents than when applied to sentences extracted from more than 20 documents. The high non-redundancy score is also attributed to the modified MMR formula which harshly penalizes redundancy.

Reducing the number of documents also reduces significantly the computation time. We ran experiments not only on the selected five documents, but also on the whole collection of documents, belonging to one topic/question. The computational effort to calculate $score_{word}$ was considerably higher, than using the selected documents, due to the higher amount of words to analyze. The results did not justify the higher computation load.

Another experiment concerned word analysis. The topic words were expanded using similar words from Dekang Lin's thesaurus, and all were further expanded to all their WordNet synsets. This resulted a larger search space, which increased the computational effort needed to compute the scores. Again, the results did not justify this increase in computational effort.

We added a fall-back method based on the results from these experiments. In those cases where we found too few chains based on the original method, we computed lexical chains based on the same method, but on the whole document set. Additionally, we did not filter the topic words for stop words, but rather used all occuring words. And finally, we used the similar words for finding relationships between words as well. The reason for adding this method was that sometimes too few sentences were selected for the summary and through this fall-back method we were able to provide the extraction method with more material to choose from. In such cases the computational effort for using similar words as well as the full documents set for the topic justified the results.

## 7 Conclusions

We have presented EMLR's first participation at the DUC competition on the topic-driven multi-document summarization task. The system ranked 14th out of 30 participating teams in manual evaluation, and had particularly good results for linguistic quality, ranking 4th overall. The document filtering step has reduced the number of redundant sentences with respect to the topic, which has contributed to the system's high rank for non-redundancy.

We plan to develop a more accurate method for document filtering, with a flexible threshhold and not a fixed number of documents as in the current system. For increasing responsiveness we aim for an abstractive summary in our system's next version, by aggregating small pieces of relevant information from the document extracted sentences.

**Acknowledgements** This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany.

## References

Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic*

*Text Summarization*, pages 111–121. Cambridge/MA, London/England: MIT Press.

Regina Barzilay. 2003. *Information Fusion for Mutlidocument Summarization: Paraphrasing and Generation*. Ph.D. thesis, Columbia University.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Alistair Moffat and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, pages 335–336.

Yllias Chali and Maheedhar Kolla. 2004. Summarization techniques at DUC 2004. In *Proceedings of the 2004 Document Understanding Workshop* Boston, USA, May 6-7, 2004.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, Seattle, Wash., 29 April – 3 May, 2000, pages 132–139.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the 2005 Document Understanding Workshop*, Vancouver, B.C., Canada, October 9-10, 2005.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 7–12 July 2003, pages 562–569.

Ben Hachey, Gabriel Murray, and David Reitter. 2005. The Embra system at DUC 2005: Query-oriented multi-document summarization with a very large latent semantic space. In *Proceedings of the 2005 Document Understanding Workshop*, Vancouver, B.C., Canada, October 9-10, 2005.

Maheedhar Kolla and Yllias Chali. 2005. Experiments in DUC 2005. In *Proceedings of the 2005 Document Understanding Workshop*, Vancouver, B.C., Canada, October 9-10, 2005.

Taku Kudoh and Yuji Matsumoto. 2000. Use of Support Vector Machines for chunk identification. In *Proceedings of the 4th Conference on Computational Natural Language Learning*, Lisbon, Portugal, 13–14 September 2000, pages 142–144.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In S. Braun, K. Kohn, and J. Mukherjee, editors, *Corpus Technology and Language Pedagogy*, pages 197–214. Peter Lang, Frankfurt, Germany.

Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH '05)*, Lisbon, Portugal, 4–8 September 2005.

Vivi Nastase and Stan Szpakowicz. 2006. A study of two graph algorithms in topic-driven summarization. In *HLT-NAACL 06, Proceedings of TextGraphs: Graph-based algorithms for Natural Language Processing*, pages 29–32, New York City, 9 June 2006.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing*, Philadelphia, Penn., 17–18 May 1996, pages 133–142.

H.Gregory Silber and Kathleen F. McCoy. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.