Generating Update Summaries with Spreading Activation

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

Abstract

For the update summaries task of the Text Analysis Conference 2008 we have implemented a novel summarization technique based on query expansion with encyclopedic knowledge and activation spreading in a large document graph. We have also experimented with sentence compression for building the summaries. The results are average – ranked 27 out of 58 for responsiveness in manual evaluation – but we find the approach promising.

1 Introduction

EML Research has participated in the update task of the Text Analysis Conference (TAC) 2008, for topic-driven multi-document update summarization. The task consists of two stages: produce a 100-word summary from a set of documents that address the topic associated with this collection; produce a second summary based on a second set of documents associated with the same topic, such that the new summary presents novel information compared with the first summary.

Our summarization algorithm follows four steps:

1. expand the query using encyclopedic knowledge from Wikipedia;
2. spread an activation in a large graph that covers all documents in a collection to be summarized – nodes are terms/NEs in the documents, edges correspond to grammatical dependency relations;
3. rank the nodes of the graph with a PageRank algorithm (Brin & Page, 1998), to select from the most highly activated nodes the ones that are also important in the documents;
4. rank sentences based on their relatedness to the topic and activation, and form the summary from the highest ranking sentences that have minimal overlap. Apart from a purely extractive approach, this year we have forayed into abstractive summarization, by compressing sentences.

The motivation for including encyclopedic knowledge for query expansion is that in understanding a text – the short query or the associated documents – we rely on more than lexical semantic knowledge. We expand the terms in the query using hyperlinks in the first paragraph of their corresponding Wikipedia articles. The next step is to connect these terms with the documents to be summarized, and expand the query further within the document. This expansion is important, as it incorporates document specific information in the expanded query, allowing the system to adjust to the information from the documents to be summarized. For this we use activation spreading in a large graph that represents the terms in the documents and the grammatical relations between them. We can control how far the influence of the query terms and their expansions should be felt in this graph through a signal decay parameter. For the update task this may be particularly appealing, since the information to be summarized in later stages may not be directly related to the topic. To clarify this point, we present in Figure 1 a topic from the training data for the update pilot task from the Document Understanding Conference (DUC) 2007.

This topic has associated 3 disjunct (temporally ordered) sets of documents – A, B, C – each to be used to produce a 100-word summary with novel information (compared to the previous ones). According to the human summarizers, the summary of set A was supposed to give information about the terrorist attacks; the second (set B) about measures taken by the government about prosecuting those guilty for the attacks, and about international reactions after the attacks; the third (set C) about

<topic>
<num> D0746C < /num>
<title> Terrorist attacks in Luxor, Egypt < /title>
<narr>
What attacks have occurred against tourists in Luxor, Egypt? Provide details about the attacks and the subsequent ramifications thereof. < /narr>
<docs>
...
< /docs>
< /topic>

Figure 1: Sample topic from training data for the update pilot task for DUC 2007

actions taken by the government to boost tourism in the area again after the attacks. This example shows that only the first summary is concerned with the attacks. The other two are ramifications of these events, and the topic does not provide many clues about what they should contain. In this case we should look for finding terms that are more loosely connected to the topic.

2 Related Work

The system we present in this paper is an adaptation of the system described in (Nastase, 2008) to the update summarization task. In constructing this system we build upon previous work on query expansion and graph-based summarization models.

Barzilay & Elhadad (1999) use WordNet to model a text's content relative to a topic based on lexical chains. The sentences intersected by the most and strongest chains are chosen for the extractive summary. Alternative sources for query expansion and document processing have also been explored. Amini & Usunier (2007) use the documents to be summarized themselves to cluster terms, and thus expanding the query "internally". More advanced methods for query expansion use "topic signatures" – words and grammatically related pairs of words that model the query and even the expected answer from sets of documents marked as relevant or not (Lin & Hovy, 2000; Harabagiu, 2004).

Graph-based methods for text summarization work usually at the level of sentences (Erkan & Radev, 2004; Mihalcea & Tarau, 2004). Edge weights between sentences represent a similarity measure, and a PageRank algorithm is used to determine the sentences that are the most salient from a collection of documents and closest to a given topic. At the word level, Leskovec et al. (2004) build a document graph using subject-verb-object triples, semantic normalization and coreference resolution. They use several methods (node degree, PageRank, Hubs, etc.) to compute statistics for the nodes in the network, and use these as attribute values in a machine learning algorithm, where the attribute that is learned is whether the node should appear in the final summary or not. Annotations for training come from human produced summaries. Mohamed & Rajasekaran (2006) incrementally build a graph for a document collection by combining graph-representations of sentences. Links between entities in a sentence can be *isa* (within an NP) or *related_to* (between different phrases in a sentence). Nodes and relations are weighted according to their connectivity, and sentence selection for the final summary is based on the most inter-connected nodes. Ye & Chua (2006) build an extractive summary based on a concept lattice, which captures in a hierarchical structure co-occurrences of concepts among sentences. Nodes higher in this structure correspond to frequently co-occurring terms, and are assumed to be more representative with respect to the document topic.

Mani & Bloedorn (1999) build a "chronological" graph, in which sentence order is respected and each occurrence of a concept is a separate node. Edges between nodes cover several types of relations: adjacency (ADJ); identity – instance of the same word (SAME); other semantic links, in particular synonymy and hypernymy; PHRASE links connect components of a phrase; NAME indicate named entities; COREF link coreferential name instances. Among other things, they identify regions of the text salient to a user's query, based on spreading activation starting from query words in this document graph. Spreading activation was introduced in the 60s and 70s to model psychological processes of memory activation in humans (Quillian, 1967; Collins & Loftus, 1975).

As described in (Nastase, 2008), we use Wikipedia as a source of knowledge for related concepts – the texts of hyperlinks in an article describing a concept are taken as its related concepts. The query is further expanded by using spreading activation to move away from the topic in a large graph that covers all documents for a given topic. From the nodes thus reached we select using a PageRank algorithm the ones that are most important in the documents. We study the impact of a decay parameter which controls how far to move from the topic, and the number of highest ranked nodes to be added to the expanded topic. The summary is built based on word associations in the documents' graph.

3 Query Expansion with Encyclopedic Knowledge

In TAC/DUC topic-driven multi-document summarization, the topic has a title, an ID that links it to a set of

Matthew Shepard

Matthew Wayne Shepard (December 1, 1976 October 12, 1998) was a gay American student at the University of Wyoming who was murdered near Laramie on the night of October 6October 7, 1998. Shepard died at Poudre Valley Hospital in Fort Collins, Colorado , on October 12, 1998, from severe head injuries. His murder brought national as well as international attention to the issue of hate crime legislation at the state and federal levels.

```
'''Matthew Wayne Shepard''' (December 1,
1976 October 12, 1998) was a gay [[United
States|American]] student at the [[University
of Wyoming]] who was murdered near [[Laramie,
Wyoming|Laramie]] on the night of October 6October
7, 1998.  Shepard died at [[Poudre Valley Hospital]]
in [[Fort Collins, Colorado]], on October 12, 1998,
from severe head injuries.  His murder brought
national as well as international attention to the
issue of [[hate crime]] legislation at the state and
federal levels.
```

Extracted related concepts for *Matthew Shepard*:

American, University of Wyoming, Laramie, Wyoming, hate crime, Fort Collins

Figure 2: First paragraph for article *Matthew Shepard* in the English Wikipedia, and the extracted related concepts.

documents, and one or more sentences and/or questions, as illustrated in Figure 1. Topic processing is done in several steps:

1. Preprocessing: Produce the dependency pair representation of the topics using the Stanford Parser (Klein & Manning, 2003)[1]. Pairs that have closed-class words are filtered out, and the remaining words are lemmatized[2]. We extract named entities (NEs), as the parser splits them as any other phrase. In the dependency pairs we replace an NE's fragments with the complete NE.

2. Query expansion with Wikipedia: Extract all open-class words and NEs from the topic, and expand them using Wikipedia articles whose titles refer to these words or phrases.

For each Wikipedia article we extract as related concepts the texts of the hyperlinks in the first paragraph (see Figure 2). The reason for not including links from the entire article body is that apart from the first paragraph, which is more focused, often hyperlinks are included whenever the underlying concept appears in Wikipedia, without it being particularly relevant to the article.

To expand a word (or NE) w from the query, we search for an article having w as the title, or part of the title.

1. If one exact match is found (e.g. Matthew Shepard), extract the related concepts for this article.
2. If several exact or partial matches are found, use the larger context of the query to narrow down to the intended meaning. For example, *Turkey* – referring to the country – appears in several topics in the DUC 2007 data. There are multiple entries for "Turkey" in Wikipedia – for the country, the bird, cities with this name in the U.S. among others. We use a Lesk-like measure, and compute the overlap between the topic query and the set of hyperlinks in the first paragraph (Lesk, 1986). We choose the expansion for the entry with the highest overlap. If the query context does not help in disambiguation, we use the expansions for all partial matches that tie for the highest overlap.

[1] `http://nlp.stanford.edu/software/lex-parser.shtml`

[2] Using XTAG morphological database `ftp://ftp.cis.upenn.edu/pub/xtag/morph-1.5/morph-1.5.tar.gz`.

4 Topic Expansion with Spreading Activation and PageRank

Concepts related to the ones in the topic provide a good handle on the documents to summarize – they indicate parts of the document that should be included in the summary. It is however obvious that the summary should contain more than that, and this information comes from the documents to be summarized. Amini & Usunier (2007) have shown that expanding the query within the set of documents leads to good results. Following this idea, to find more relevant concepts we look for words/NEs which are related to the topic, and at the same time important in the collection of documents for the given topic. The methods described in this section are applied on a large graph that covers the entire document collection for one topic. The documents are processed in a similar way to the query – parsed with the Stanford Parser (output in dependency relation format), lemmatized using XTAG's morphological data file. The graph consists of nodes corresponding to lemmatized words and NEs in the documents, and edges correspoding to grammatical dependency relations.

4.1 Spreading Activation

To find words/NEs related to the topic we spread an activation signal starting from the topic words and their expansions which are given a node weight of 1 (in a manner similar to Mani & Bloedorn (1999), and using an algorithm inspired by Anderson (1983)). As we traverse the graph starting from these nodes, the signal is propagated by assigning a weight to each edge and each node traversed based on the signal strength. The signal strength diminishes with the distance from the node of origin depending on a signal decay parameter, according to the formula:

$$
\begin{aligned}
w_n(N_0) &= 1 \\
s_t &= (1 - decay) * \frac{w_n(N_t)}{Out(N_t)} \\
w_n(N_{t+1}) &= s_t \\
w_e(N_t, N_{t+1})_{t+1} &= w_e(N_t, N_{t+1})_t + s_t
\end{aligned}
$$

where N_t is the current node; N_{t+1} is the node we are moving towards; $w_n(N_t)$ is the weight of node N_t; s_t is the signal strength at step t; $Out(N_t)$ is the number of outgoing edges from node N_t; $w_e(N_t, N_{t+1})_t$ is the weight of the edge between N_t and N_{t+1} at time t (i.e., before actually traversing the edge and spreading the activation from N_t); $w_e(N_t, N_{t+1})_{t+1}$ is the weight of the edge after spreading activation. The weight of the edges is cumulative, to gather strength from all signals that pass through the edge. Activation is spread sequentially from each node in the (expanded) topic.

4.2 PageRank

The previous step has assigned weights to edges in the graph, such that higher weights are closer to the topic and/or topic expanded words. After this initialization of the graph, we run the PageRank algorithm to determine more important nodes. By running this algorithm after initializing the graph edge weights, from the nodes that are closer to topic and topic expanded words we boost those that are more important in the documents.

The starting point of the PageRank algorithm is the graph with weighted edges obtained in the previous step. Analysis of the documents graph for several topics has revealed that there is a large highly interconnected structure, and many disconnected small (2-3 nodes) fragments. PageRank will run on this dense core structure. The PageRank algorithm is guaranteed to converge if the graph is aperiodic and irreducible – based on the Ergodic theorem for Markov chains (Grimmett & Stirzaker, 1989). Aperiodicity implies that the greatest common divisor of the graph's cycles is 1 – this condition is met.

		D0711C	D0740I
Topic		Summarize Microsoft's antitrust problems, including its alleged illegal behaviour and antitrust proceedings against the company.	Report on the planning, attempts and first successful balloon circumnavigation of the earth by Bertrand Piccard and his crew.
expanded		proceeding, alleged, illegal, summarize, microsoft, include, behaviour, object, action, relation, antitrust, problem, company	first, circumnavigation, Earth, round, successful, crew, planning, plan, attempt, flight, lift, air, Sun, Bertrand Piccard, balloonist, Swiss, Switzerland, balloon, report, picture, air, helium
top ranked	A	object, trial, effort, fee, ibm, spend, take, practice, call, accuse, violation, witness, deny, marketing, price	Andy Elson, cold, circumnavigate, round, spend, calm, person, pilot, Wimver Straeten, Swiss, Switzerland, space, Chateau D'Oex, try, announce, spectator
	B	effort, trial, document, equivalent case, take, monopolist, justice department, lawyer, government, violation, engage, harm, soldier, avoid, prove, product, suit	ballonist, delay, fly, take, dead, travel, foot, set, capsule, make, frigid, bad-smelling, thin, venture, circuit, become, complete
	C	breakup, remedy, previous, proposal, demonstrate, order, modify, act, conduct, accountable, amend, separate, restriction, suit, decide	hope, need, two, helium, use, sealed, envelope, burner, huge, bags, force, heat, nylon, sun, expand, used, complete

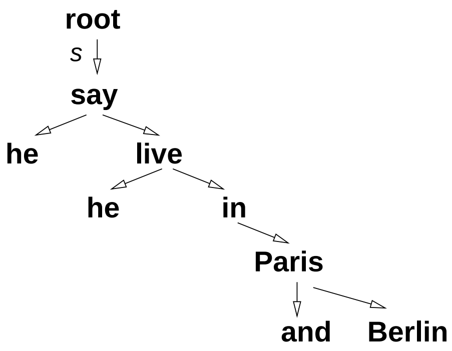
Table 1: Top ranked nodes after expanding the topic with spreading activation and PageRank

Irreducibility of the graph means that it has no leaves, and there are no two nodes with the same set of neighbours. The remedy in such cases is to connect each leaf to all other nodes in the graph, and conflate nodes with the same set of neighbours.

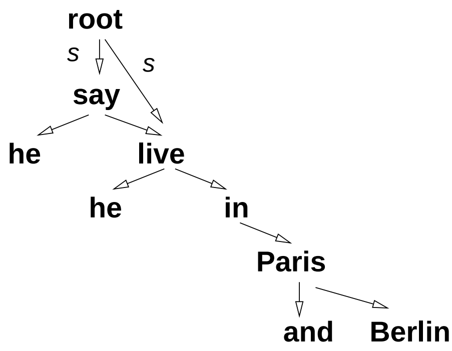
Once the graph topology meets the PageRank convergence conditions, we run the algorithm. The original formula for computing the rank of a node at each iteration step is:

$$PR(n_i) = \frac{1-d}{N} + d \sum_{n_j \in Adj_{n_i}} \frac{PR(n_j)}{Out(n_j)}$$

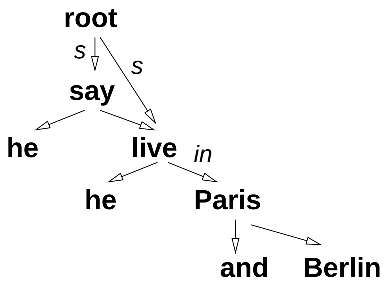
where n_i is a node, d is the damping factor (we follow the standard practice and use $d = 0.85$), N is the number of nodes in the graph, $PR(n_i)$ is the rank of node n_i, Adj_{n_i} is the set of nodes adjacent to n_i, and $Out(n_j)$ is the number of outgoing edges from n_j (our graph is non-directed, so this number is the total number of edges



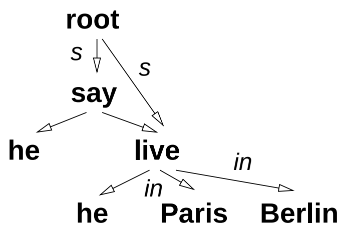
(a) The root is inserted



(b) Edges to the verbs are added



(c) Prepositions are removed



(d) Conjuncts are split

Figure 3: The dependency structure of *He said that he lived in Paris and Berlin* after the transformations

with one end in n_j). We adjust this formula to reflect the weights of the edges, and the version used is the following:

$$PR(n_i) = \frac{1-d}{N} + d \sum_{n_j \in Adj_{n_i}} PR(n_j) w_{out}(n_j);$$

$$w_{out}(n_j) = \sum_{n_k \in Adj_{n_j}} w_e(n_k, n_j)$$

In Table 1 we show examples of top ranked nodes for several topics, extracted with this algorithm. The words in italics are keywords/phrases from the topic query, and the top ranked nodes are listed in decreasing order of their rank.

5 Sentence Compression

A well-known drawback of extractive summarization is that an informative sentence may contain irrelevant information which one would like to avoid in the summary. Given how short a summary must be, it is desirable to compress selected sentences. E.g., relative clauses or appositions can often be removed without affecting the gist of the sentence.

There are two possible ways of integrating sentence compression in a summarization system. One can either compress all the sentences and then extract the most important ones, or first rank all of them and then compress the top n. Here, we explore the latter possibility in order not to loose any information which could reveal relations between sentences. Thus, we compress sentences pre-selected for the summary to check if this improves the resposiveness of the summary with a possible minor drop in its linguistic quality.

5.1 Compression Algorithm

Several compression methods have been developed for English (Knight & Marcu, 2002; Turner & Charniak, 2005; Clarke & Lapata, 2008, inter alia). We apply our unsupervized method (**?**) which has shown state-of-the-art results when evaluated automatically on a compression corpus[3]. In a nutshell, the compression algorithm proceeds as follows:

1. The sentence is parsed with the Stanford parser which showed best results in our experiments on the compression corpus. The parser has an option to convert a phrase structure tree into a dependency tree which we use.

2. The dependency tree is transformed so that the relations between the open-class words become more explicit. E.g., a root node is inserted and an edge from the root to every inflected verb is added; a chain of coordinated conjuncts is split and each of them is attached to the head word (see Figure 3).

3. The transformed tree, which in most cases is a directed graph, is compressed. Edges which are not syntactically important and do not point to informative words get removed. A set of constraints guarantees that the resulting graph is a tree. Integer Linear Programming[4] is used to find a globally optimal solution efficiently.

4. The resulting tree is linearized by placing the words in the original order, i.e. in the order from the uncompressed sentence.

6 Summary generation

Sentences are ranked based on their overlap with the topic and their content. After ranking we choose from the best sentences those with a minimal overlap, and form the 100 word summary.

6.1 Ranking

This is a modified version of the algorithm described in Nastase & 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.

We count not only how many words in the topic are mentioned in a candidate sentence, but also how many of the expanded query words can be found there. The same was done for dependencies. Thus, to compute the score of a sentence, we combine the weighted scores for lexical overlap with the topic (W_S), content of Wikipedia-expanded topic words ($W_S^{W_{exp}}$), content of top ranked nodes (W_S^{top}), dependency overlap with the topic ($Dep_{S,T}$) and dependency overlap with other sentences ($Dep_{S,*}$):

$$W_S = \{w_i | w_i \in S, w_i \in T\}$$
$$W_S^{W_{exp}} = \{w_j | w_j \in S, w_j \in T^{exp}\}$$
$$W_S^{top} = \{w_k | w_k \in S, w_k \in Top\}$$
$$Dep_{S,T} = \{(w_x, w_y) | (w_x, w_y) \in S, (w_x, w_y) \in T\}$$
$$Dep_{S,*} = \bigcup_{i \in \{1,..,n\}} Dep_{S,S_i}$$

$$\begin{aligned} score(S) &= |W_S| * w_{word} \\ &+ |W_S^{W_{exp}}| * w_{expWord} \\ &+ |W_S^{top}| * w_{topWord} \\ &+ |Dep_{S,T}| * w_{depRelation} \\ &+ |Dep_{S,*}| * w_{subgraphEdge} \end{aligned}$$

6.2 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 100 words limit and whether this sentence would introduce redundancy. We use a threshold parameter to control how much extra information to allow. Sentence overlap is based on lexical overlap (after stop-word elimination), normalized by the length of the sentence.

7 Results and Discussion

7.1 System development

Our system has several parameters that can influence the performance. System development for parameter tuning was done on the DUC 2007 update test data. The weights of the sentence scoring formula were set empirically to the following values: $w_{word} = 5$, $w_{expWord} = 1.1$, $w_{topWord} = 1.1$, $w_{subgraphEdge} = 1$, $w_{depRelation} = 2$. The redundancy threshold value is 0.5.

The most interesting of the system's parameters are the signal decay parameter for the activation spreading method and the number of top ranked nodes we choose after the PageRank algorithm to add to the query expansion. The number of top ranked nodes chosen was 20. The signal decay parameter is adjusted for each summarization stage (corresponding to the three document collections per topic). To find the appropriate values for these parameters we perform multiple runs on the DUC 2007 update data. We obtained excellent performance

[3]The corpus is available from `http://homepages.inf.ed.ac.uk/s0460084/data`. It consists of news stories from the British National Corpus and the American News Text Corpus.

[4]We use `lp_solve` in our implementation: `http://sourceforge.net/projects/lpsolve`.

during the development phase, our tuned system ranking 2nd in ROUGE-2 (0.10166), ROUGE-SU4 (0.14223) and BE (0.06391) automatic evaluations.

7.2 TAC 2008 results

We have submitted three runs for TAC 2008:

ID 10 This is an extractive summarization method, which relies on Wikipedia expansion of topic words, activation spreading with decay 0.9999 for document set A, and 0.999 for document set B, and Page Rank for detecting top ranked nodes connected to the query in the document collection.

ID 40 This method is our attempt for abstractive summarization. The sentences are scored similarly and using the same settings as run 10. From the ranked sentences, the top ones are compressed and put together to form the 100 word summaries.

ID 61 This method is similar to run 10, with a difference in signal decay: for document set B, the signal decay was 0.99, to allow us to explore the effect of allowing the signal to travel further in the document graph.

7.3 Activation Spreading

Runs 10 and 40 were also manually evaluated, run 61 was only automatically evaluated. In all automatic evaluations, run 61 was better than the others. Figure 4 shows the comparison between runs 10 and 61, to allows us to see the difference in performance due to the signal decay parameter. Results are ordered increasingly based on the BE scores for run 61.

We have looked closer at the outliers to understand the variation in performance. The first peek, where run 10 (decay value 0.999) performs better is or topic D0842G: *Natural Gas Pipeline: Follow the progress of pipelines being built to move natural gas from Asia to Europe. Include any problems encountered and implications resulting from the pipeline construction.* The summaries produced from document set B for the two decay value vary in only one sentence:

decay = 0.999

> Croatia and Hungary are weighing construction of a gas pipeline from the Adriatic Sea to mainland Europe in order to decrease reliance on Russian gas, the prime ministers of the two central European countries said Thursday.

decay = 0.99

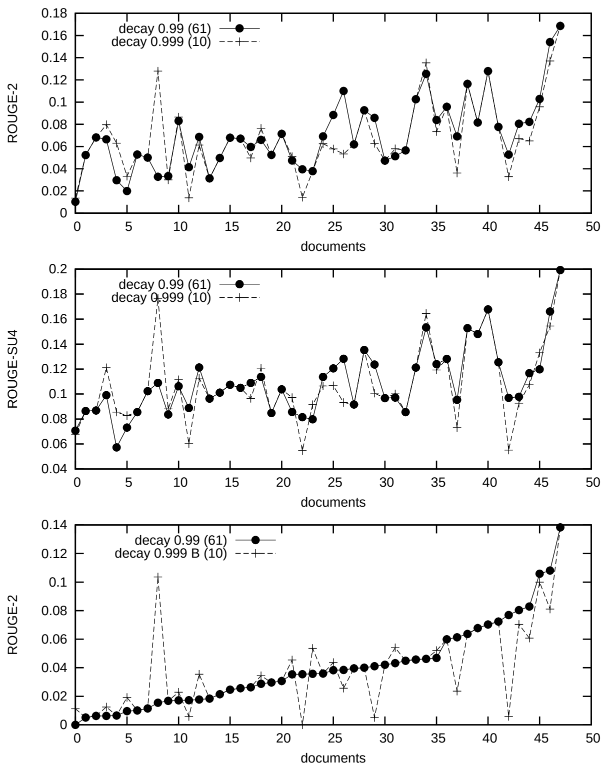


Figure 4: Impact of signal decay in spreading activation on summarization performance - comparison of sets B for run 10 and 61.

> Russia moves natural gas shipments through a grid of Soviet-era pipelines so complex it is virtually impossible to guarantee the gas it pumps into Ukraine comes out the other side and reaches European customers.

The sentence chosen by the system with lower signal decay contains information about older gas pipelines, not the ones being built now, as the topic requests. The sentence's score is boosted by the following high-ranking nodes: *shipment*, *Russia*, *Ukraine*, *other*, which are not among the top chosen nodes for the higher decay value.

The other outliers we looked at correspond to the point where a lower decay leads to better performance:

Topic D0819D: *Paris Riots: Describe the violent riots occurring in the Paris suburbs beginning October 27, 2005. Include details of the causes and casualties of the riots and government and police responses.*

The difference between the summaries is two sentences:

decay = 0.999:

> For the first time Saturday afternoon, clashes between police and rioters erupted in the heart

> of a major French city, Lyon, where officers used teargas to disperse stone-throwing youths in the historic Place Bellecour in the city center.
> Far-right leader Jean-Marie Le Pen in an interview on the private radio station RTL1 on Sunday blamed the rioting on "uncontrolled immigration from the Third World" and, while endorsing the use of curfews, he described the government response as insufficient.

decay = 0.99

> The assembly bans in Paris and Lyon were imposed under emergency legislation activated by the government of President Jacques Chirac on Tuesday in response to the worst outbreak of urban violence in France since the student uprising of May 1968.
> The French cabinet Monday approved a bill to extend emergency police powers for three months in response to the violence that has been raging in poor city suburbs.

In this case the lower decay allows the system to rank high the following nodes: *violence, bans, suburb, police, power, bill*.

The results support our hypothesis that allowing the system to choose words/concepts further from the topic for successive summarization stages leads to a better sentence selection. The sentence scoring favours longer sentences, which are more likely to contain also irrelevant information. We will look into normalizing the sentence score such that we can choose several shorter sentences that are more focused on the required topic.

7.4 Sentence Compression

The evaluation results are presented in Table 2. These include the ROUGE scores as well as the scores of manual evaluation.

ROUGE-2	ROUGE-SU4	LING.QUALITY	RESP.
0.067	0.108	1.958	1.990

Table 2: The results for the compressed summaries

Overall, the system performed poorly and was ranked low in the automatic as well as in the manual evaluations. Clearly, poor linguistic quality of the compressions affected the responsiveness score which is lower than the responsiveness of the uncompressed summaries (the sentence ranking method is the same). Having analyzed a number of compressed sentences, we identified three main sources of ungrammaticality:

- Parser errors affect the quality of compressions significantly since the method exclusively relies on the dependency representation.

- Some modifiers removed during compression are crucial for correct sentence interpretation. For example, *The ban supports an anti-sweets campaign by the Paediatrics Society of Thailand to reduce the numbers of children hooked on sugar*[5] got compressed to *The ban supports an anti-sweets campaign by the Paediatrics Society of Thailand to reduce the numbers of children*.

- The transformation rules we applied led to wrong assumptions and need to be adjusted for future experiments.

8 Conclusions

We have presented EMLR's participation in the update task of TAC 2008. Our system ranked 27th out of 58 systems in manually assessed responsiveness.

We have experimented with a novel summarization approach, that expands the query terms with related concepts using hyperlinks in Wikipedia articles, and salient nodes from the documents to be summarized. Such nodes are found by sending an activation signal from the topic and topic expanded terms, and then choosing from the nodes activated the ones that are most important in the documents. The signal decay parameter allows us to control how far the influence of the topic words should reach. We have found that controlling this parameter, we can produce better results for the second stage (set B). Analysis of the results per topic has revealed that not all topic should be treated the same way, and that we could improve the performance by adjusting the decay parameter dynamically, based on characteristics of the topic. We plan to investigate this in future work.

Another novel aspect of our system was sentence compression based on grammatical dependency relations. Despite the discouraging results, we would like to continue experiments with sentence compression for summarization. First, we are going to fix the errors due to wrong tree transformations. Then we plan to more carefully analyze cases where a modifier is necessary for a correct interpretation and modify the scoring function accordingly. We would also like to cluster related sentences and experiment with sentence fusion in the future.

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

[5] From the document D0825-B.

References

Amini, M. R. & N. Usunier (2007). A contextual query expansion approach by term clustering for robust text summarization. In *Proc. of DUC-07*.

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behaviour*, 22:261–295.

Barzilay, R. & M. Elhadad (1999). Using lexical chains for text summarization. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization*, pp. 111–121. Cambridge, Mass.: MIT Press.

Brin, S. & L. Page (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:1–7.

Clarke, J. & M. Lapata (2008). Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Collins, A. M. & E. F. Loftus (1975). A spreading-activation theory of semantic processing. *Psychological Review*, (82):407–428.

Edmundson, H. (1969). New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Erkan, G. & D. R. Radev (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fellbaum, C. (Ed.) (1998). *WordNet: An Electronic Lexical Database.* Cambridge, Mass.: MIT Press.

Filippova, K. & M. Strube (2008). Dependency tree based sentence compression. In *Proceedings of the 5th International Conference on Natural Language Generation*, Salt Fork, Ohio, 12–14 June 2008, pp. 25–32.

Gotti, F., G. Lapalme, L. Nerima & E. Wehrli (2007). GOFAIsum: A symbolic summarizer for DUC. In *Proc. of DUC-07*.

Grimmett, G. & D. Stirzaker (1989). *Probability and Random Processes.* Oxford University Press.

Harabagiu, S. M. (2004). Incremental topic representations. In *Proc. of COLING-04*, pp. 583–589.

Hickl, A., K. Roberts & F. L. C. C. Lacatusu (2007). LCC's GISTexter at DUC 2007: Machine reading for update summarization. In *Proc. of DUC-07*.

Klein, D. & C. D. Manning (2003). Accurate unlexicalized parsing. In *Proc. of ACL-03*, pp. 423–430.

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

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual Conference on Systems Documentation*, Toronto, Ontario, Canada, pp. 24–26.

Leskovec, J., M. Grobelnik & N. Milic-Frayling (2004). Learning sub-structures of document semantic graphs for document summarization. In *Proc. of LinkKDD-04*.

Lin, C.-Y. & E. Hovy (2000). The automated acquisition of topic signatures for automatic summarization. In *Proc. of COLING-00*, pp. 495–501.

Mani, I. & E. Bloedorn (1999). Summarizing similarities and differences among related documents. *Information Retrieval*, 1(1):35–67.

Mihalcea, R. & P. Tarau (2004). TextRank: Bringing order into texts. In *Proc. EMNLP-04*, pp. 404–411.

Mohamed, A. A. & S. Rajasekaran (2006). Query-based summarization based on document graphs. In *Proc. of DUC-06*.

Nastase, V. (2008). Topic-driven multi-document summarization with encyclopedic knowledge and activation spreading. In *Proc. EMNLP-08*.

Nastase, V. & S. Szpakowicz (2006). A study of two graph algorithms in topic-driven summarization. In *Proc. of TextGraphs at HLT-NAACL-06*, pp. 29–32.

Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioural Science*, 12(5):410–430.

Rath, G., A. Resnick & T. Savage (1961). The formation of abstracts by the selection of sentences. *American Documentation*, 12(2):139–143.

Turner, J. & E. Charniak (2005). Supervised and unsupervised learning for sentence compression. In *Proc. of ACL-05*, pp. 290–297.

Ye, S. & T.-S. Chua (2006). NUS at DUC 2006: Document concept lattice for summarization. In *Proc. of DUC-06*.