A Study of Two Graph Algorithms in Topic-driven Summarization

Vivi Nastase[1] and **Stan Szpakowicz**[1,2]
[1] School of Information Technology and Engineering,
University of Ottawa, Ottawa, Canada
[2] Institute of Computer Science,
Polish Academy of Sciences, Warsaw, Poland
{vnastase, szpak}@site.uottawa.ca

Abstract

We study how two graph algorithms apply to topic-driven summarization in the scope of Document Understanding Conferences. The DUC 2005 and 2006 tasks were to summarize into 250 words a collection of documents on a topic consisting of a few statements or questions. Our algorithms select sentences for extraction. We measure their performance on the DUC 2005 test data, using the Summary Content Units made available after the challenge. One algorithm matches a graph representing the entire topic against each sentence in the collection. The other algorithm checks, for pairs of open-class words in the topic, whether they can be connected in the syntactic graph of each sentence. Matching performs better than connecting words, but a combination of both methods works best. They also both favour longer sentences, which makes summaries more fluent.

1 Introduction

The DUC 2005 and 2006 summarization challenges were motivated by the desire to make summarization relevant to real users. The task was focussed by specifying an information need as a *topic*: one or a few statements or questions (Dang, 2005). Systems usually employ such data as a source of key words or phrases which then help rank document sentences by relevance to the topic.

We explore other information that can be extracted from a topic description. In particular, we look at connections between open-class words. A dependency parser, MiniPar (Lin, 1998), builds a dependency relation graph for each sentence. We apply such graphs in two ways. We match a graph that covers the entire topic description against the graph for each sentence in the collection. We also extract all pairs of open-class words from the topic description, and check whether they are connected in the sentence graphs. Both methods let us rank sentences; the top-ranking ones go into a summary of at most 250 words. We evaluate the summaries with the summary content units (SCU) data made available after DUC 2005 (Nenkova and Passonneau, 2004; Copeck and Szpakowicz, 2005). The experiments show that using more information than just keywords leads to summaries with more SCUs (total and unique) and higher SCU weight.

We present related work in section 2, and the data and the representation we work with in section 3. Section 4 shows the algorithms in more detail. We describe the experiments and their results in section 5, and draw a few conclusions in section 6.

2 Related work

Erkan and Radev (2004), Mihalcea (2004), Mihalcea and Tarau (2004) introduced graph methods for summarization, word sense disambiguation and other NLP applications.

The summarization graph-based systems implement a form of sentence ranking, based on the idea of prestige or centrality in social networks. In this case the network consists of sentences, and significantly similar sentences are interconnected. Various measures (such as node degree) help find the most *central* sentences, or to score each sentence.

In topic-driven summarization, one or more sentences or questions describe an information need which the summaries must address. Previous systems extracted key words or phrases from topics and used them to focus the summary (Fisher et al., 2005).

Our experiments show that there is more to topics than key words or phrases. We will experiment with using grammatical dependency relations for the task of extractive summarization.

In previous research, graph-matching using grammatical relations was used to detect textual entailment (Haghighi et al., 2005).

3 Data

3.1 Topics

We work with a list of topics from the test data in the DUC 2005 challenge. A topic has an identifier, category (general/specific), title and a sequence of statements or questions, for example:

Workshop on TextGraphs, at HLT-NAACL 2006, pages 29–32,
New York City, June 2006.

```
d307b
specific
New Hydroelectric Projects
What hydroelectric projects are planned
or in progress and what problems are
associated with them?
```

We apply MiniPar to the titles and contents of the topics, and to all documents. The output is post-processed to produce dependency pairs only for open-class words. The dependency pairs bypass prepositions and subordinators/coordinators between clauses, linking the corresponding open-class words. After post-processing, the topic will be represented like this:

```
QUESTION NUMBER: d307b
LIST OF WORDS:
associate, hydroelectric, in, plan,
problem, progress, project, new, them
LIST OF PAIRS:
relation(project, hydroelectric)
relation(project, new)
relation(associate, problem)
relation(plan, project)
relation(in, progress)
relation(associate, them)
```

The parser does not always produce perfect parses. In this example it did not associate the phrase *in progress* with the noun *projects*, so we missed the connection between *projects* and *progress*.

In the next step, we expand each open-class word in the topic with all its *WordNet* synsets and one-step hypernyms and hyponyms. We have two variants of the topic file: with all open-class words from the topic description $Topics_{all}$, and only with nouns and verbs $Topics_{NV}$.

3.2 Documents

For each topic, we summarize a collection of up to 50 news items. In our experiments, we build a file with all documents for a given topic, one sentence per line, cleaned of XML tags. We process each file with MiniPar, and post-process the output similarly to the topics. For documents we keep the list of dependency relations but not a separate list of words. This processing also gives one file per topic, each sentence followed by its list of dependency relations.

3.3 Summary Content Units

The DUC 2005 summary evaluation included an analysis based on *Summary Content Units*. SCUs are manually-selected topic-specific summary-worthy phrases which the summarization systems are expected to include in their output (Nenkova and Passonneau, 2004; Copeck and Szpakowicz, 2005). The SCUs for 20 of the test topics became available after the challenge. We use the SCU data to measure the performance of our graph-matching and path-search algorithms: the total number, weight and number of unique SCUs per summary, and the number of negative SCU sentences, explicitly marked as not relevant to the summary.

4 Algorithms

4.1 Topic↔sentence graph matching (GM)

We treat a sentence and a topic as graphs. The nodes are the open-class words in the sentence or topic (we also refer to them as keywords), and the edges are the dependency relations extracted from MiniPar's output. In order to maximize the matching score, we replace a word w_S in the sentence with w_Q from the query, if w_S appears in the *WordNet* expansion of words in w_Q.

To score a match between a sentence and a graph, we compute and then combine two partial scores:

S_N (node match score) the node (keyword) overlap between the two text units. A keyword count is equal to the number of dependency pairs it appears with in the document sentence;

S_E (edge match score) the edge (dependency relation) overlap.

The overall score is $S = S_N + WeightFactor * S_E$, where $WeightFactor \in \{0, 1, 2, ..., 15, 20, 50, 100\}$. Varying the weight factor allows us to find various combinations of node and edge score matches which work best for sentence extraction in summarization. When $WeightFactor = 0$, the sentence scores correspond to keyword counts.

4.2 Path search for topic keyword pairs (PS)

Here too we look at sentences as graphs. We only take the list of words from the topic representation. For each pair of those words, we check whether they both appear in the sentence and are connected in the sentence graph. We use the list of *WordNet*-expanded terms again, to maximize matching. The final score for the sentence has two components: the node-match score S_N, and S_P, the number of word pairs from the topic description connected by a path in the sentence graph. The final score is $S = S_N + WeightFactor * S_P$. $WeightFactor$, in the same range as previously, is meant to boost the contribution of the path score towards the final score of the sentence.

5 Experiments and results

We produce a summary for each topic and each experimental configuration. We take the most highly ranked (complete) sentences for which the total number of words does not exceed the 250-word limit. Next, we gather SCU data for each sentence in each summary from the SCU information files. For a specific experimental configuration – topic representation, graph algorithm – we produce summaries for the 20 documents with the weight factor values $0, 1, 2, ..., 15, 20, 50, 100$. Each experimental configuration generates 19 sets of average results, one

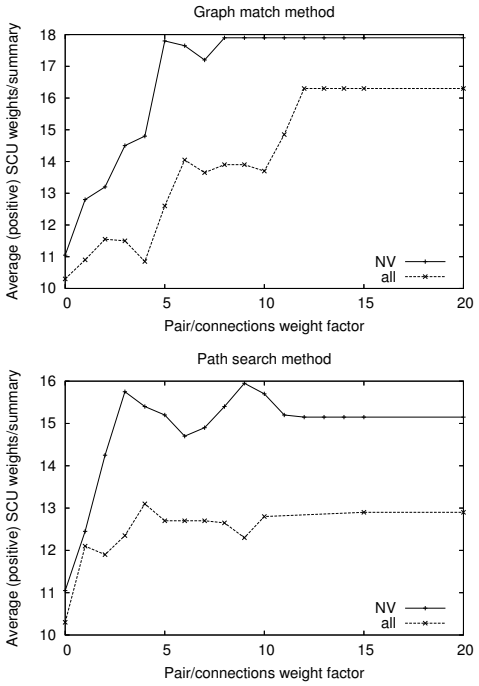

Figure 1: Average SCU weights for graph matching (GM) and path search (PS) with different topic representations

per weight factor. For one weight factor, we generate summaries for the 20 topics, and then average their SCU statistics, including SCU weight, number of unique SCUs and total number of SCUs. In the results which follow we present average SCU weight per summary. The number of unique SCUs and the number of SCUs closely follow the presented graphs. The overlap of SCUs (number of SCUs / number of unique SCUs) reaches a maximum of 1.09. There was no explicit redundancy elimination, mostly because the SCU overlap was so low.

We compare the performance of the two algorithms, GM and PS, on the two topic representations – with all open-class words and only with nouns and verbs. Figure 1 shows the performance of the methods in terms of average SCU weights per summary for each weight factor considered [1].

The results allow us to make several observations.

- Keyword-only match performs worse that either GM or PS. The points corresponding to keyword (node) match only are the points for which the weight factor is 0. In this case the dependency pairs match and paths found in the graph do not contribute to the overall score.
- Both graph algorithms achieve better performance for only the nouns and verbs from the topic than for all open-class words. If, however, the topic requests entities or events with specific properties, described by adjectives or adverbs, using only nouns and verbs may produce worse results.
- GM performs better than PS for both types of topic descriptions. In other words, looking at the same words that appear in the topic, connected in the same way, leads to better results than finding pairs of words that are "somehow" connected.
- Higher performance for higher weight factors further supports the point that looking for word connections, instead of isolated words, helps find sentences with information content more related to the topic.

For the following set of experiments, we use the topics with the word list containing only nouns and verbs. We want to compare graph matching and path search further. One issue that comes to mind is whether a combination of the two methods will perform better than each of them individually. Figure 2 plots the average of SCU weights per summary.

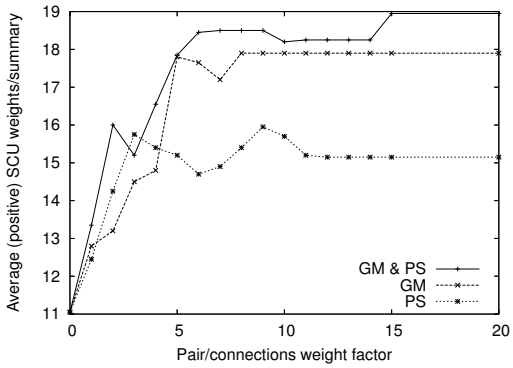

Figure 2: Graph matching, path search and their combination

We observe that the combination of graph matching and path search gives better results than either method alone. The sentence score combines the number of edges matched and the number of connections found with equal weight factors for the edge match and path score. This raises the question whether different weights for the edge match and path would lead to better scores. Figure 3 plots the results produced using the score computation formula $S = S_N + WeightFactor_E * S_E + WeightFactor_P * S_P$, where both $WeightFactor_E$ and $WeightFactor_P$ are integers from 0 to 30.

The lowest scores are for the weight factors 0, when sentence score depends only on the keyword score. There is an increase in average SCU weights

[1] The summary statistics level off above a certain weight factor, so we include only the non-flat part of the graph.

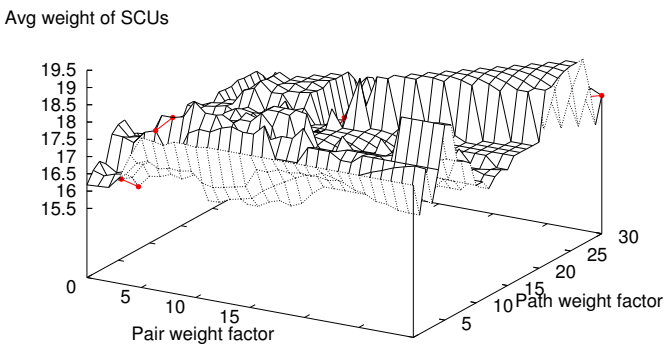

Figure 3: Graph match and path search combined with different weight factors

towards higher values of weight factors. A transparent view of the 3D graph shows that graph match has higher peaks toward higher weight factors than path search, and higher also than the situation when path search and graph match have equal weights.

The only sentences in the given documents tagged with SCU information are those which appeared in the summaries generated by the competing teams in 2005. Our results are therefore actually a lower bound – more of the sentences selected may include relevant information. A manual analysis of the summaries generated using only keyword counts showed that, for these summaries, the sentences not containing SCUs were not informative. We cannot check this for all the summaries generated in these experiments, because the number is very large, above 1000. An average summary had 8.24 sentences, with 3.19 sentences containing SCUs. We cannot say much about the sentences that do not contain SCUs. This may raise doubts about our results. Support for the fact that the results reflect a real increase in performance comes from the weights of the SCUs added: the average SCU weight increases from 2.5 when keywords are used to 2.75 for path search algorithm, and 2.91 for graph match and the combination of path search and graph match. This shows that by increasing the weight of graph edges and paths in the scoring of a sentence, the algorithm can pick more and better SCUs, SCUs which more people see as relevant to the topic. It would be certainly interesting to have a way of assessing the "SCU-less" sentences in the summary. We leave that for future work, and possibly future developments in SCU annotation.

6 Conclusions

We have studied how two algorithms influence summarization by sentence extraction. They match the topic description and sentences in a document. The results show that using connections between the words in the topic description improves the accuracy of sentence scoring compared to simple keyword match. Finding connections between query words in a sentence depends on finding the corresponding words in the sentence. In our experiments, we have used one-step extension in *WordNet* (along IS-A links) to find such correspondences. It is, however, a limited solution, and better word matches should be attempted, such as for example word similarity scores in *WordNet*.

In summarization by sentence extraction, other scores affect sentence ranking, for example position in the document and paragraph or proximity to other high-ranked sentences. We have analyzed the effect of connections in isolation, to reduce the influence of other factors. A summarization system would combine all these scores, and possibly produce better results. Word connections or pairs could also be used just as keywords were, as part of a feature description of documents, to be automatically ranked using machine learning.

References

Terry Copeck and Stan Szpakowicz. 2005. Leveraging pyramids. http://duc.nist.gov/pubs/2005papers/uottawa.copeck2.pdf.

Hoa Trang Dang. 2005. Overview of DUC 2005. http://duc.nist.gov/pubs/2005papers/OVERVIEW05.pdf.

Güneş Erkan and Dragomir Radev. 2004. LexRank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, (22).

Seeger Fisher, Brian Roark, Jianji Yang, and Bill Hersh. 2005. Ogi/ohsu baseline query-directed multi-document summarization system for duc-2005. http://duc.nist.gov/pubs/2005papers/ohsu.seeger.pdf.

Aria Haghighi, Andrew Ng, and Christopher Manning. 2005. Robust textual inference via graph matching. In *Proc. of HLT-EMNLP 2005*, Vancouver, BC, Canada.

Dekang Lin. 1998. Dependency-based evaluation of MiniPar. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proc. of EMNLP 2004*, Barcelona, Spain.

Rada Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proc. of ACL 2004*, Barcelona, Spain.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. In *Proc. of NAACL-HLT 2004*.