Query-Biased Text Summarization as a Question-Answering Technique

Yllias Chali
Department of Mathematics and Computer Science
University of Lethbridge
yllias@cs.uleth.ca

Stan Matwin and **Stan Szpakowicz**
School of Information Technology and Engineering
University of Ottawa
{stan, szpak}@site.uottawa.ca

Abstract

Text summarization may soon become a competitive method of answering queries asked of large text corpora. A query brings up a set of documents. These documents are then filtered by a summarizer: it constructs brief summaries from document fragments conceptually close to the query terms. We present the implementation of such a summarization system, based on lexical semantics, and discuss its operation. It is a configurable system, in the sense that the user will be able to choose among two or more implementations of every major step.

1. Introduction

A variety of text processing tools have been recently proposed to help Internet users digest textual information whose volume would otherwise be unmanageable. A text summarizer can be viewed as a back-end to an information retrieval system. A short summary produced in response to a user query can help determine whether the retrieved document is relevant; a longer summary or adequate excerpt could stand in for a complete document.

Text summarization may be applied to answering queries asked of large text corpora. A query brings up an initial set of documents that are next filtered by a summarizer: it constructs brief summaries from document fragments conceptually close to the query terms.

We present a system that operates in three phases. We are currently completing the implementation and working on the evaluation. This question answering system is partially modeled on a text summarization system procedure by extraction of sentences (Hovy & Radev 1998; Mani & Maybury 1997). The input text (a document retrieved by a search engine or by an information retrieval system) is first segmented at places where there are probable subtopic shifts; that is, every segment is likely to contain a cohesive discussion on one major theme. Next, the system classifies segments with regard to their subtopics, characterized by lexical chains (these are discussed in Section 3). Segments with strong evidence of the topic or topics suggested by the user's query are identified – they can be considered most relevant. Finally, sentences for the summary are extracted from these presumably most relevant segments.

The system includes several tools that rely on lexical analyses for the determination of query topics and segment topics. These tools are presented in Sections 2 through 5 of the paper. Some of these tools are now, and all of them eventually will be, available in more than one implementation. This will allow the summarizer to be variously configured by choosing the best set of modules for a given user or a given type of documents.

Section 6 of the paper contains a small example of the system's operation. The last section briefly discusses related work, offers a few conclusions, and presents a few pressing items of future work.

2. Segmentation

Segmentation of a natural language text is motivated by the observation that comprehension of longer texts may benefit from automatic chunking into smaller cohesive sections. The input text is divided in a way that reflects a meaningful grouping of contiguous portions of the text. The text is cut up into a linear sequence of adjacent segments. Probable segment boundaries are found at those paragraph breaks where a change in the distribution of lexical material signals one or more subtopic shifts.

Subtopic-based segmentation into groups of paragraphs should be useful for many text analysis tasks, including information retrieval and summarization. In our case, text segmentation is interesting for the following reasons.

- Segmentation identifies, with adequate probability, the paragraph breaks where the text changes topic. Thus a text, especially a short one, can comprise merely a single segment, or perhaps several different segments, when it touches on several distinct topics. A typical segment contains a few to several hundred words.
- Segmentation makes it possible to preselect for the

further, more costly semantically-based phases of summarization only segments most probably relevant to the query terms. This can considerably speed up the process of answering user queries.

- The resulting summaries may turn out better focused. This is again because the system produces a summary from topically more uniform segments. We assume that drastic topic shifts in a summary are undesirable, and that less relevant or quite irrelevant segments would not contribute much to an acceptable summary.
- The summary can be smoother if it is derived from a small set of segments most relevant to the query. A summary constructed from a complete document is likely to be less cohesive.

In keeping with the idea of configurable summarization, we have two segmenters. They are used independently now, but in the future we will reconcile their results for a more credible set of segments. The TextTiling system (Hearst 1997) and the Columbia University's Segmenter (Kan, McKeown, & Klavans 1998) are both in the public domain, and can be licensed for research purposes.

Segmentation is followed by a characterization of each segment (or, in the near future, only of the preselected segments) in terms of their lexical composition and their relation to the user's interests, suggested by the query.

3. Lexical chaining

Structural theories of text are concerned with identifying units of text that are about the "same thing". In such units, there is a strong tendency for semantically related words to be used. The notion of cohesion, introduced by (Halliday & Hasan 1976), is a device for "sticking together" different parts of the text to function as a whole. This is achieved through the use of grammatical cohesion (reference, substitution, ellipsis and conjunction) and lexical cohesion (occurrence of semantically related words). Lexical cohesion exists not only between two terms, but among sequences of related words, called lexical chains (Morris & Hirst 1991). Lexical chains provide an easy-to-determine context to aid in the resolution of word sense ambiguity. They also tend to delineate portions of text that have a strong unity of meaning.

We investigate how lexical chains can serve as an indicator of the text segment topic for the purpose of segment selection. Selected segments will contribute to an answer to the user's query. Lexical chain computation proceeds as follows.

1. We select the set of candidate words. To this end, we run a part-of-speech tagger (Brill 1992) on a text segment. Only the open class words that function as noun phrases or proper names are chosen for further processing.
2. The candidate words are "exploded" into word senses. In this step, all senses of a word are considered. They are located in an on-line thesaurus. For configurability, we have implemented lexical chain construction using two different thesauri: Roget's (Roget 1988) and WordNet (Miller *et al.* 1993). The latter is in the public domain, and there is a public-domain on-line version of the former.

   Lexical chains in a text can be identified using any lexical resource that relates words by their meaning. Information taken from WordNet is represented as follows. The WordNet database is composed of synonym sets called synsets. A synset contains one or more words that have similar meaning. A word may appear in many synsets, depending on the number of senses that it has, and each word sense may have a set of antonyms. Synsets are interconnected by several types of links that signal different lexical relations. For example, two synsets can be connected by a hypernym (respectively hyponym) link, which indicates that the target synset is a superodinate (respectively subordinate) of the source synset. Similarly, two synsets can be connected by a meronym (respectively holonym) link, when the target synset is a part (respectively an encompassing whole) of the source synset. These two types of relations are jointly named ISA (respectively INCLUDES) in our system.

   A word sense is represented as illustrated in *Figure 1*. At the first level we have synonyms and antonyms; next, direct hypernyms and hyponyms, and other close "relatives" such as meronyms or holonyms; next, words related to the first level words; and so on.

   A similar representation exists for information taken from Roget's thesaurus.
3. We find the semantic relatedness among the sets of senses. A semantic relationship exists between two word senses when we compare their representations and find a non-empty intersection between the sets of words. The strength of a semantic relationship is measured; this measure reflects the length of the path taken in the matching with respect to the levels of the two compared sets. This semantic relationship measure is given by the following formula:

   $$2 * Maxlevel - Level1 - Level2$$

   $Maxlevel$ is the maximum level under consideration, $Level1$ and $Level2$ are respectively the levels for the first and the second word sense.
4. We build up chains which are sets such as

   $$\{(word_1[sense_{11}, \ldots]), (word_2[sense_{21}, \ldots]), \ldots\}$$

   in which $word_i$-$sense_{ix}$ is semantically related to $word_j$-$sense_{jy}$ for $i \neq j$
5. We retain the longest chains relying on different preference for different relationships. Our order of preference, from the stronger to the weaker relationships, is this:

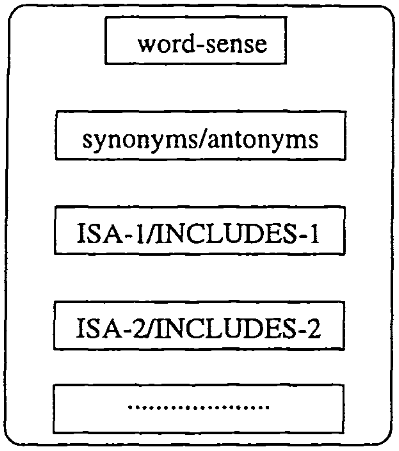

Figure 1: Word sense representation

word repetition ≫ *synonym/antonym* ≫ ISA-1/INCLUDE-1 ≫ ISA-2/INCLUDE-2 ≫ ...

This is handled in the algorithm by scoring lexical chains. A score sums up the semantic relationship measure between each pair of chain members.

We now show the output of the lexical chainer on fragment (1). The chains (2) and (3) have been computed using WordNet and the Roget's thesaurus, respectively. The numbers in parentheses are the word occurrence positions within the text.

(1) With its distant orbit - 50 percent farther from the sun than the earth - and slim atmospheric blanket, Mars experiences frigid weather conditions. Surface temperatures typically average about -60 degrees Celsius (-76 degrees Fahrenheit) at the equator and can dip to -123 degrees Celsius near the poles. Only the midday sun at tropical latitudes is warm enough to thaw ice on occasion, but any liquid water formed in this way should evaporate almost instantly because of the low atmospheric pressure.

(2) { orbit (4), degree (31 34 44), latitude (55) }
{ degree (31 34 44), occasion (63) }
{ degree (31 34 44), way (71) }
{ sun (11 52), earth (14), mars (20) }
{ weather (23), ice (61), water (67) }

(3) { orbit (4), sun (11 52), earth (14), weather (23), equator (38) }

Lexical chains are computed for each text segment, followed by segment selection.

4. Segment selection

This phase aims at selecting those text segments that are closely related to the topics suggested in the user's query. The summarization system does not analyze the query. We assume that the user has specified the query terms, as it is done, for example, when using an information retrieval system.

The segment selection algorithm proceeds as follows.

1. When the user posts a query, we consider a process of query-term sense disambiguation, if the query terms are related semantically and can be disambiguated. This is handled by a process akin to lexical chaining among the query terms. Suppose we choose the longest chain based on the order of preference of WordNet links, shown in step 5 of the lexical chaining algorithm. We contend that such a choice is equivalent to disambiguating the query terms that are members of this chain by choosing the appropriate sense.

   If query terms are not semantically related at all, sense disambiguation cannot be performed. In that case, we assume that the user has validated the query through an interface that requests, for example, the choice among possible senses of each term.

2. The query is expanded by adding semantically related words to the query terms. This allows us to get a query term representation similar to the chain member representation. Consequently, the two representations can be matched easily.

3. For each text segment, we compute the number of matches between the two representations (i.e. expanded query and the lexical chains of this segment). This leads to a ranking process by giving us a score for each segment with regard to a query term. The rank is the total number of matches between the expanded query and all the lexical chains in this segment. The scores are represented as $score(term_i, segment_j)$ in formula (4).

4. Based on the scores of query term occurrences in segments, we rank the text segments using a variation of the $tf * idf$ scoring technique:

   (4) $score(seg_j) = \sum_{i=1}^{p} score(term_i, seg_j) \; / \; s_i$

   where s_i is the number of segments in which $term_i$ occurs, and p is the number of query terms.

   The top n segments - with the highest scores - are chosen for the process of sentence extraction.

We observe that the process of segment selection can substantially limit the amount of matching.

5. Sentence extraction

We have adopted Sanfilippo's method (1998) of computing saliency and connectivity of text units such as, for example, sentences. He proposes to count the number of shared words between units. Text units which contain a greater number of shared words are more likely to provide a better abridgment of the original text. Sanfilippo gives two reasons.

- The more often a word with information content occurs in a text, the more topical to the text the word is likely to be.

- The greater the number of times two text units share a word, the more connected they are likely to be.

In our case, these shared words are those members of the chains that contribute to segment selection, that is, match the query terms. The text units are sentences.

Each sentence is ranked by summing up the number of shared chain members over sentences. More precisely, the score for *sentence*$_i$ is the number of words that belong to *sentence*$_i$ and also to those chains which have been considered in the segment selection phase.

The answer to the user's original query is the ranked list of top-scoring sentences. If the answer expected by the user should be limited in size, a suitable front segment of the list is returned.

6. Example

Consider the following text (5), whose two paragraphs are two segments according to the Columbia University segmenter.

(5) With its distant orbit - 50 percent farther from the sun than the earth - and slim atmospheric blanket, Mars experiences frigid weather conditions. Surface temperatures typically average about -60 degrees Celsius (-76 degrees Fahrenheit) at the equator and can dip to -123 degrees Celsius near the poles. Only the midday sun at tropical latitudes is warm enough to thaw ice on occasion, but any liquid water formed in this way should evaporate almost instantly because of the low atmospheric pressure.

Although the atmosphere holds a small amount of water, and water-ice clouds sometimes develop, most Martian weather involves blowing dust or carbon-dioxide. Each winter, for example, a blizzard of frozen carbon-dioxide rages over one pole, and a few meters of this dry-ice snow accumulate as previously frozen carbon-dioxide evaporates from the opposite polar cap. Yet even on the summer pole, where the sun remains in the sky all day long, temperatures never warm enough to melt frozen water.

The lexical chains have been computed using WordNet. For the first segment, the chains appear in (2) in Section 3. For the second segment, the chains are shown in (6).

(6) { amount (7), pole (35 60), meter (39) }
{ atmosphere (3), cloud (12), weather (17), blizzard (28) }
{ amount (7), winter (24), summer (59), day (69) }
{ atmosphere (3), weather (17), snow (43) }
{ water (9), weather (17), snow (43) }

If the user's query contains the terms "Mars" and "weather", the sentences will be ranked as follows:

(7) $S_{1,1} \simeq S_{2,1}$ [7] $\gg S_{1,3}$ [4] $\gg S_{2,2}$ [3]

Here, $S_{i,j}$ stands for sentence j within segment i, and specific score values are given in square brackets for the sake of illustration, and are not the scores computed by the system.

If the user's query contains the terms "Mars" and "ice" instead, the ranking will be as follows:

(8) $S_{2,1}$ [7] $\gg S_{1,1}$ [4] $\gg S_{1,3} \simeq S_{2,2}$ [3]

7. Discussion and future work

We have described a system for query-biased text summarization which is part of the Intelligent Information Access project (Chali *et al.* 1998; Lankester, Debruijn, & Holte 1998). A summary, as an abridgment of the original text, can be treated as an answer to the user query in an information retrieval session.

Question answering systems are typically based on knowledge bases and databases. Our system, on the other hand, relies mostly on knowledge acquired from natural language text. We process unrestricted text using linguistic resources, such as on-line thesauri.

In an information extraction (IE) task, a query and a set of predefined templates are given. An IE system selects the best template, fills it and generates the content. A considerable progress made so far has been amply documented in the literature, for example in (Grishman & Sundheim 1996), (Appelt *et al.* 1993), (Radev & McKeown 1998).

However successful this approach to information extraction has been, doubts remain about its effectiveness. According to (Hovy & Marcu 1998):

- It seems that information extraction templates work only for very particular template definitions. Can this approach scale up?
- What about information that does not fit into any template?

To overcome these limitations, we consider query-biased summarization for an arbitrary text without requiring its full understanding, but using widely available knowledge sources.

(Barzilay & Elhadad 1997) investigate the production of summaries based on lexical chaining. The summaries are built using a scoring based on chain length, and the extraction of significant sentences is based on heuristics using chain distribution. For example, a sentence may be chosen if it contains the first appearance of a chain member in the text. In our system, summaries are built by extracting from the text the sentences most relevant and pertinent to the user's query.

We are aware that our system requires an evaluation in order to confirm its effectiveness. To this end, we are participating in the Question Answering track competition of the next Text Retrieval Conference TREC-8.

Our system relies on linguistically involved and subtle, but rather slow, public-domain resources. We realize that the system may pose efficiency problems. We propose to address them in two ways.

- We will construct and apply in the system a scaled-down, but optimized for access time, version of WordNet.

- We will control the extent of semantic expansion in step 2 of the segment selection phase. Limiting expansion is certain to speed up the system, but we must experiment to find such parameter settings that do not impair the selectivity of our algorithm.

Acknowledgment

This work was carried out when the first author was at the University of Ottawa, under the support grant by the Natural Sciences and Engineering Research Council of Canada.

References

Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D. J., and Tyson, M. (1993). Fastus: A finite-state processor for information extraction from real-world text. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, IJCAI'93*, 1172–1178.

Barzilay, R., and Elhadad, M. (1997). Using lexical chains for text summarization. In *ACL/EACL Workshop on Intelligent Scalable Text Summarization*, 10–17.

Brill, E. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing, ACL*, 152–155.

Chali, Y., Barker, K., Copeck, T., Matwin, S., and Szpakowicz, S. (1998). The design of a configurable text summarization system. Technical report, TR-98-04, School of Information Technology and Engineering, University of Ottawa.

Grishman, R., and Sundheim, B. (1996). Message Understanding Conference - 6: A brief history. In *Proceedings of the International Conference on Computational Linguistics*, 466–471.

Halliday, M., and Hasan, R. (1976). *Cohesion in English*. London: Longman.

Hearst, M. A. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics* 23(1):33–64.

Hovy, E., and Marcu, D. (1998). Tutorial notes on automated text summarization. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*.

Hovy, E., and Radev, D., eds. (1998). *Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization.*

Kan, M.-Y., McKeown, K. R., and Klavans, J. L. (1998). Linear segmentation and segment relevance. In *Proceedings of 6th International Workshop of Very Large Corpora (WVLC-6)*, 197–205.

Lankester, C., Debruijn, B., and Holte, R. (1998). Prototype system for intelligent information access: Specification and implementation. Technical Report TR-98-05, School of Information Technology and Engineering, University of Ottawa.

Mani, I., and Maybury, M., eds. (1997). *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization.*

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1993). Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University.

Morris, J., and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1):21–48.

Radev, D. R., and McKeown, K. R. (1998). Generating natural language summaries from multiple on-line sources. *Computational Linguistics* 24(3):469–500.

Roget, P. M. (1988). *Roget's Thesaurus*. London: Longman.

Sanfilippo, A. (1998). Ranking text units according to textual saliency, connectivity and topic aptness. In *Proceedings of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume II, 1157–1163.