Query-based extracting: how to support the answer?

Wauter Bosma
Human Media Interaction
University of Twente
the Netherlands
bosmaw@cs.utwente.nl

Abstract

Human-made query-based summaries commonly contain information not explicitly asked for. They answer the user query, but also provide supporting information. In order to find this information in the source text, a graph is used to model the strength and type of relations between sentences of the query and document cluster, based on various features. The resulting extracts rank second in overall readability in the DUC 2006 evaluation. Employment of better question answering methods is the key to improve also content-based evaluation results.

1 Introduction

In recent years, attention has shifted from generic summarization toward query-based summarization. While a generic summary includes information which is central to the source documents, a query-based summary should formulate an answer to the query.

At DUC 2005, the answer of most participants to the query-based summarization task was to include sentences in the summary which in one way or another matched the query. However, hand-crafted model summaries show that human summarizers do not only include direct answers to the query in their summaries, but also supporting information. This information provides general background knowledge, or other information to make the actual answer more understandable or to make the reader more receptive to the answer. For instance, in response to the question which measures have been taken to improve automobile safety, three of the DUC 2006 model summaries mentioned laws enforcing seat belt use. Two out of these three summaries first mentioned the reasons why these steps are deemed necessary. Moreover, extrinsic evaluations have shown that users appreciate receiving more information than just an answer to the explicitly formulated information need (Lin et al., 2003; Bosma, 2005).

One could argue that information overlap between documents can be exploited to find this supporting information, assuming that salient information is present in many source documents. However, it seems that there is no consistent relation between the number of occurrences of a particular piece of information in model summaries and source documents. For instance, three out of four model summaries about international Carter Center activities mention Carter Center's founding date, while this information occurs only once in the corresponding cluster of 25 source documents.

Several participants of DUC 2005 took cohesion of source documents into account, but used it as a means to enhance cohesion of the summary, rather than to find new information. On the other hand, Blair-Goldensohn (2005) reported positive results from experiments with the Columbia system giving a relevance bonus to sentences nearby sentences answering the query. Blair-Goldensohn also suggested that exploiting text structure could help improving linguistic quality. This paper describes the Twente summarization system, which uses a cross-document structural analysis of the source document set to generate a query-based extract, thereby copying part of the structure from the document set to the summary.

The focus of the summarization system is to create readable summaries, in particular paying attention to coherence and non-redundancy. This paper argues that answers to the query should be supported with non-answers in order to create a readable and coherent summary. An entailment system is used to find answers to the query, and to detect entailment between sentences across documents. Sentences which are entailed by another sentence of a source document contain redundant information, and are not included in the summary. As an indication of a relation between two sentences within a document, layout and cosine similarity is used. These sources of information are combined to form a graph representation of the document set, in which the relevance of a sentence is measured as the graph distance from the sentence to a query sentence. This method allows a sentence to be included in the summary (a) if it is likely to answer the query, or (b) if it is closely related to a sentence answering the query. The latter case may occur only if the related sentence is included in the summary as well.

In section 2, the features used to decide which sentences are relevant are discussed. Section 3 describes how an existing graph-based approach was extended to multi-document summarization by constructing a graph network of sentences across documents, one of which is the query. After discussing evaluation results (section 4), this paper wraps up with conclusions in section 5.

2 Features for multi-document extraction

The DUC 2006 summarization challenge is to create a 250-word summary from a set of 25 source documents, given a topic—a query—stating an information need. The Twente summarizer is restricted to extracting, i.e. each sentence in the summary is also present in one of the source documents. No text is rewritten or revised.

A DUC query is an expressed information need, typically formulated as one or more questions or imperatives. In this paper, responses to the explicitly expressed information need are called *answers*. Other information which satisfies an implicit information need that may be present is called *answer supporting information*.

Ideally, a query-based multi-document extraction system produces the set of sentences *most relevant* to the query in an *appropriate ordering*, while not violating the word limit. But which sentences are most relevant, and what is an appropriate ordering?

First of all, a sentence which gives a direct answer to the query is considered relevant and should be included in the summary. The dependency tree alignment algorithm of Marsi et al. (2006) is used to find answers to the query.[1]The algorithm was designed for recognizing textual entailment, and exploits hierarchical syntactic sentence structure by finding the largest common subtree between query sentence and source document sentence. Lemma equivalence (van den Bosch and Daelemans, 1999) and WordNet synonymy and hyponymy (Fellbaum, 1998) is

[1] The algorithm of Marsi et al. (2006) uses two parameters, named SP and PW, to configure the weight distribution among aligned nodes. In the summarization system, the alignment score of the entailment algorithm equals the size of the largest common sub tree as a fraction of the size of the hypothesis, i.e. the parameters are assigned the values $SP = 1 - \frac{|v'_j|}{|v'|}$ and $PW = \frac{1}{|v|}$, where $|v|$ is the number of nodes in the sub tree of which node v is the root.

used for alignment on the lexical level. The MaltParser system (Nivre and Scholz, 2004) is used for syntactic analysis of query and document sentences.

The use of an alignment algorithm for finding answers is based on the observation that recognizing a question/answer relation is similar to recognizing an entailment relation, and both can be found using syntactic structure. Bouma et al. (2006) show that it is likely that a sentence answers a question if the syntactic structure of question and candidate answer sentence sentence is similar.

Recognizing textual entailment is also useful for detecting redundancy across documents. If a sentence in one document is entailed by a sentence in another document, and the latter sentence is selected for inclusion in the summary, the entailed sentence should not be in the summary, as to avoid redundancy.

In addition to direct answers to questions, sentences which elaborate on answers are also included. Lacking a system for automated detection of discourse-level relations as used in Bosma (2005), layout information is used to relate sentences within a document. All source documents contain prior annotation of paragraph boundaries. Because the first sentence of a paragraph tends to contain the most important information, if a sentence of a paragraph is decided to be relevant, the first sentence of the paragraph is most likely to be relevant as well. And vice versa, if the first sentence of a paragraph is included, another sentence from the same paragraph is more likely to be relevant than a sentence from another paragraph.

Another means to discover related sentences is by measuring cosine similarity. This and other methods based on word overlap have been previously used in text summarization (e.g. Erkan and Radev, 2004). From a high cosine similarity between sentences it follows that the sentences have common terms, but it is difficult to distinguish between a redundancy relation and other semantic relations from cosine similarity, in contrast to entailment. This makes cosine similarity less suitable to detect cross-document relations. Therefore, for sentences across documents, solely redundancy—which follows from the entailment relation—is used to determine whether they are eligible for inclusion in the summary. On the other hand, sentences in the same document rarely have a redundancy relation, making cosine similarity more suitable for sentences of the same document.

Including sentences containing redundant information in the summary should obviously be avoided. If a sentence A in one document is entailed by a sentence B in another document, at most one of them should be included in the summary. But if a third sentence C elaborates on A, it most probably also elaborates on B. This makes C a more likely candidate for inclusion, if B is in the summary as well. In other words, a sentence which is entailed by a sentence in the summary is not eligible for inclusion in the summary, but a sentence closely related to the entailed (redundant) sentence is.

Finally, the length of a sentence is taken into account. Longer sentences typically depend less on contextual interpretation and contain less anaphora than shorter sentences. Hence, the extraction algorithm is biased toward extracting longer sentences.

To summarize, alignment of dependency trees is measured to find answers to questions, and to detect redundancy across documents. Paragraph boundaries are used as an indication of structural relations between sentences, and cosine similarity is used to find semantic relations between sentences within a document.

The extracted sentences are ordered by their original ordering in the source documents, and sentences from documents from which more sentences are extracted are written first.

3 Proximity Graphs

Bosma (2005) shows how the relevance value of a sentence can be derived from the graph distance of the sentence to the query. This is done using a graph in which nodes represent sentences. In this previous work, prior selection of an answer sentence by a question answering system is assumed. The distance from the answer sentence to each of the other sentences is then computed by taking the length of the shortest path connecting the two sentences in the graph. The sentences closest to the answer sentence are considered the most relevant, and are included in the summary.

In this paper, this approach is extended in order to make it suitable for the DUC task by making two significant modifications to the summarization system. First, rather than relying on a prior question answering step, answers are related to the query explicitly by creating an integrated graph of all documents, including the query sentences.

Second, the previous approach used handcrafted analyses of the rhetorical structure of source documents for summarization. This approach is less feasible for the DUC task, which involves large quantities of text. To perform the DUC 2006 task, several automatically derived factors to determine the distance between two sentences are combined, rather than using rhetorical analyses.

Figure 1 shows an overview of the extraction process. From the document set, three graphs are created, each graph containing a node for each sentence in the query as well as in the source documents. Thus, the nodes of the three graphs are identical, but the edges and their labels (reflecting the strength and type of a relation between between two sentences) depend on the algorithm used: entailment, layout or cosine similarity.

Edges are directed and labeled by their strength. A closer relation is represented by a higher strength value. The distance between two sentences connected by an edge is calculated as p^{-1}, where p is the strength of the edge. This is inspired by electrical circuits, where resistance is the inverse of conductance. The distance from one sentence to another is the sum of the distances of the shortest path that connects them. Since the edges are directed, $distance(a, b) = distance(b, a)$ is not necessarily true.

In addition to a strength value, an edge may have a flag marking the relation type it represents as ‘redundant’. In Figure 1, this is denoted by ‘r’. If the final edge of the shortest path from a query sentence to a document sentence represents a redundancy relation, the sentence cannot be included in the summary.

The graphs for entailment, layout and cosine similarity are then merged into a combined graph. For each pair of nodes (v_i, v_j), the strength of the edge between those nodes is computed according to equation 1, where the triple $(v_i, v_j, p_{i,j})$ is an edge from v_i to v_j with strength $p_{i,j}$; G_k is one of the graphs to be combined; and $sscore(v_j)$ is a function which returns the appropriateness of a sentence, and can be used to favor sentences over other sentences, based on their form. Finally, $sscore(v_j) = \sqrt{|v_j|}$, where $|v_j|$ is the number of characters in the sentence corresponding to v_j.

$$strength(v_i, v_j) = \sum \{sscore(v_j) \cdot p_{i,j} \mid (v_i, v_j, p_{i,j}) \in \bigcup_k G_k\} \tag{1}$$

In the first graph, the edges and corresponding labels are based on the dependency tree alignment algorithm. The alignment value of the two sentences T and H, $align(T, H)$ is the fraction of nodes in the dependency tree of H which are aligned with a node in the dependency tree of T, ranging from 0 to 1. For each query sentence Q and source document sentence A, an edge from Q to A is created, labeled $0.15 \cdot align(A, Q)$. For each of the 10 sentences best aligned with the query, A_i, and each sentence R in the cluster which is not from the same document as A_i, an edge marked ‘redundant’ is created from A_i to R with strength $5 \cdot align(A_i, R)$.

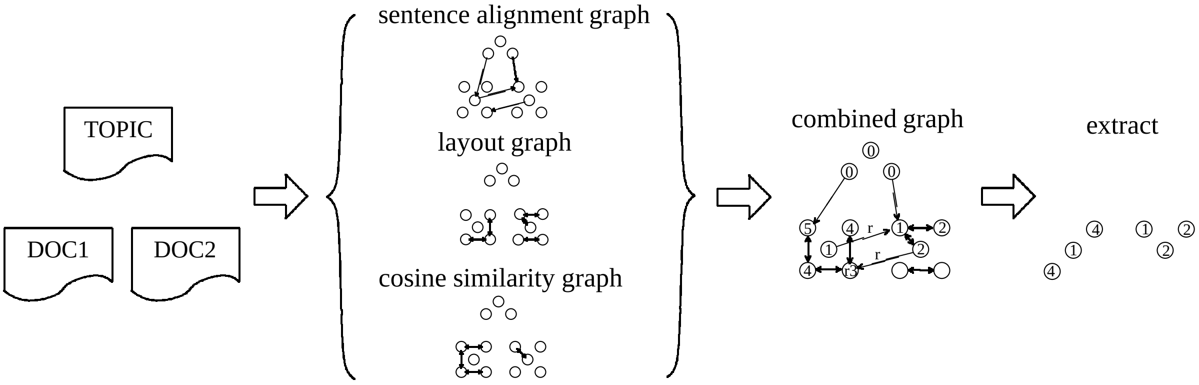

Figure 1: Overview of the extraction process. Although for clarity reasons no edge labels are displayed, all graphs are labeled, with labels representing the distance between two sentences.

The second graph is based on paragraph boundaries in source documents. For each paragraph, a bidirectional edge is created from the first sentence to each of the other sentences in the paragraph with strength 0.1.

Edges of the third graph are created based on the similarity between two sentences within a document. Sentences are represented as vectors of the $tf \cdot idf$ values of their lemmas. Then, for all possible pairings of sentences from the same document, a bidirectional edge is created, of which the strength is the cosine similarity of the two sentences (ranging from 0 to 1).

After the combined graph is constructed, the sentences with the smallest distance from a query sentence are extracted. The extraction procedure stops when including the next closest sentence would result in a violation of the 250-word limit.

The order in which sentences are presented in the summary is, where possible, the same as the sentence order in the source text. If the summary contains sentences from more than one source document, the order of documents depends on the length of the shortest path from the query to any sentence of the document: sentences from the document with the sentence most relevant to the query are presented first.

4 DUC Evaluation Results

Our effort to generate coherent extracts resulted in good DUC 2006 evaluation results of linguistic quality. Figure 2 compares manual DUC 2006 assessments by NIST assessors of

- baseline summaries consisting of the leading sentences of the most recent document of each cluster, up to 250 words;
- the median performance of 34 submissions;
- this submission;
- the best submission;
- hand-crafted summaries.

On average over all summaries and all evaluated aspects of linguistic quality (Figure 2 (a)–(e)), this submission performed second-best of 34 submissions. Although scores for all linguistic quality aspects are above the median, the referential clarity score is relatively low because no co-reference resolution or any form of sentence revision is involved.

Unfortunately, the Twente summarizer cannot yet show likewise performance for content-based evaluation metrics, and scores just below the median. Responsiveness (Figure 2 (f)) is a manual evaluation of how well a summary responds to the information need expressed in the query; and Figure 3 shows the resulting

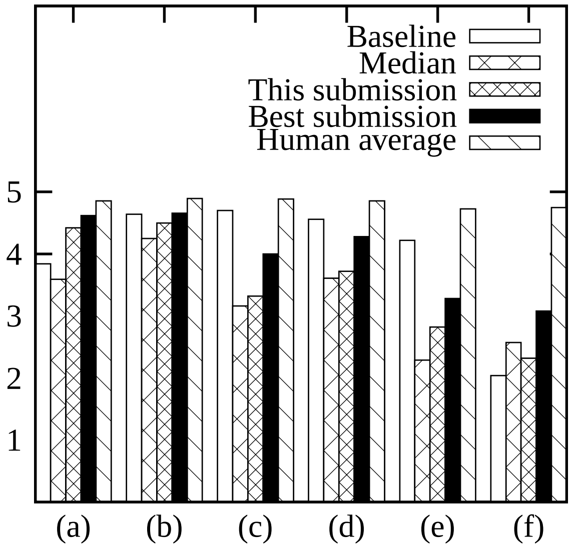

Figure 2: Average human assessment of various aspects of the quality of 50 summaries: (a) grammaticality, (b) non-redundancy, (c) referential clarity, (d) focus, (e) structure and coherence, (f) responsiveness as evaluated by NIST assessors on a five point Likert scale.

score of various methods to automatically measure how well information in the summary resembles information contained in hand-crafted model summaries.

An explanation for this is that adding supporting information reduces content-based scores for two reasons. First, an answer can be supported in various ways, one not necessarily better than the other. The result is that variation in the choice of information by summarizers may be even greater for answer supporting information than for answers. Consequently, performance drops if more supporting information is added. This is reflected by the low Pyramid evaluation score, which focuses on content unit overlap, and is therefore especially sensitive to such discrepancies.

Further research will have to point out whether agreement between human summarizers indeed varies between answers and non-answers. Content-based evaluations typically make the assumption that information overlap between model summaries and system-generated summaries is quantifiable as a basis for system performance. If agreement between human summarizers is indeed greater for supporting information, evaluation metrics should take this into account.

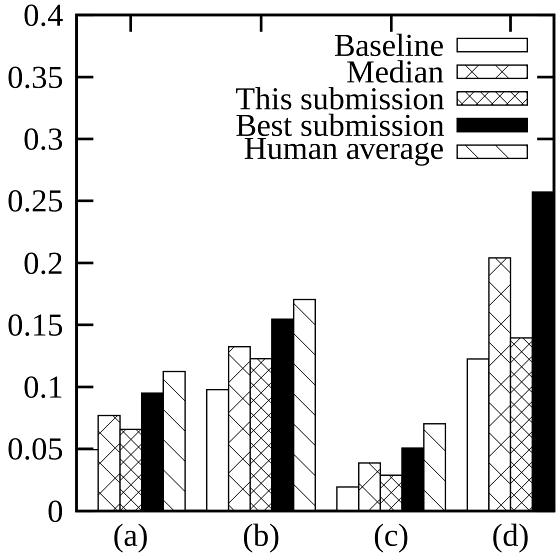

Figure 3: Average score of 50 summaries as produced by NIST and Columbia University using four evaluation metrics: (a) ROUGE-2, (b) ROUGE-SU4, (c) Basic Elements, (d) Pyramid.

Second, if the alignment algorithm fails to find a correct answer, information supporting this answer may also be irrelevant to the query. A problem encountered with the alignment algorithm is that queries are formulated in a very general way, while the documents discuss specific issues or events. In order to bridge this gap in specificity, reasoning with common sense and world knowledge is required. For instance, consider the following query statement and sentence from the corresponding document cluster.

Query: "What devices and procedures have been implemented to improve automobile safety?"

Document: "Seat belts have also been greatly improved both for comfort and for holding one in place in the event of a sudden stop."

The above document sentence provides (part of) an answer to the query, but the sentences only have the word ‘improve’ in common. Moreover, ‘improve’ in the query applies to ‘automobile safety’, while in the document, it applies to ‘seat belts’. In order to recognize the answer, we have to know that a seat belt is a device, that ‘holding one in place in the event of a sudden stop’ is important for ‘automobile safety’, and that improvement of a device implies that the device has been implemented.

This probably hurts more in a system trying to find supporting information. If document sentences are directly matched with the query, it is more likely that at least a fraction of the ‘answers’ in the summary are correct than if the summary elaborates on a possibly incorrect answer.

5 Conclusion

The coherence-based approach to query-based extracting presented here appeared to be one of the top performers in overall linguistic quality in the DUC 2006 multi-document summarization task. The novelty of this approach is that a generated summary may contain answer sentences as well as answer supporting sentences, resulting in a greater coherence. These results are motivating to continue research on coherence aware summarization.

Acknowledgements

This work is funded by the Interactive Multimodal Information Extraction (IMIX) program of the Netherlands Organization for Scientific Research (NWO).

References

Sasha Blair-Goldensohn. From definitions to complex topics: Columbia University at DUC 2005. In *Proceedings of Document Understanding Workshop*, Vancouver, Canada, 2005.

Antal van den Bosch and Walter Daelemans. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 285–292, San Francisco, CA, USA, 1999.

Wauter Bosma. Extending answers using discourse structure. In H. Saggion and J.L. Minel, editors, *RANLP Workshop on Crossing Barriers in Text Summarization Research*, pages 2–9, Borovets, Bulgaria, 2005. Incoma Ltd.

Gosse Bouma, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. Question answering for Dutch using dependency relations. In *Proceedings of the CLEF2005 Workshop*, Vienna, Austria, 2006. Springer.

Güneş Erkan and Dragomir R. Radev. Lexrank: graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, 2004.

Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. Language, Speech, and Communication Series. The MIT Press, Cambridge, MA, USA, 1998.

Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. The role of context in question answering systems. In *CHI ’03 Extended Abstracts On Human Factors in Computing Systems*, pages 1006–1007, New York, NY, USA, 2003. ACM Press.

Erwin Marsi, Emiel Krahmer, Wauter Bosma, and Mariët Theune. Normalized alignment of dependency trees for detecting textual entailment. In B. Magnini and I. Dagan, editors, *Second PASCAL Recognising Textual Entailment Challenge*, pages 56–61, Venice, Italy, 2006. PASCAL.

Joakim Nivre and Mario Scholz. Deterministic dependency parsing of english text. In *Proceedings of COLING 2004*, pages 23–27, Geneva, Switzerland, 2004.