Enhancing Query-oriented Summarization based on Sentence Wikification

Yajie Miao, Chunping Li
School of Software
Tsinghua University
Beijing 100084, China

yajiemiao@gmail.com, cli@tsinghua.edu.cn

ABSTRACT
Query-oriented summarization is primarily concerned with synthesizing an informative and well-organized summary from a document collection for a given query. In the existing summarization methods, each individual sentence in the document collection is represented as Bag of Words (BOW). In this paper, we propose a novel framework which improves query-oriented summarization via sentence wikification, i.e., enriching sentence representation with Wikipedia concepts. Furthermore, we exploit semantic relatedness of Wikipedia concepts as a smoothing factor in sentence wikification. The experiments with benchmark dataset show that sentence wikification performs effectively for improving query-oriented summarization, and helps to generate more high-quality summaries.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Retrieval models*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis.*

General Terms
Algorithms, Performance, Experimentation.

Keywords
Query-oriented Summarization, Sentence Wikification, Semantic Relatedness.

1. INTRODUCTION
Given a specific query, query-oriented summarization (QS) aims to automatically extract salient information from a document set and assemble them into concise answers in natural language. This task subsumes interesting applications. For instance, though search engines can respond to users' queries by returning lists of web pages, browsing the pages for desired information is either time-consuming or unachievable. In this case, it would be nice if we summarize the points on the target query, which helps users to digest the returned pages easily. A distinct characteristic of QS is that the sentences included in the summary are required to be closely relevant to the query. Therefore, the performance of QS relies highly on accurate measurement of text similarity. Traditionally, QS is based on the BOW approach, in which both the query and sentences are represented with word vectors. This approach suffers from the shortcoming that it merely considers lexical elements (words) in the documents, and ignores semantic relations among sentences.

In this paper, we examine sentences on a different dimension, i.e., Wikipedia concepts. Through sentence wikification, each sentence is mapped to a vector whose elements are Wikipedia concepts. In this feature space, we get the *Concept* similarity which serves to complement the original *Word* similarity derived from BOW. Then, we take *semantic relatedness* of Wikipedia concepts into account. We argue that two different concepts can be taken as matched if they are semantically close to each other. Following this, we propose the *Matrix* similarity, in which semantic relatedness is utilized for smoothing the process of concept matching. Finally the renewed sentence similarity resulting from wikification is employed to benefit query-oriented summarization.

To the best of our knowledge, this study is the first attempt to address the problem of combining sentence wikification with QS. We conduct extensive experimental studies to evaluate the proposed framework on the DUC 2005 dataset. The experiments show that sentence wikification indeed takes effect in boosting the performance of QS. Also, we observe that the Matrix similarity brings more significant improvements than Concept.

The rest of the paper is organized as follows. Section 2 introduces background information on query-oriented summarization. Section 3 presents our proposed framework. In Section 4, we show and discuss the experimental results. Finally, we have conclusion in Section 5.

2. QUERY-ORIENTED SUMMARIZATION
In this section, we introduce two existing solutions to QS. Formally, we denote the given query as q and the collection of documents as D. The goal of QS is to generate a summary which best meets the information needs expressed by q. To do this, a QS system generally takes two steps: first, the sentences in D are ranked with respect to q; second, top sentences are selected until the length of the summary is reached. For convenience, we let S denote all the sentences in D.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

A straightforward method [9], i.e., *TFIDF*, is to compute text similarity between the query and sentences, and rank the sentences based on this value. Both the query and the sentences can be represented with TF*IDF vectors. Therefore, the query-sentence similarity is naturally obtained as the cosine value of their TF*IDF vectors. Since this method is quite simple, we do not give elaborations on it.

Graph-based models [1, 2, 3, 5, 8] have been proved to be effective in sentence ranking and summary generating. In these models, a graph is constructed in which each node represents a sentence in S. Each edge measures the similarity between the corresponding pair of sentences. We consider two factors when deciding whether sentence s_i is selected to be included in the summary. First, s_i is relevant to the query q. Second, s_i is similar with other sentences which have high query-sentence similarity. This idea is captured by the model in Figure 1, which mixes query-sentence and sentence-sentence similarity together. In the figure, we denote the similarity of sentence s_i with the query q as $sim(s_i, q)$, and the similarity between sentence s_i and s_j as $sim(s_i, s_j)$. Then, using a computation process based on Random Walk, the saliency score for sentence s_i can be calculated iteratively as follows.

$$Score^{(n+1)}(s_i) = d \cdot \frac{sim(s_i, q)}{\sum_{s_j \in S} sim(s_j, q)} + (1-d) \cdot \sum_{s_j \in S} \frac{sim(s_i, s_j)}{\sum_{s_k \in S} sim(s_j, s_k)} Score^{(n)}(s_j), \quad (1)$$

where $Score^{(n)}(s_i)$ is the score of s_i in n^{th} iteration, d is a combination coefficient for trading off the two parts. We call this model as *Graph* in the following formulations.

3. THE PROPOSED FRAMEWORK

3.1 Sentence Wikification

In the traditional BOW approach, each sentence s_i is mapped to a vector of words, that is,

$$wordvector_i = \{tfidf_{s_i}^{w_1}, tfidf_{s_i}^{w_2}, \cdots, tfidf_{s_i}^{w_N}\}, \quad (2)$$

where $tfidf_{s_i}^{w_j}$ is the TF*IDF value of word w_j in sentence s_i, and N is the total number of words. Then the similarity between two sentences s_i and s_j is measured by the cosine value of $wordvector_i$ and $wordvector_j$. This similarity is shortly named as *Word*, which is denoted as $wordsim(s_i, s_j)$.

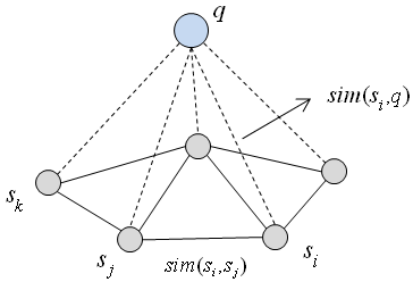

Figure 1. The Graph model.

Sentence wikification is the practice of representing a sentence with a set of Wikipedia concepts. We take the *exact-match* strategy introduced in [4] as our wikification method. Specifically, to wikify a sentence s_i, we scan this sentence and find Wikipedia concepts that appear explicitly in it. To find high-quality Wikipedia concepts, we also adopt extra operations such as excluding meaningless concepts and merging partial concepts. For instance, for the sentence "How do European Union countries feel about the US opposition to the Kyoto Protocol?", the concepts "Kyoto", "Protocol" and "Kyoto Protocol" (all of them appear in the sentence) should be treated as a single concept "Kyoto Protocol". In addition, the concepts "Position" and "Proto", though in the sentence, obviously cannot act as interpretation of the sentence, and thus should be eliminated.

These searched concepts are used to comprise the *concept vector* for the sentence. Formally, the sentence s_i is associated with a concept vector, that is,

$$conceptvector_i = \{\text{var}_{s_i}^{c_1}, \text{var}_{s_i}^{c_2}, \cdots, \text{var}_{s_i}^{c_W}\}, \quad (3)$$

where $\text{var}_{s_i}^{c_j}$ is a binary variable which indicates whether concept c_j appears in sentence s_i, and W is the total number of Wikipedia concepts appearing in S. Then the Concept similarity between s_i and s_j is the cosine similarity of their concept vectors.

$$conceptsim(s_i, s_j) = \frac{conceptvector_i \cdot conceptvector_j}{|conceptvector_i| \cdot |conceptvector_j|}. \quad (4)$$

3.2 The Matrix Similarity

The Concept similarity is able to represent sentence similarity on the dimension of Wikipedia concepts. However, Concept is too "rigid" because it allows matching of two concepts only when they are identical. In other words, only concepts which are shared by two sentences can contribute to their Concept similarity. As a result, some concept vectors, such as {Kyoto protocol, Emissions trading, Carbon dioxide} and {Global warming, Greenhouse gas, Fossil fuel}, have no Concept similarity, though they are quite close according to human judgment. To solve this problem, we turn to semantic relatedness of Wikipedia concepts, a value indicating the extent to which two Wikipedia concepts are close to each other, e.g., "Kyoto protocol" is closer to "Global warming" than to "Financial crisis". An effective and efficient method for semantic relatedness is Wikipedia Link-based Measure (WLM) [6, 7] which infers semantic relatedness from link structures in Wikipedia. The basic intuition behind WLM is that if two concepts are cited by (or link to) many common concepts, they are much likely to be highly related. A well-developed demo of WLM can be found at http://wdm.cs.waikato.ac.nz:8080/service?task=compare.

With sentence wikification, the set of sentences S are collaboratively represented with a *concept matrix*. We assume that semantic relatedness of each concept pair has been given. The computation of the Matrix similarity takes two steps.

First, with the semantic relatedness values, we create a *relatedness matrix*. The elements of the matrix represent semantic relatedness among concepts. To avoid excessive matching, we set a relatedness confidence, denoted as *confidence* , on the semantic relatedness values. Only two concepts whose semantic relatedness exceeds *confidence* can have a value in the relatedness matrix. If the semantic relatedness between two concepts c_i and c_j is $SR(c_i, c_j)$, then the (i, j) element in the relatedness matrix is $RM(i, j) = SR(c_i, c_j)$ if $SR(c_i, c_j) > confidence$, and 0 otherwise.

Second, the concept matrix is multiplied by the relatedness matrix (see Figure 2). This matrix multiplication generates a new *relatedness-concept matrix*. Each element of this matrix is

$$r\,\mathrm{var}_{s_i}^{c_j} = \sum_{k=1}^{W} \mathrm{var}_{s_i}^{c_k} \cdot RM(c_k, c_j), \quad (5)$$

where $r\,\mathrm{var}_{s_i}^{c_j}$ is the *relatedness-concept value* of concept c_j in sentence s_i . Equation (5) indicates that the relatedness-concept value of c_j in s_i equals the weighted sum of the values in the original concept vector, and the weighting coefficients are semantic relatedness of concepts.

After these two steps are finished, each sentence s_i is represented with a renewed *relatedness-concept vector*

$$rconceptvector_i = \{r\,\mathrm{var}_{s_i}^{c_1}, r\,\mathrm{var}_{s_i}^{c_2}, \cdots, r\,\mathrm{var}_{s_i}^{c_W}\}. \quad (6)$$

Then the Matrix similarity of two sentences is computed as the cosine similarity of their relatedness-concept vectors.

We give an illustration of the steps in Figure 2. An observation is that the relatedness matrix serves as a bridge to associate sentences with their semantically-related concepts. A sentence consequently obtains weights on the concepts that do not appear explicitly in it.

3.3 Improving QS

We combine linearly the Concept or Matrix similarity with the basic Word similarity and obtain the final sentence similarity. For instance, when considering Concept, the combined similarity is

$$combinesim(s_i, s_j) = wordsim(s_i, s_j) + \alpha \cdot conceptsim(s_i, s_j), \quad (7)$$

where α is a factor to control the balance between Word and Concept similarity. The combined similarity is substituted into either the TFIDF or the Graph model. Using a similar iterative computation, we can get the saliency score for each sentence. For limit of space, we do not give the updating formula (like Equation (1)) here. Note that we replace both the query-sentence and the sentence-sentence similarity in Graph with this combined similarity.

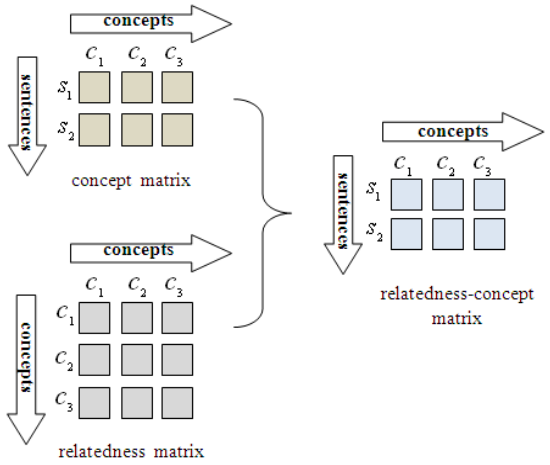

Figure 2. Illustration of the Matrix similarity.

3.4 Redundancy Checking

After the sentences are scored, some of the top sentences may express similar meaning or convey duplicate information. If they are selected simultaneously, the final summary will be redundant. We adopt such a method to address this redundancy: each candidate sentence, before being added to the final output, is compared with the sentences that are already contained in the summary. Only the candidates, whose similarity with all the sentences in the summary is below a predefined threshold λ , can be added to the summary.

4. EXPERIMENTS

In this section, we conduct experimental studies to test the effectiveness of the proposed framework. Before going to the details, we first describe the dataset and evaluation metrics.

4.1 Dataset and Performance Metrics

Document Understanding Conference (DUC) has set a series of QS tracks and provided benchmark datasets. We use the DUC2005 dataset which consists of 50 queries. Each query corresponds to a collection of 25-50 relevant documents. The task is to generate a summary of 250 words for each query from the associated document collection. For preprocessing, we partition the documents into individual sentences with the Sentence-Detector function of the OpenNLP [1] package. Moreover, stop words are removed from the vocabulary.

For quantitative evaluation, we use the ROUGE toolkit which has been widely adopted by DUC for automatic summarization evaluation. ROUGE measures summary quality by counting overlapping units such as n-gram, word sequence and word pairs between a generated summary and a set of reference summaries. In our experiments, we run ROUGE-1.5.5[2] with the parameter settings consistent with DUC 2005: -n 4 -l 250 -w 1.2 -m -2 4 -u -r 1000 -f A -p 0.5 -t 0, where "-l 250" indicates the evaluated summaries have the length of 250 words. In the results, we report three of the ROUGE metrics: ROUGE-1, ROUGE-L and ROUGE-SU.

4.2 Performance Evaluation

We take the basic TFIDF and Graph models (with Word similarity solely) as baselines. The parameters are set in the following ways. Both the coefficient d in the Graph model and the threshold λ in redundancy checking are empirically set to

[1] http://opennlp.sourceforge.net/

[2] http://research.microsoft.com/~cyl/download/ROUGE-1.5.5.tgz

0.3. For the controlling factor α , we experiment on a wide range of values and choose the best one in terms of the evaluation metrics. In Graph, when α is fixed to its best value, we continue to tune *confidence* from 0 to 1.0 with 0.1 as the step size. All the iterative algorithms converge when the difference between the scores computed at two successive iterations for any sentences falls below a threshold (10^{-5} in this study).

Table 1. Experimental results.

Models	Similarity	Rouge1	RougeL	RougeSU
TFIDF	Word (Baseline)	0.34974	0.31813	0.11782
	Word + Concept	0.35597	0.32350	0.12190
	Word + Matrix	0.35870	0.32623	0.12076
Graph	Word (Baseline)	0.36648	0.33714	0.12570
	Word + Concept	0.36916	0.33972	0.12664
	Word + Matrix	0.37124	0.34005	0.12780

Table 1 shows the experimental results when different types of similarity are used. We can see that the introduction of the Concept similarity improves QS in each case and on every metric. This proves that sentence wikification is an effective strategy for enhancing the performance of query-oriented summarization. Also, when combining the Word similarity with the Matrix, rather than Concept, similarity, we can obtain even better results. The difference between Concept and Matrix is that in Matrix, we reweight the concept vectors with semantic relatedness. The smoothing effects of semantic relatedness result in sentence similarity which is more consistent with human judgment, and therefore helps to produce more desirable query-oriented summaries.

When the type of sentence similarity is identical, the Graph model consistently outperforms TFIDF. Even the baseline Graph (Word) can perform better than the enhanced TFIDF (Word + Matrix). This observation conforms to previous literatures which show that Graph, considering both query-sentence and sentence-sentence similarity in a unified computation process, has advantages over the simpler TFIDF model.

We also investigate the influence of the relatedness confidence, i.e., *confidence* . Figure 3 shows the ROUGE-1 metric for Word + Matrix when we use the Graph model and *confidence* is set to various values. The best performance is achieved when *confidence* is set to 0.4. This tells us that the concept pairs with small (less-than-0.4) semantic relatedness actually contribute little to the QS task. Meanwhile, when *confidence* exceeds 0.4, we filter the concept pairs too aggressively and lose some valuable information, which causes the performance to decrease.

5. CONCLUSION AND FUTURE WORK

In this paper, we study whether sentence wikification can improve the performance of query-oriented summarization (QS).

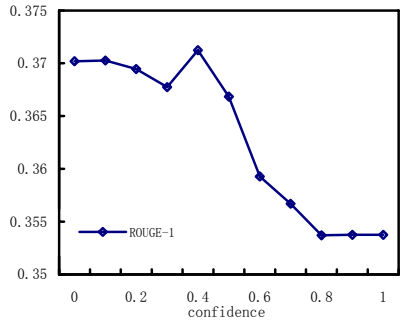

Figure 3. QS performance as *confidence* varies.

In our proposed framework, both queries and sentences in QS are enriched with Wikipedia concepts as additional features. Also, we present the computation of the Matrix similarity, in which semantic relatedness of Wikipedia concepts is considered for smoothing concept matching. Then the combined sentence similarity is employed in the TFIDF or Graph model. From the experiments, we can conclude that sentence wikification improves QS effectively. In addition, the incorporation of semantic relatedness enables us to get better results.

6. ACKNOWLEDGMENTS

This work was supported by National Natural Science Funding of China under Grant No. 90718022 and National 863 Project under Grant No. 2009AA01Z410.

7. REFERENCES

[1] Y. Chali, and S. R. Joty. Improving the Performance of the Random Walk Model for Answering Complex Questions. In *Proceedings of ACL-HLT'08*, pages 9-12, 2008.

[2] Y. Chali, and S. R. Joty. Exploiting Syntactic and Shallow Semantic Kernels to Improve Random Walks for Complex Question Answering. In *Proceedings of ICTAI'08*, pages 123-130, 2008.

[3] G. Erkan, D. R. Radev. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. In *Journal of Artificial Intelligence Research, Vol. 22, 2004.*

[4] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting Wikipedia as External Knowledge for Document Clustering. In *Proceedings of SIGKDD'09, pag*es 389-396, 2009.

[5] F. Li, Y. Tang, M. Huang, and X. Zhu. Answering Opinion Questions with Random Walks on Graphs. In *Proceedings of ACL'09*, pages 733-745, 2009.

[6] D. Milne. Computing Semantic Relatedness using Wikipedia Link Structure. In *Proceedings of the 5th New Zealand Computer Science Research Student Conference*, 2007.

[7] D. Milne, and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08)*, 2008.

[8] J. Otterbacher, G. Erkan, and D. R. Radev. Using Random Walks for Question-focused Sentence Retrieval. In *Proceedings of HLT-EMNLP'05*, pages 915–922, 2005.

[9] J. Tang, L. Yao, and D. Chen. Multi-topic based Query-oriented Summarization. In *Proceedings of SDM'09*, pages 1148-1159, 2009.