WikiSummarizer - A Wikipedia-based Summarization System

Yajie Miao, Chunping Li
School of Software, Tsinghua University

Abstract

This paper presents our WikiSummarizer system at the guided summarization track of TAC 2010. This system is derived from a novel framework which improves summarization through sentence wikification, i.e., enriching sentence representation with concepts from Wikipedia. By examining sentences in the feature space of Wikipedia concepts, we are able to obtain sentence similarity which is more consistent with human judgment. Furthermore, we develop a strategy which incorporates semantic relatedness of Wikipedia concepts into sentence wikification as a smoothing factor. Our system outperforms the baseline and has achieved competitive results in TAC 2010.

1. Introduction

In TAC 2010, the guided summarization task requires participants to generate a 100-word summary for each topic from a given text collection. There are five pre-defined categories that the topics fall into. An additional component of guided summarization involves "update" summarization. In the "update" setting, the user is assumed to have already read the earlier articles. Then, the summarization systems need to generate a 100-word "update" summary from a subsequent 10 news articles.

The performance of summarization relies highly on measurement of sentence similarity. Wikipedia is a comprehensive and well-organized knowledge repository. We propose to map individual sentences into concept vectors, which is referred to as sentence wikification. Then, the derived concept similarity is used to compliment the original lexical similarity. This results in more accurate sentence similarity and thereby enhances summarization.

The rest of the paper is organized as follows. Section 2 introduces a baseline graph-based summarization method. Section 3 presents our new summarization framework. In Section 4, we deal with the "update" setting. In Section 5, we show and discuss the experimental results. Finally, we give conclusion and future work in Section 6.

2. Traditional Graph-based Summarization

Our baseline system is the graph-based summarization method. We denote the given topic as t and the set of sentences as S. A summarization system firstly ranks the sentences in S with respect to t. Then, top sentences are selected until the required summary length (e.g., 100 words) is reached.

The graph-based method has been adopted widely in sentence ranking [1, 2, 7] and summary generating. Specifically, a graph is constructed in which each node represents a sentence. Each edge measures the similarity between the corresponding pair of sentences. Then a sentence s_i is selected into the summary not only because it is relevant (similar) to the topic t but also because s_i is similar to other sentences with high topic-sentence similarity. This idea is captured by Equation (1), where we denote the similarity between sentence s_i and the topic as $sim(s_i, \mathrm{t})$ and the similarity between sentence s_i and s_j as $sim(s_i, s_j)$. Following the Random Walk [8] paradigm, the score for sentence s_i can be calculated iteratively as follows.

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

where $Score^{(n)}(s_i)$ is the score of s_i in n^{th} iteration, and d is a combination coefficient for trading off the two parts. When the iteration converges, the sentences are ranked according to their scores and top sentences are selected.

3. WikiSummarizer

Figure 1 presents the overall structure of our summarization method with sentence wikification.

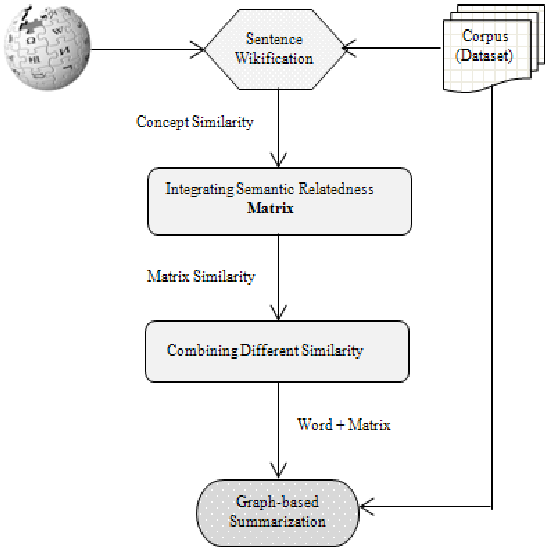

Figure 1. The overall structure of WikiSummarizer.

3.1 Sentence Wikification

In the traditional *Bag of Word* (BOW) approach, each sentence is mapped to a *word vector*, whose elements are usually TF*IDF value of words from the vocabulary. Then the similarity between two sentences s_i and s_j is measured by the cosine value of their word vectors. This *Word* similarity is denoted as $word-sim(s_i, s_j)$.

In general, sentence wikification refers to the practice of representing a sentence with a set of Wikipedia concepts. Though there are more sophisticated strategies for achieving this, we take the *exact-match* strategy introduced in [4] as our wikification method. Its procedures are as follows. To wikify a sentence s_i, we traverse the whole set[1] of Wikipedia concepts and select out the ones that appear explicitly in s_i. In order for high-quality Wikipedia concepts, we also adopt two extra operations. First, we merge some partial concepts into an entire one. 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, i.e., "Kyoto Protocol". Second, we exclude meaningless concepts which result from our exact-match method. For example, the concepts "Position" and "Proto", though contained by the sentence, obviously cannot act as interpretation of the sentence, and thus should be eliminated.

With the retrieved concepts, each sentence is represented with a *concept vector*. Formally, the sentence s_i is associated with a concept vector:

$$conceptvector_i = \{\mathrm{var}_{s_i}^{c_1}, \mathrm{var}_{s_i}^{c_2}, \cdots, \mathrm{var}_{s_i}^{c_W}\},$$

where $\mathrm{var}_{s_i}^{c_j}$ 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 the sentence collection S.

3.2 Smoothing Concept Matching with Semantic Relatedness

We can compute the similarity of concept vectors via their cosine value. However, this potentially brings problems. For instances, two concept vectors, {Kyoto protocol, Emissions trading, Carbon dioxide} and {Global warming, Greenhouse gas, Fossil fuel}, have 0 similarity, though they are quite close according to human judgment. So we also use semantic relatedness of Wikipedia concepts to smooth the matching of concept vectors. Semantic relatedness is a numeric value (between 0 and 1) which indicates the extent to which Wikipedia concepts are semantically close to each other. For instance, "Kyoto protocol" and "Global warming" have the semantic relatedness of 0.7, whereas the semantic relatedness between "Kyoto protocol" and "Financial crisis" is around 0.4.

After sentence wikification, the set of sentences can be represented with a concept matrix. First, with the semantic relatedness values, we build a relatedness matrix. The elements of this matrix are semantic relatedness among the corresponding pair of concepts. We also set a threshold of 0.7 on the semantic relatedness values. Only two concepts whose semantic relatedness exceeds 0.4 can have a value in the relatedness matrix.

Second, the concept matrix is multiplied by the relatedness matrix (see Figure 2), which generates a new relatedness-concept matrix. In this new matrix, we use $r\,\mathrm{var}_{s_i}^{c_j}$ to denote the relatedness-concept value of concept c_j in sentence s_i. Then, rather than a concept vector, each sentence s_i is represented with a relatedness-concept vector :

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

[1] This set can be downloaded at http://download.wikipedia.org/.

Then the sentence similarity between two sentences is computed as the cosine value of their relatedness-concept vectors. Since it stems from wikification, this similarity is called Wiki similarity, i.e., $wiki - sim(s_i, s_j)$.

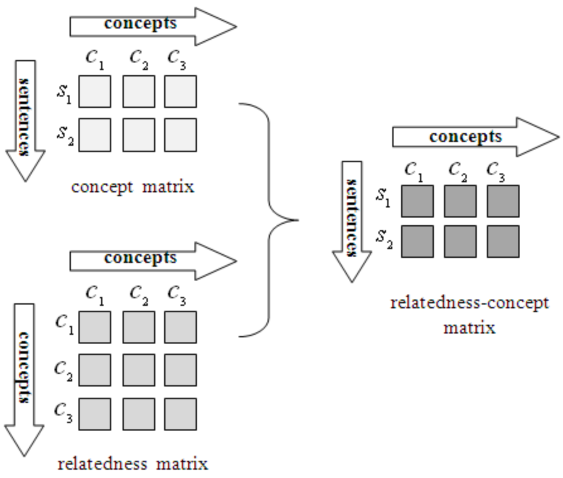

Figure 2. Incorporation of semantic relatedness.

We give an example in Figure 3 to show how Wiki similarity works. Without incorporation of semantic relatedness, the similarity between s_1 and s_2 equals 0, even though they apparently have close semantic meanings. However, their Wiki similarity approximately equals 0.83, which is a more reasonable result.

s_1 : {Kyoto protocol, Emissions trading, Carbon dioxide}

s_2 : {Global warming, Greenhouse gas, Fossil fuel}

C : {Kyoto protocol, Emissions trading, Carbon dioxide, Global warming, Greenhouse gas, Fossil fuel}

	c_1	c_2	c_3	c_4	c_5	c_6
s_1	1	1	1	0	0	0
s_2	0	0	0	1	1	1

concept matrix

	c_1	c_2	c_3	c_4	c_5	c_6
s_1	1.79	1.79	1.0	0.71	2.27	0.7
s_2	2.19	0.75	0.74	2.49	2.57	2.50

relatedness-concept matrix

	c_1	c_2	c_3	c_4	c_5	c_6
c_1	1.0	0.79	0	0.71	0.78	0.70
c_2	0.79	1.0	0	0	0.75	0
c_3	0	0	1.0	0	0.74	0
c_4	0.71	0	0	1.0	0.78	0.71
c_5	0.78	0.75	0.74	0.78	1.0	0.79
c_6	0.70	0	0	0.71	0.79	1.0

relatedness matrix

Figure 3. An example for Wiki similarity.

A key issue for Wiki similarity is computation of semantic relatedness. In our implementation, we adopted Wikipedia Link-based Measure (WLM) [5, 6], a sophisticated link-based metric, to calculate the semantic relatedness values between Wikipedia concepts. An important feature of Wikipedia is the hyperlinks among Wikipedia pages (articles). These human-generated links represent mutual endorsement among Wikipedia concepts. 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. The web demo of WLM can be found at http://wdm.cs.waikato. ac.nz:8080/-service?task=compare. By exploring link structures in Wikipedia, WLM achieves comparable accuracy with the well-known ESA [3] method, while improving the efficiency of ESA significantly.

3.3 Combined Similarity and Summarization

We obtain the final sentence similarity by combining Wiki and Word similarity linearly.

$$final - sim(s_i, s_j) = word - sim(s_i, s_j) + \alpha \cdot wiki - sim(s_i, s_j)$$

where α is a factor to control the balance between Word and Wiki similarity. The topic-sentence and sentence-sentence similarity values in the Graph method are replaced with their corresponding final similarity, rather than the original Word similarity. Using a similar iterative computation like Equation (1), we can get the score for each sentence. The top sentences are selected to form the desired summary.

3.4 Redundancy Checking

For removing duplicate information in the top sentences, we perform redundancy checking based on comparison of sentence similarity: 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. We empirically set λ to 0.3.

4. Generating Update Summaries

Our summarization steps for the update task is similar to those for the traditional one. Specifically, for an update topic, we wikify the sentences in its corresponding collection. Then, these sentences are ranked based on the graph-based method in Section 2, with the renewed sentence similarity coming from wikification. Finally, the top ranked sentences are selected to generate the summary.

However, when removing redundancy, we consider the ***prior summary*** generated in the traditional task for q. Since the user is assumed to have read the earlier articles, the sentences contained in ***update summary*** should not be too similar to those in prior summary. Therefore, only the candidates, whose similarity with *both the sentences in the current update summary and those in the formed prior summary* is below the predefined threshold λ, can be added to the summary. In this way, we are likely to avoid including redundant information which has been stated in earlier articles.

5. Experimental Results

To verify the effectiveness of our system, we firstly compare it with the top-performance systems of DUC 2005, solely on the traditional summarization task. Then, we report its performance in TAC 2010.

5.1 Comparison on DUC 2005

Table 1 shows the ROUGE results (ROUGE-1 and ROUGE-SU) for the top three systems on DUC 2005. Together, we present the results for the baseline (i.e., the Graph method) and our system. We can see that our system ranks 3rd place among all the submitted runs, which proves that WikiSummarizer has competitive results. Also, WikiSummarizer outperforms the baseline, and this confirms the effectiveness of wikification.

Table 1. Results on DUC 2005.

Method	Rouge1	RougeSU
System15	0.37469	0.13133
System4	0.37436	0.12746
System17	0.36900	0.12933
Graph (Baseline)	0.36648	0.12570
WikiSummarizer	**0.37124**	**0.12780**

5.2 Results in TAC 2010

Finally our new method is evaluated in TAC 2010 guided summarization track. The combined results on both traditional and update summarization are shown in Table 2. Our WikiSummarizer ranks 7th place among 43 submitted runs. Therefore, WikiSummarizer has achieved competitive performance on the challenging tasks of TAC 2010. Since there are still unsettled problems (e.g., the setting of parameters) in this new method, we confidently believe that WikiSummarizer has the potential to gain further improvements.

Table 2. Results in TAC 2010.

Method	Evaluation Score
System13	0.04417
System8	0.04350
System4	0.04115
WikiSummarizer	**0.03939**

6. Conclusion and Future Work

In this paper, we report our system in TAC 2010. We enrich each topic statement and sentence with Wikipedia concepts as additional features. Also, we take semantic relatedness of Wikipedia concepts into consideration. Finally, the combined sentence similarity is employed in the guided summarization track. From the experiments, we can conclude that our new framework improves the baseline significantly, and achieves competitive results in DUC 2005 and TAC 2010. Therefore, sentence wikification performs effectively for enhancing summarization.

For future work, we will examine other forms of information in Wikipedia for creating more comprehensive representation of sentences. Also, we will consider adapting our framework to other applications in text mining such as clustering and question answering.

References

[1] Y. Chali, and S. R. Joty. Exploiting Syntactic and Shallow Semantic Kernels to Improve Random Walks for Complex Question Answering. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'08)*, pages 123-130, 2008.

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

[3] E. Gabrilovich, and S. Markovitch. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1606-1611, 2007.

[4] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting Wikipedia as External Knowledge for Document Clustering. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD'09)*, pages 389-396, 2009.

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

[6] 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.

[7] J. Otterbacher, G. Erkan, and D. R. Radev. Using Random Walks for Question-focused Sentence Retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP'05)*, pages 915–922, 2005.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *Unpublished draft*, 1998.