TimedTextRank: Adding the Temporal Dimension to Multi-Document Summarization

Xiaojun Wan
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
wanxiaojun@icst.pku.edu.cn

ABSTRACT
Graph-ranking based algorithms (e.g. TextRank) have been proposed for multi-document summarization in recent years. However, these algorithms miss an important dimension, the temporal dimension, for summarizing evolving topics. For an evolving topic, recent documents are usually more important than earlier documents because recent documents contain much more novel information than earlier documents and a novelty-oriented summary should be more appropriate to reflect the changing topic. We propose the TimedTextRank algorithm to make use of the temporal information of documents based on the graph-ranking based algorithm. A preliminary study is performed to demonstrate the effectiveness of the proposed TimedTextRank algorithm for dynamic multi-document summarization.

Categories and Subject Descriptors:
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *abstracting methods*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*

General Terms: Algorithms, Design, Experimentation

Keywords: Multi-Document Summarization, Dynamic Summarization, TimedTextRank, Temporal Dimension

1. INTRODUCTION
Automated multi-document summarization has drawn much attention in recent years. Multi-document summary is usually used to provide concise topic description about a cluster of documents and facilitate users to browse the document cluster. Recently, graph-ranking based algorithms have been proposed to rank sentences or passages. LexPageRank [3] is an approach for computing sentence importance based on the concept of eigenvector centrality. Mihalcea and Tarau [5, 6] also propose the TextRank algorithm based on PageRank and HITS to compute sentence importance for document summarization.

All the above algorithms have shown good effectiveness for static multi-document summarization, i.e. the document set to be summarized does not change over time. However, in some real applications, the document set is dynamically changing over time. For example, in a practical topic detection system (e.g. *Google News*[4], *Baidu News* [1], etc.), news documents are fed into the system in real time and news topics are detected by clustering documents. The documents within a topic are changing dynamically, in other words, the news topic is evolving over time. New documents are continuously added into the topic during the whole lifecycle of the topic and they bring new information about the topic. Summarization for an evolving topic differs from summarization for a static topic in that the latter aims to reflect the important information of the topic, and the summary should put equal emphasis on both new documents and old documents in the topic, while the former aims to reflect the both important and novel information of the topic, and the summary should put more emphasis on new documents.

In this study, we propose the TimedTextRank algorithm to put more emphasis on new information in the dynamic document set (topic). The algorithm extends the previous graph-ranking based algorithm by adding the temporal dimension, similar to the work proposed for publication search [7]. The temporal information of documents is taken into account in the algorithm and the sentences in new documents are valued more highly than the sentences in old documents. A preliminary evaluation is performed and the results demonstrate the effectiveness of the proposed TimedTextRank for dynamic multi-document summarization.

2. THE PROPOSED TIMEDTEXTRANK
Before describing the TimedTextRank algorithm, we first introduce the original TextRank algorithm. The TextRank algorithm makes use of the relationships between sentences and selects sentences according to the "votes" or "recommendations" from their neighboring sentences, which is similar to PageRank and HITS. It first builds an affinity graph to reflect the relationships among all sentences in the document set, and then computes the informativeness score of each sentence based on the affinity graph. The informativeness of a sentence indicates how much information about the main topic the sentence contains. The sentences with the highest informativeness scores are chosen into the summary. In order to keep the redundancy in the summary as minimal as possible, a post-processing step is often used to remove redundant information in the highly informative sentences.

In more detail, given a sentence collection $S=\{s_i \mid 1\le i\le n\}$ of the document set for a specified topic at a specified time, the affinity weight $sim(s_i,s_j)$ between any sentence pair of s_i and s_j is calculated using the standard Cosine measure. The weight associated with term t is calculated with the $tf_t.isf_t$ formula, where tf_t is the frequency of term t in the sentence and isf_t is the inverse sentence frequency of term t, i.e. $1+log(N/n_t)$, where N is the total number of sentences and n_t is the number of sentences containing term t. If sentences are considered as nodes, the sentence collection can be modeled as an undirected graph by generating a link between two sentences if their affinity weight exceeds 0, i.e. an undirected link between s_i and s_j ($i\neq j$) with affinity weight $sim(s_i,s_j)$ is constructed if $sim(s_i,s_j)>0$; otherwise no link is constructed. Thus, we construct an undirected graph G reflecting the relationships between sentences by their content similarity. We use an adjacency (affinity) matrix **M** to describe G with each entry corresponding to the weight of a link in the graph, i.e. **M** =

Copyright is held by the author/owner(s).
SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

$(M_{i,j})_{n\times n}$ is defined by $M_{i,j} = sim(s_i,s_j)$ for $i \neq j$. Then $\mathbf{M}$ is normalized to $\widetilde{\mathbf{M}}$ to make the sum of each row equal to 1.

Based on the affinity graph G, the informativeness score $IFScore(s_i)$ for sentence s_i can be deduced from those of all other sentences linked with it and it can be formulated in a recursive form as follows:

$$IFScore(s_i) = d \cdot \sum_{all\ j \neq i} IFScore(s_j) \cdot \widetilde{M}_{j,i} + \frac{(1-d)}{n} \quad (1)$$

where d is the damping factor usually set to 0.85.

We now describe the TimedTextRank algorithm. Since we aim to take into account the temporal dimension of each document and put more emphasis on the sentences in new documents, we modify the TextRank algorithm by re-weighting each "vote" when evaluating the importance of a specified sentence. The votes cast from the sentences in new documents are attached more importance than the votes cast from the sentences in old documents. The algorithm calculates the time-weighted informativeness score for each sentence as follows:

$$IFScore^T(s_i) = d \cdot \sum_{all\ j \neq i} w_j \cdot IFScore^T(s_j) \cdot \widetilde{M}_{j,i} + \frac{(1-d)}{n} \quad (2)$$

Equation (2) is a modified version of Equation (1). In this equation, w_j is the time based weight for each "vote". Its value depends on the publication time of the document containing sentence s_j. The earlier the document is published, the smaller the weight w_j is. Since exponential average is extensively used in time-series prediction, we choose to decay the weights exponentially according to time [7]:

$$w_j = DecayRate^{(y-t_j)/24} \quad (3)$$

where y is the current time, t_j is the publication time of the document containing sentence s_j and $(y - t_j)$ is the time gap in hours. *DecayRate* is a parameter and we simply use 0.5 in the experiments to illustrate the algorithm. Note that if *DecayRate* is 1, the TimedTextRank algorithm will be the same as the original TextRank algorithm. The *DecayRate* parameter could be tuned according to the nature of a dataset or the user.

Note that the post-processing step of redundancy removing is the same for both the TextRank method and the TimedTextRank method, whose details are omitted due to page limit.

3. PRELIMINARY EVALUATION

Though static multi-document summarization has been extensively evaluated by DUC [2], there is no ground dataset for evaluating dynamic multi-document summarization. In this study, we evaluate the proposed TimedTextRank in a real topic detection system using user study.

The system collects news web pages from main Chinese news portals (e.g. *sina.com*, *sohu.com*, *163.com*, etc.) and then detects hot topics from the large amount of web documents. The publication time of each web page is recorded. The algorithm for topic detection extends the incremental single-pass clustering algorithm by adding the steps of topic merging and topic eliminating to keep only the hottest topics and present them to users. The details of the algorithm are omitted due to page limit. The document set for a specified topic is changing dynamically by adding new documents in real time. For each of the top ten topics, the system produces two five-sentence summaries, one by the TimedTextRank algorithm (i.e. Equation (2)) and the other by the TextRank algorithm (i.e. Equation (1)). The two summaries are presented to users for comparison.

Three subjects were involved in the user study. They were required to express an opinion over a 5-point scale for each summary of each topic at every hour from 9:00 am to 5:00 pm, where 1 stood for "*the summary is not at all good*", 3 for "*the summary is somewhat good*" and 5 for "*the summary is extremely good*". We collected the responses of subjects and averaged them across topics, as shown in Table 1.

Table 1. Results of preliminary study

	TextRank	TimedTextRank
9:00 am	3.0	3.2
10:00 am	3.5	3.4
11:00 am	3.1	3.5
12:00 am	2.9	3.2
1:00 pm	2.8	3.4
2:00 pm	3.1	3.0
3:00 pm	3.4	3.8
4:00 pm	3.4	3.8
5:00 pm	3.5	4.1
Average	3.19	3.49

Seen from the above table, the TimedTextRank algorithm significantly outperforms the original TextRank algorithm. The proposed algorithm can produce better summaries for evolving topics. The results also validate that users really have more interest in new information in the document set.

The preliminary study demonstrates the effectiveness of the proposed TimedTextRank for dynamic multi-document summarization. We will perform more thorough user study to make the conclusion more convincing in future work. We will improve the efficiency of the algorithm in future work by evaluating only fresh sentences instead of re-evaluating all sentences in the documents during each summarization process.

4. REFERENCES

[1] Baidu News: http://news.baidu.com

[2] DUC: http://duc.nist.gov

[3] ErKan, Günes, Radev, D. R.: LexPageRank: Prestige in Multi-Document Text Summarization. In Proceedings of EMNLP2004.

[4] Google News: http://news.google.com

[5] Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In Proceedings of EMNLP2004.

[6] Mihalcea, R. and Tarau, P.: A language independent algorithm for single and multiple document summarization. In Proceedings of IJCNLP2005.

[7] Yu, P. S., Li, X. and Liu, B.: Adding the temporal dimension to search – a case study in publication search. In Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05).