

# On-line Summarization of Time-series Documents using a Graph-based Algorithm

**Satoko Suzuki**

Graduate School of Humanities and  
Sciences, Ochanomizu University  
2-1-1 Otsuka Bunkyo-ku Tokyo, Japan  
suzuki.satoko@is.ocha.ac.jp

**Ichiro Kobayashi**

Graduate School of Humanities and  
Sciences, Ochanomizu University  
2-1-1 Otsuka Bunkyo-ku Tokyo, Japan  
koba@is.ocha.ac.jp

## Abstract

As enormous amount of electronic documents on the Web have been increasing, the necessity of automatic summarization has also been increasing to help people grasp the essential points of the documents. Many summarization techniques dealing with single document and multi-documents have been studied. However, due to the increase of the documents which report the change of topics along a timeline, called time-series documents, in recent years, a summarization technique which generates a summary of time-series documents, called timeline summarization, has been actively studied as an area of automatic summarization. There are different difficulties in summarizing time-series documents from other type of automatic summarization. The basic approach for timeline summarization is to extract sentences which describe major events in object documents in chronological order to generate a timeline summary.

However, unlike the prior studies of timeline summarization, we particularly focus on on-line summarization of time-series documents and propose an on-line graph-based timeline summarization method. With our proposed method, a summary of time-series documents can be generated at any point of time when it is required. We conduct experiments to investigate the ability of our proposed method, evaluate the results with ROUGE metrics, and show our proposed method produces a better summary compared to other representative summarization methods.

## 1 Introduction

The automatic summarization techniques have been required due to increasing the amount of electronic documents. The object documents handled by automatic summarization are diverse from newspaper and academic articles to the documents used in social network services such as Weblog, Twitter, etc. Depending on object documents, an appropriate summarization technique is applied. Due to the increase of electronic documents updated day by day, in recent years, new techniques which summarize time-series documents, called timeline summarization, have been actively studied. There are different difficulties in summarizing time-series documents from other type of automatic summarization, because as for timeline summarization we have to summarize current information taking account of the information from the past. Besides, we have to decide the important information which should be included in a summary, tracking the change of topics. The basic approach for timeline summarization is to extract sentences which describe major events in object documents in chronological order to generate a timeline summary.

In this study we particularly focus on the development of an on-line method to summarize time-series documents. To achieve this, we have to deal with various problems, for example, how to combine the current information with the information from the past in order to recompile object information to be summarized, how to track topics, how to rank important information, etc. To deal with these problems, we employ graph structure to represent sentence re-

lation and apply graph-based algorithm to extract important information from the graph — here, the graph is evolved along a timeline by combining the current and past information, and then it represents object information to be summarized. In our proposed method, a summary can be generated at any point of time by request.

The structure of this paper is as follows. Related studies are summarized in section 2. In section 3, we show our proposed method. In section 4, we explain the experiments based on our proposed method and discuss the results. Finally, we conclude this study in section 5.

## 2 Related Studies

As an application of automatic summarization, summarization of time-series document, called timeline summarization, has recently been actively studied. At an early stage of the development of the technique, Allan et al. (2001) have proposed several summarization methods for time-series documents and built evaluation corpus for the method. Chieu et al. (2004) have proposed a framework for making a timeline of events occurrence. They extract events, which correspond to important sentences, relevant to a query from documents and place such events along a timeline. Yan et al. (2011a) have also proposed a method to extract important sentences to make a timeline summary expanding the graph-based sentence ranking algorithm used for multi-document summarization, and proposed a summarization method, called Evolutionary Trans-Temporal Summarization (ETTS), which extracts sentences from different points of time in a particular period, and they have also proposed a method to optimize the function for the combination of important factors such as relevance, coverage, coherence and variety of words in a generated summary (Yan et al., 2011b). Tran et al. (2013a) have employed support vector machine based ranking algorithm to rank sentences with 28 features and selected sentences with high ranking score for a timeline summary. They have reported that their method outperforms other representative timeline summarization methods, e.g., (Yan et al., 2011a) and the reason for that is because they leverage some latent factors under supervision of human timelines.

As the studies using graph representation for the relation among sentences, Erkan et al. (2004) have introduced the PageRank algorithm to rank sentences in the order of high centrality and then extracted the sentences with high ranking score as important sentences which are expected to be included in a summary. Yan et al. (2012) have introduced hierarchical graph structure to represent both textual and semantic relation among sentences. Moreover, Li et al. (2013) have proposed a method called Evolutionary Hierarchical Dirichlet Process (EHDP) to consider the development of topics along a timeline. In their method, they have introduced a non-parametric Bayesian topic model to represent latent information among sentences — in their method, coverage and coherence are mainly considered for extracting sentences for a summary.

As the studies to focus on the topic development along a timeline, Zhao et al. (2013) have focused on the attention attracted to topics of interest — they call it social attention — to make a timeline summary which reflects user’s interest. Hu et al. (2014) have explored the interactions of storylines in a news topic. They have especially focused on the coherence between news articles and discovered storyline interactions for timeline summarization.

Unlike the prior studies mentioned above, we focus on on-line summarization for time-series documents. The information reported in time-series documents such as news paper articles is updated day by day, and we often need a summary for the information which has so far been reported. Therefore, in this study we aim to propose an on-line summarization of time-series documents with a graph-based algorithm. By our method, a summary can be generated at any point of time when it is required.

## 3 On-line graph-based timeline summarization

Figure 1 illustrates an overview of our proposed method.

The basic framework for our proposed method is that a summary can be generated at any point of time when it is required, with the sentences which are both passed over from the past articles and the articles of that day. In each day, a graph representing the relation among sentences is constructed and

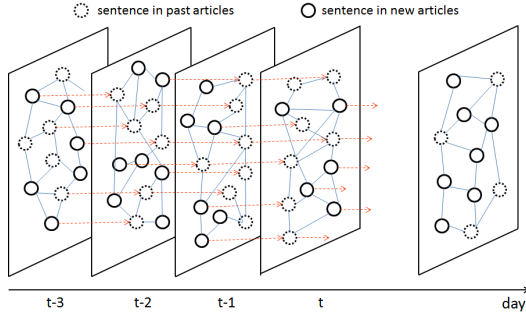


Figure 1: Overview of our proposed method

LexRank algorithm (Erkan et al., 2004) is applied to the graph to rank the sentences, which correspond to the nodes in the graph, based on the centrality of the sentences in the graph. Based on the ranking score, a particular number of top sentences are selected and recompiled as a new object documents to be summarized, and then a summary is generated with the constraint on summary length. The summary candidate sentences are passed over to the next day as the sentences from past articles, and then a new graph consisting of both sentences from past and that day is reconstructed and the same procedure is applied to the updated graph. Like this, a series of the procedures is repeatedly applied to the summary candidate sentences in each day, and a summary of time-series documents is generated at any point of time when it is required.

The algorithm of our proposed method is shown in Algorithm 1.

---

**Algorithm 1** ranking algorithm

---

- 1: **Input:**  $D_t, \epsilon, l$
  - 2:  $S = \{ \}$
  - 3:  $\epsilon \leftarrow$  threshold
  - 4: **for**  $t = 0$  to  $T$  **do**
  - 5:    $S' \leftarrow S + D_t$
  - 6:   ranking  $S'$  with LexRank
  - 7:   **if** length of  $S' > \epsilon$  **then**
  - 8:      $S \leftarrow$  top  $\epsilon$  sentences of  $S'$
  - 9:   **else**
  - 10:     $S \leftarrow S'$
  - 11:   **end if**
  - 12: **end for**
  - 13: **return** top  $l$  sentences of  $S$
- 

In Algorithm 1,  $D_t, \epsilon,$  and  $l$  are provided as input values. Here,  $D_t$  is a set of documents provided at time  $t = \{0, \dots, T\}$ ,  $\epsilon$  is the threshold of graph size, and  $l$  is the number of sentences included in a summary.  $S$  is a set of sentence candidates which are expected to be included in a generated summary. As mentioned above, as for the ranking algorithm for sentences, we employ LexRank algorithm proposed in (Erkan et al., 2004) – the detail procedure is shown in Algorithm 2.

Here, the similarity between two sentence vectors is defined as in equation (1).

$$\begin{aligned} & \text{idf-modified-cosine}(x, y) \\ &= \frac{\sum_{w \in x, y} tf_{w, x} tf_{w, y} (\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (tf_{x_i, x} \text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (tf_{y_i, y} \text{idf}_{y_i})^2}} \end{aligned} \quad (1)$$

In the above equation,  $tf_{w, s}$  indicates the frequency of word  $w$  in sentence  $s$ .  $x$  and  $y$  indicate sentences and  $x_i$  and  $y_i$  indicate words in  $x$  and  $y$ , respectively.  $\text{idf}_i$  is defined equation (2).

$$\text{idf}_i = \log \frac{N_d}{n_i} \quad (2)$$

$N_d$  is the total number of the documents in  $S'$ , and  $n_i$  is the number of documents in which word  $i$  occurs.

The score of node  $u$  is calculated based on equation (3).

$$p(u) = \frac{d}{N} + (1 - d) \sum_{v \in \text{adj}[u]} \frac{p(v)}{\text{deg}(v)} \quad (3)$$

Here,  $N$  indicates the number of nodes in a graph,  $\text{adj}[u]$  is a set of nodes adjoining sentence  $u$ .  $d$  is the damping factor to estimate the similarity between noncontiguous nodes with a particular rate. The generated graph shall be an unweighted graph whose edges are pruned by threshold  $t$ . Correspondingly, as a summarization method using a weighted graph, Continuous LexRank (Cont.LexRank) has been proposed (Erkan et al., 2004). In the method, edges are not pruned by threshold, but the similarity between the objective node and other nodes are accounted when calculating the score of the node. Therefore,

**Algorithm 2** LexRank

---

```

1: Input: An array S of  $n$  sentences, cosine threshold  $t$  output: An array L of LexRank scores
2: Array CosineMatrix[n][n];
3: Array Degree[n];
4: Array L[n];
5: for  $i \leftarrow 1$  to  $n$  do
6:   for  $j \leftarrow 1$  to  $n$  do
7:     CosineMatrix[i][j] = idf-modified-cosine(S[i],S[j]);
8:     if CosineMatrix[i][j] >  $t$  then
9:       CosineMatrix[i][j] = 1;
10:      Degree[i]++;
11:    end
12:  else
13:    CosineMatrix[i][j] = 0;
14:  end
15: end
16: end
17: for  $i \leftarrow 1$  to  $n$  do
18:   for  $j \leftarrow 1$  to  $n$  do
19:     CosineMatrix[i][j] = CosineMatrix[i][j]/Degree[i];
20:   end
21: end
22: L = PowerMthod(CosineMatrix, $n,\epsilon$ );
23: return L;

```

---

equation (3) which is used for calculating the score of a node is enhanced as shown in equation (4).

$$\begin{aligned}
p(u) &= \frac{d}{N} + (1-d) \\
&\times \sum_{v \in \text{adj}[u]} \frac{\text{idf-modified-cosine}(u,v)}{\sum_{z \in \text{adj}[v]} \text{idf-modified-cosine}(z,v)} p(v)
\end{aligned} \tag{4}$$

In our proposed method, we limit the size of a graph. If the size of a graph, i.e., the number of sentences, exceeds the predefined threshold, it will be reduced to the size of the threshold. Then the sentences represented in the graph are ranked by LexRank algorithm and are extracted according to the ranking score for a summary. In generating a summary, we use MMR-MD (Maximal Marginal Relevance-Multi Documents) proposed in (Goldstein et al., 2000) to avoid redundancy in a summary. This index works to avoid extracting similar sentences in a summary by providing penalty corresponding to the similarity between a newly extracted

sentence and the already extracted sentences. It is often used for query-based summarization. In our method, it is required to extract the sentences which have high ranking score and are not similar to the already extracted sentences as a part of a summary. Therefore, we modify MMR-MD as shown in equation (5). We call our modified MMR-MD MMR' hereafter.

$$\begin{aligned}
\text{MMR}' &\equiv \arg \max_{s_i \in S \setminus S'} [\lambda \text{score}(s_i) \\
&\quad - (1 - \lambda) \max_{s_j \in S'} \text{sim}(s_i, s_j) \times \eta]
\end{aligned} \tag{5}$$

- $S$  : A set of summary candidate sentences
- $S'$  : A set of already extracted sentences as summary  $S$
- $s_i$  : A sentence in  $S \setminus S'$
- $\lambda$  : Weighting parameter
- $\eta$  : Adjustment coefficient

As for the calculation of similarity between sentences, we use cosine similarity. In order to adjust

the weighting balance between ranking score, i.e.,  $score(s_i)$ , and penalty score, i.e.,  $sim(s_i, s_j)$ , we have introduced  $\lambda$  as an weighting parameter, besides introduced  $\eta$  as an adjustment coefficient for balancing exponential order between the two terms in the equation.

## 4 Experiment

### 4.1 Data

As the data for experiments, we use the data set for timeline summarization used in (Tran et al., 2013a; Tran et al., 2013b). This data set consists of the newspaper articles about 9 topics and is collected from multiple news resources. The referential summary manually generated by humans is prepared for each topic. Table 1 shows the detail about the data set used in the experiments.

Table 1: Data set used in the experiments

Topic	News sources	Num. of documents	Total num. of sentences
H1N1	Guardian	76	2630
H1N1	Reuters	207	4769

H1N1 is the topic about influenza. Guardian and Reuters are the names of news agency.

### 4.2 Evaluation metrics

The proposed methods are evaluated based on the comparison between the referential summary and a generated summary. We employ ROUGE (Lin et al., 2004) — in particular, we employ ROUGE-1, ROUGE-2, and ROUGE-L, as metrics to evaluate our method. Here, ROUGE-1 and ROUGE-2 are metrics based on unigram and bigram matching between the referential summary and a generated summary, respectively. ROUGE-L is metrics based on the longest common subsequence part between a referential summary and a generated summary. We calculate recall, precision, and F-score in each metrics.

Equation (6), (7), and (8) show the recall, precision, and F-score of N-gram in ROUGE metrics.

$$\text{ROUGE-N-R} = \frac{\sum_{S \in RS} \sum_{N\text{-gram} \in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S \in RS} \sum_{N\text{-gram} \in S} \text{Count}(\text{N-gram})} \quad (6)$$

$$\text{ROUGE-N-P} = \frac{\sum_{S \in CS} \sum_{N\text{-gram} \in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S \in CS} \sum_{N\text{-gram} \in S} \text{Count}(\text{N-gram})} \quad (7)$$

$$\text{ROUGE-N-F} = \frac{2 \times \text{ROUGE-N-P} \times \text{ROUGE-N-R}}{\text{ROUGE-N-P} + \text{ROUGE-N-R}} \quad (8)$$

Here, R, P, F stand for recall, precision, and F-score, respectively.  $S$  indicates a summary,  $RS$  indicates a referential summary, and  $CS$  indicates a candidate summary.  $\text{Count}(\text{N-gram})$  is the number of N-gram in a summary and  $\text{Count}_{\text{match}}(\text{N-gram})$  is the maximum number of N-grams which are common to both referential summary and generated summary. We evaluated the result in two cases: i.e., with and without stop words by introducing stemming processing.

As the baseline to compare the ability between the proposed method and other methods, we prepare two summaries: one is generated by randomly selecting sentences and the other is generated by Cont.LexRank algorithm. Hereafter, we call Cont.LexRank “LexRank”. In terms of the length of a generated summary, we adopted the same length as that of the referential summary of each topic. Moreover, as pre-processing, stop words, e.g., ‘a’, ‘the’, etc., are removed and stemming processing is adopted for all object documents. We employed Porter’s algorithm (Porter, 1980) for stemming.

### 4.3 Result and discussion

The experiment result of each data evaluated with ROUGE metrics is shown from Table 2 to Table 5. Table 2 and 4 are the results in the case of using stop words when evaluating by ROUGE, and Table 3 and 5 are the results in the case of without stop words. Here, MMR’ was not introduced in the above results. In the tables, R1, R2, and RL stand for ROUGE-1, ROUGE-2, and ROUGE-L, respectively. R1-R indicates the value of recall in ROUGE-1. The results shown in the tables are represented in the form of being rounded off to three decimal place. The best score in each metrics is expressed in bold fonts.

Looking at the results, random gets the lowest score at any evaluation metrics. On the other hand, the proposed method gets close scores or higher

Table 2: H1N1 Guardian

Methods	R1-R	R1-P	R1-F	R2-R	R2-P	R2-F	RL-R	RL-P	RL-F
random	0.386	0.389	0.415	0.067	0.069	0.072	0.360	0.367	0.388
LexRank	<b>0.602</b>	0.389	0.472	<b>0.187</b>	<b>0.121</b>	<b>0.147</b>	<b>0.566</b>	0.365	0.444
our method	0.593	<b>0.403</b>	<b>0.476</b>	0.172	0.116	0.138	0.562	<b>0.383</b>	<b>0.451</b>

R1: ROUGE-1, R2:ROUGE-2, RL: ROUGE-L, -R: Recall, -P: Precision, -F: F-score

Table 3: H1N1 Guardian without stop words

Methods	R1-R	R1-P	R1-F	R2-R	R2-P	R2-F	RL-R	RL-P	RL-F
random	0.197	0.223	0.212	0.022	0.024	0.023	0.197	0.218	0.207
LexRank	<b>0.465</b>	0.305	0.368	0.127	0.083	0.101	<b>0.446</b>	0.292	0.353
our method	0.453	<b>0.311</b>	<b>0.369</b>	<b>0.130</b>	<b>0.089</b>	<b>0.106</b>	0.441	<b>0.303</b>	<b>0.359</b>

Table 4: H1N1 Reuters

Methods	R1-R	R1-P	R1-F	R2-R	R2-P	R2-F	RL-R	RL-P	RL-F
random	0.415	0.253	0.317	0.030	0.020	0.024	0.394	0.237	0.297
LexRank	0.609	0.245	0.349	<b>0.150</b>	0.060	0.086	0.576	0.232	0.330
our method	<b>0.641</b>	<b>0.255</b>	<b>0.358</b>	0.147	<b>0.063</b>	<b>0.088</b>	<b>0.603</b>	<b>0.241</b>	<b>0.34</b>

Table 5: H1N1 Reuters without stop words

Methods	R1-R	R1-P	R1-F	R2-R	R2-P	R2-F	RL-R	RL-P	RL-F
random	0.175	0.120	0.142	0.005	0.004	0.005	0.170	0.116	0.138
LexRank	0.454	0.196	0.273	<b>0.083</b>	<b>0.036</b>	<b>0.050</b>	0.443	0.191	0.267
our method	<b>0.495</b>	<b>0.205</b>	<b>0.284</b>	0.078	<b>0.036</b>	0.049	<b>0.474</b>	<b>0.200</b>	<b>0.274</b>

scores than LexRank. As for the results shown in Table 3, i.e., H1N1 Guardian without stop words, the proposed method gets higher scores than LexRank at most evaluation metrics. However, as for the results shown in Table 4, i.e., H1N1 Reuters, the proposed method gets higher scores at many metrics than LexRank when evaluating the result with stop words. However, both methods produce similar results.

### Introducing MMR'

We conducted an experiment to investigate the influence on the accuracy of summarization results by introducing MMR'. We used the articles of H1N1 of Guardian and set the graph size as 2000. For evaluation, we employed ROUGE-1 whose result is considered as being close to the sense of humans (Lin et al., 2004). The results of the experiment obtained by changing the value of an adjustment coefficient  $\eta$  are shown from Figure 2 to 4.

The figures show the results in the case of with-

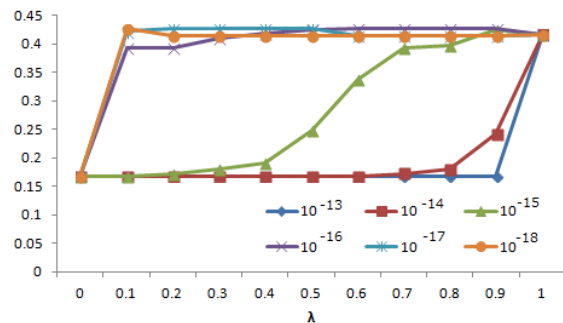


Figure 2: ROUGE-1/Recall(without stop words)

out stop words. The vertical axis shows ROUGE-1 value, and the horizontal axis shows the value of the weighting parameter  $\lambda$ . Each line in the figures show the value of  $\eta$ . Figure 2, 3, and 4 show the changes of recall, precision and F-score by ROUGE-1 evaluation, respectively. Looking at the results, at any metrics, when  $\eta = 10^{-15}$ , the value changes gently, if  $\eta$  is larger than  $10^{-15}$ , the penalty term influences

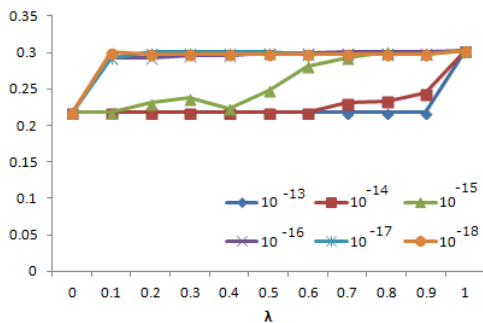


Figure 3: ROUGE-1/Precision(without stop words)

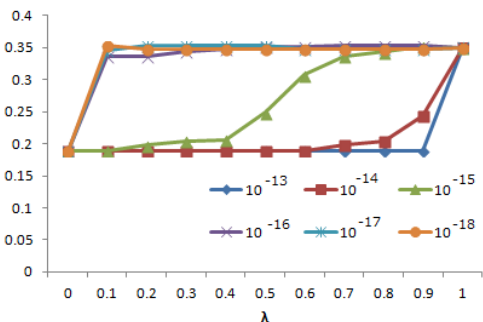


Figure 4: ROUGE-1/F-score(without stopword)

the result, and if  $\eta$  is smaller, the ranking score influences the result. As a result, even though we can find some parts where the accuracy is higher if we put weight on the penalty term rather than the ranking score term, the difference is a little. So, we have not been able to confirm that MMR' works to raise the accuracy in this experiment.

#### 4.4 Supplementary experiment

Each sentence is represented as a sentence vector constructed with the words included in itself and the frequency of those words. When making a sentence or document vector, we usually remove stop words from the vector because the stop word do not represent the contents of the documents. As well as this, the word which do not appear in documents are hardly to represent the contents of the documents. Based on this, we have conducted experiments on removing such words from the object documents to be summarized and investigated the summarization result. As object documents, we used the articles about H1N1 of Guardian from which low frequency words are removed, and we set the graph size 2000

as experimental settings. Figures 5 and 6 show the results, that is, ROUGE-1 values of recall, precision and F-score in the cases of with and without stop words, respectively.

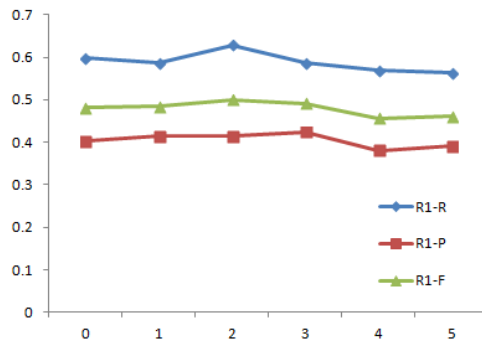


Figure 5: ROUGE without low frequency words (with stop words)

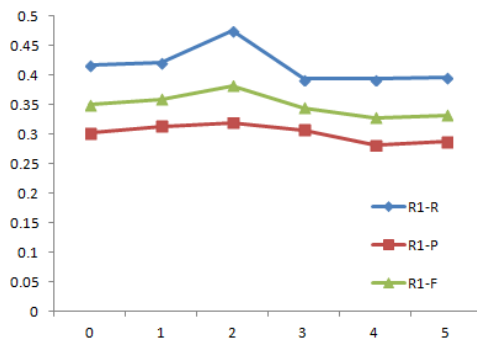


Figure 6: ROUGE without low frequency words (without stop words)

In the figures, the horizontal axis indicates the frequency for the words which are removed from the object documents — for example, 3 indicates the case where the words appearing less than three times in object documents are removed from the documents and a summary is generated with the recompiled documents. The vertical axis indicates evaluation value — for evaluation we employ ROUGE-1 value. Each line in the figure indicates each metrics. We see from the figures that the summary generated in the case of removing the words which appear less than twice or three times in the documents gets highly evaluated. Furthermore, in the case that a summary is generated by removing the words which appear less than twice from objective documents,

the accuracy is better than the case without any pre-processing. Compared with other methods, the result is summarized in Table 6. Here, the case of removing the words whose frequency is less than twice is shown in the table. The best score is expressed in bold fonts.

Table 6: Comparison with other methods (ROUGE-1)

Methods	with			without		
	R1-R	R1-P	R1-F	R1-R	R1-P	R1-F
random	0.386	0.389	0.415	0.197	0.223	0.212
LexRank	0.602	0.389	0.472	0.465	0.305	0.368
our method	0.593	0.403	0.476	0.453	0.311	0.369
cut2	<b>0.629</b>	<b>0.415</b>	<b>0.500</b>	<b>0.475</b>	<b>0.319</b>	<b>0.382</b>

Making a comparison among all methods, the result indicates that the best summary is generated by our proposed method with recompiled documents removing the words appearing less than twice from the original documents. From this result, removing low frequency words from the objective documents to be summarized can be regarded as useful pre-processing to make sentence vectors which reflect the contents of the document.

## 5 Conclusions

In this study, we have proposed a graph-based online automatic summarization for time-series documents. The algorithm of our proposed method can deal with the renewal of the object documents to be summarized over time, and generate a summary of the documents at any time when it is required. Furthermore, by providing the limit on the size of a graph, it is not necessary to take account of irrelevant sentences for a summary. To evaluate the proposed method, we conduct an experiment to compare the proposed method with the other two methods, i.e., the method of randomly extracting sentences to make a summary and LexRank. We employ ROUGE as evaluation metrics. As a result, the proposed method gets close or higher accuracy than LexRank. Moreover, we have introduced an index to avoid redundancy in a summary by modifying MMR-MD — with the modified index, we provide a penalty for selecting a similar sentence to the already extracted sentences. As the result of the

experiments using MMR', we have confirmed that MMR' works to raise the accuracy in some cases, however, have not yet confirmed its usefulness as a whole. Furthermore, we assume that the low frequency words are not regarded as important to represent the contents of documents as well as stop words, therefore, we conducted a supplementary experiment on considering word frequency in object documents to be summarized. As the result of the experiment, we have confirmed that the accuracy gets better than any other methods if we remove the words which do not appear less than twice in object documents and generate a summary. By this fact, we can say that removing low frequency words leads to making sentence vectors which reflect the contents of documents, and it works well as a pre-processing for generating a summary.

As future work, as the first issue, we like to consider how to decide the initial value for the importance of a sentence in a graph — it is expected to be decided based on the ranking score at the previous time so as it will be useful for generating a summary at the current time. As the second issue, we will investigate more about the possibility of introducing MMR'. In this paper, we have not yet confirmed that the usefulness of MMR'. However, we think that providing a penalty for similarity of sentences should work to generate a summary, so we like to propose a better metrics which works well in the proposed algorithm. As the third issue, we could not use enough data in the experiments in this study, so, we will use more data to confirm the proposed method is useful. Finally, we will compare the proposed method with the other timeline summarization methods.

**Acknowledgements:** This work is partly supported by Kayamori Foundation of Information Science Advancement.

## References

- James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. *Temporal Summaries of News Topics*. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 10-18.



- Hai Leong Chieu and Yoong Keok Lee. 2004. *Query Based Event Extraction along a Timeline*. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 425-432.
- Gunes Erkan and Dragomir R. Radev. 2004. *LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization*. Journal of Artificial Intelligence Research, Vol. 22, No. 1, pp. 457-479.
- Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. *Multi-Document Summarization By Sentence Extraction*. In Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization, Vol. 4, pp. 40-48.
- Po Hu, Min-Lie Huang, and Xiao-Yan Zhu. 2014. *Exploring the Interactions of Storyline from Informative News Events*. Journal of Computer Science and Technology, Vol. 29, No. 3, pp.502-518.
- Hongyan Jing. 2000. *Sentence Reduction for Automatic Text Summarization*. In Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 310-315, Association for Computational Linguistics.
- Kevin Knight and Daniel Marcu. 2002. *Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression*, Journal of Artificial Intelligence, Vol. 139, No.1, pp. 91-107, Elsevier Science Publishers Ltd.
- Jiwai Li and Sujian Li. 2013. *Evolutionary Hierarchical Dirichlet Process for Timeline Summarization*. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Vou. 2, pp. 556-560.
- Chin-Yew. Lin. 2004. *ROUGE: a Package for Automatic Evaluation of Summaries*. In Proceedings of the Workshop on Text Summarization Branches Out, pp. 74-81.
- Ryan McDonald. 2007. *A study of global inference algorithms in multi-document summarization*. In Proceedings of the 29th European conference on IR research, pp. 557-564.
- Rada Mihalcea and Paul Tarau. 2004. *TextRank: Bringing order into texts*. In Proceedings of EMNLP-03 and the 2004 Conference on Empirical Methods in Natural Language Proceeding.
- M.F. Porter. 1980. *An algorithm for suffix Stripping*. Program, Vol. 14 No.3,pp.130-137.
- Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. *Centroid-based summarization of multiple documents*. Information Processing and Management: an International Journal, Vol.40, No 6, pp. 919-938.
- Giang Binh Tran, Tuan A. Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013. *Leveraging Learning To Rank in an Optimization Framework for Timeline Summarization*. SIGIR 2013 Workshop on Time-aware Information Access.
- Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. *Predicting Relevant News Events for Timeline Summaries*. In Proceedings of the 22nd international conference on World Wide Web Companion, pp. 91-92. International World Wide Web Conferences Steering Committee.
- Dingding Wang, Li Zeng, Tao Li, and Yi Deng. 2009. *Evolutionary Document Summarization for Disaster Management* In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 680-681.
- Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011a. *Evolutionary Timeline Summarization: a Balanced Optimization Framework via Iterative Substitution*. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 745-754.
- Rui Yan, Congrui Huang, Xiaojun Wan, Jahna Otterbacher, Xiaoming Li, and Yan Zhang. 2011b. *Timeline Generation Evolutionary Trans-Temporal Summarization*. In Proceedings of the Conference on Empirical Method in Natural Language Processing, pp. 433-443.
- Rui Yan, Xiaojun Wan, Yan Zhang, and Xiaoming Li. 2012. *Hierarchical Graph Summarization: Leveraging Hybrid Information through Visible and Invisible Linkage*. 16th Pacific-Asia Conference, PAKDD 2012, pp. 97-108.
- Wayne Xin Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. *Timeline generation with social attention*. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pp. 1061-1064.