Multiple News Articles Summarization based on Event Reference Information

Masaharu YOSHIOKA   Makoto HARAGUCHI
Graduate School of Engineering, Hokkaido University
N13 W8, Kita-ku, Sapporo-shi, Hokkaido, JAPAN
{yoshioka,makoto}@db-ei.eng.hokudai.ac.jp

Abstract

In this paper, we describe our method to generate summary from multiple news articles. Since most of news articles report several events and these events are refereed with following articles, we use this event reference information to calculate importance of a sentence in multiple news articles. We also propose a method to delete redundant description by using similarity of events. Finally we discuss its effectiveness based on the evaluation result.

1 Introduction

Nowadays, we can access a large number of text data. As a result, even for a simple topic, it becomes difficult to read through all documents that are related to the topic. Therefore, demand for multiple document summarization is increasing. In TSC-3, a task for multiple document summarization that uses news articles is proposed for tackling this issue. Most significant characteristic of multiple document summarization compared with single one is that there are redundant information in a document set. Since we assume this redundant information represents relationships among different documents, it is better to understand the relationships among the documents for finding out important sentences. In addition, we need to remove redundant information for generating compact summary.

In this paper, since most of news articles reports events that occurred at particular date and these events were refereed in following articles, we propose to use this event reference information to calculate importance of a sentence in multiple news articles. We also propose a method to delete redundant sentences by using this event information.

Finally we discuss its effectiveness based on evaluation result made by TSC-3 committee.

2 Extraction of the Event Reference Information

Since most of document sets used in TSC-3 task are categorized into a single-event type based on classification proposed by McKcown [5], a set of articles includes ones that reports occurrence of events and ones that reports following events (e.g., real fact of the event and sequel of the event) [3]. In this type of articles sets, following articles refer to the events that were already described in previous articles and add another information that were related to previous events. Therefore, we assume identification of events in different articles and reference information among these events is useful to make a summary.

Lexical cohesion method is one approach to deal with this reference information for summarizing a single document. However, in order to deal with reference information in different documents, we cannot use information such as distance between two sentences.

So we propose a method to extract event information from news articles and to identify event by using similarity measure between two events. In this paper, we define “*Event*” as follows.

Event is information that describes facts and related information on particular date.

2.1 Extraction of Events

Event is a unit to represent relationship among different articles and it should have information that is useful to identify same events. In order to extract rich event information from sentences in a document, it is better to analyze deep structure of sentences in a document; e.g., discourse analysis and anaphoric analysis. However, it is very difficult to use such deep structural information, we decide to use results of dependency analysis and we set a size of a unit simple. In addition, date information is useful for discrimination of similar events; e.g., press release in May is different from press release in April.

Based on this discussion, we select following slots to define an event.

Root is a word that dominates an event (verb that represents action or noun that represents subject or object)

Modifier is words that modify root word. Words are categorized into several groups, such as subject and object words for verbs and adjective and adnominal words for nouns.

Negative represents modality of expression.

Depth is a path length between Root of the event and root of the sentence in dependency analysis tree.

Date is a date that characterize the event. This slot is not a required slot to define an event.

ArticleDate is a date that the article was published.

Chunks represents list of word positions in a sentence.

In this method, we extract event information from a sentence by using following steps.

1. We apply Cabocha[7] to obtain dependency analysis tree.
2. We select verbs and nouns that have modification words as candidates of "Root" for events.
3. We check whether negative expression is included in root or not and set "Negative" based on this analysis.
4. We extract "Modifier" information from dependency analysis tree. At this time, we classify types of modifier by using POS tag and postpositional particle. Modifier information includes not only words that directly dependent on Root word but also modifiers for modifier words. Modifiers for a modifier word are categorized into the same category of the modifier words.
5. When we can extract date information from the sentence, we set this date as "Date" for events that has dependency with date words.
6. "Article Date" is obtained from article information.
7. "Depth" and "Chunks" are calculated by comparing event information with the dependency analysis tree.

Figure 1 shows a set of original sentence and extracted events.

```
Original:
市は「通常は５日前までに通告がある」と話し、県と基地周辺７市で１０日に抗議する。(Articles from Mainichi newspaper on January 10, 1998)
Extracted events:

Root ある
      Depth 2
      Subject 通常, 通告
      Date
      ArticleDate 980110
      Chunks 4,3,2,1
      postpositional-に ５, 日, 前
Root 話す
      Depth 1
      Subject
      Date
      ArticleDate 980110
      Chunks 5,4,2,1
      postpositional-と 通常, ５, 日, 前, ある
Root 抗議
      Depth 0
      Subject 市
      Date １０日
      ArticleDate 980110
      Chunks 9,8,7,6,10,0
      postpositional-で 県, 基地, 周辺, ７, 市
      postpositional-に １０, 日
```

Figure 1. Example of Event Extraction from a sentence

2.2 Dealing with Event Reference Information

We have already proposed an algorithm to calculate importance of a sentence in a single document based on PageRank [1] algorithm [6].

PageRank algorithm is the one that can calculate importance of WWW pages by using link analysis. Basic concept of the algorithm is distribution of page importance through link structure; i.e., page that has many links collects importance from other page and links from page with higher importance has higher importance compared to the links from ones with lower importance.

We formalize important sentence extraction algorithm by using following correspondence between web link structure and sentence structure.

A page in PageRank corresponds to a sentence.

A link in PageRank corresponds to sharing of same words in two different sentences (A and B). Since it is difficult to determine the direction of the link, we formalize that there are two links (A to B and B to A) for one sharing word.

In addition, all links in a page have same probability to distribute its importance in PageRank. However, in important sentence extraction, all words do not have same importance; e.g., sharing of large numbers of words has closer relationship than sharing of small numbers of words and sharing of rare words has closer relationship than sharing of common words. Therefore, we calculate importance of link based on the role of shared word in a sentence and Inverted Document Frequency (IDF). This important measure affects a transition matrix of PageRank that is used to calculate distribution of importance.

In this research, we expand this algorithm to deal with event reference information. Since we deal with multiple documents instead of a single document, there are following two approaches to expand this algorithm.

- First, we calculate importance of each document and distribute its importance. Second, we calculate importance of each sentence for each single document. Importance of each documents are distributed through this calculation.
- We merge all documents as a single document and calculate importance by using same algorithm for a single document.

In order to select one approach from them, we compare these two approaches according to the characteristics of single-event type news articles. Let us think about a news article that reports events that follows after first event. In this article, first events may only refereed once at the beginning of the article. When we employ the former approach, this means no links are generated for the first event and existence of the first event does not affect to calculate importance of the sentence. In contrast, this sentence has link from other articles and existence of the first event may arise importance of the sentence when we employ the latter approach.

Therefore, we employ latter approach to calculate importance of all sentences.

In previous algorithm, links are generated when two different sentences shares same word(s). In order to handle event reference information, we modify link generation algorithm.

Since we can express same events in different ways, identification of same event is difficult to identification of same word. Therefore, we introduce similarity measure between two events by using following two criteria.

1. Similarity of words
   First, we compare all words in "Root" and each category in "Modifiers." For each category, we calculate existence ratio of same word(s) that is biased by IDF. Second, we calculate weighted average of existence ratio for all categories. "Root" and "Subject" in "Modifiers" has higher weight compared to the other ones. We set threshold value to check whether the event pair belongs to candidate similar event pairs or not.
2. Judgment of consistency of date
   When the event has "Date" information, we verify consistency of date. When "Date" information lacks specific date such as year and/or month, we complement this information by using "ArticleDate" information. When one article has "Date" information and other one does not have this information, we compare them by using "ArticleDate" information. When inconsistency is found, the pair of events is removed from candidate similar event pairs.

We generate links between two different sentences that shares candidate similar event pair(s). We also calculate importance of link based on the importance of events in a sentence. Since we assume most important issues discussed in a sentence are located at the end of sentence, e calculate importance of event based on the "Depth" information. Another possible measure to calculate this importance is a frequency based measure. However, since frequency of events is already considered to calculate importance as a number of links, we do not use this measure.

We also set direction of the link as previous one; i.e., we formalize that there are bidirectional two links for one similar event pair.

We applied this event identification method for test run data and we found some links between two related events are missing. One reason of this problem is that we can express same event by using different vocabulary. However, we found some words are shared for these event sets. Therefore, we also use sharing of same word(s) for handling these relationships.

In PageRank algorithm, importance of page is calculated as a convergent vector of following recurrence formula.

m_{ij} is an element of transition matrix M of i-th row and j-th column and represents transition probability from j-th sentence to i-th sentence based on link structure. Since $\{m_{1j}, m_{2j}, \cdots, m_{nj}\}$ represents transition probability, $\sum_{i=1}^{n} m_{ij} = 1$. When there is a sentence k that has no relationship with other sentence, we set $m_{ik} = 0$.

$$\overrightarrow{r_{i+1}} = M\overrightarrow{r_i}$$

In order to handle both types of links (sharing events and sharing words), we make two matrices $M_e(me_{ij})$ and $M_w(mw_{ij})$ that corresponds to transition matrix made by sharing events and one made by sharing words, respectively. In order to satisfy constraint $\sum_{i=1}^{n} m_{ij} = 1$, overall transition matrix $M(m_{ij})$ is calculated by using parameter β and following formula.

$$\begin{aligned}
m_{ij} &= \beta * me_{ij} + (1-\beta) * mw_{ij} \\
& \quad when \sum_{i=1}^{n} me_{ij} \neq 0, \sum_{i=1}^{n} mw_{ij} \neq 0 \\
m_{ij} &= me_{ij} \\
& \quad when \sum_{i=1}^{n} mw_{ij} = 0 \\
m_{ij} &= mw_{ij} \\
& \quad when \sum_{i=1}^{n} me_{ij} = 0
\end{aligned}$$

Since a sentence that has no relationship with other sentence is meaningless to include into an abstract, we remove rows and columns that corresponds to the sentence in the matrix M. Calculation of a convergent vector is conducted by using eigen vector calculation. Since convergent vector satisfies following formula, convergent vector is an eigen vector of matrix M with eigen value = 1.

$$\overrightarrow{r_c} = M\overrightarrow{r_c}$$

2.3 Usage of Sentence Position and Initial Query

Since, in news articles, important sentences may be described early part of each article, we use sentence position information for calculating importance of each sentence. In PageRank, an algorithm to set initial importance of each page is proposed as Topic-Sensitive PageRank [2]. This algorithm is proposed to calculate importance of each page based on the category that the page belongs to. In this algorithm, they modify recurrence formula of PageRank as follows. $\overrightarrow{v}$ corresponds to initial importance vector and α is a parameter to control strength of the effect by the vector.

$$\overrightarrow{r_{i+1}} = (1-\alpha) * M\overrightarrow{r_i} + \alpha * \overrightarrow{v}$$

In this paper, we use simple formula $1/log(n+1)$ (n: sentence number in an article) for initial value of $\overrightarrow{v}$. $\overrightarrow{v}$ is normalized with $\Sigma_{i=0}^{m} v_i = 1$ (m is number of all sentences and v_i is an initial importance value for i-th sentence).

In order to handle initial query, we need to formalize effect of the query as link structures. In this research, we formalize initial query sentence is a document that has one sentence and is included in a multiple documents set. By using this formalization, we can distribute importance of the query. We may set initial importance for this sentence, but we just treat sentence as same as ones in other documents; i.e., $1/(log(1+1)$. In addition, we do not include this query sentence to following extraction process.

2.4 Text Reordering and Compaction based on Event Similarity

By using the algorithm discussed above, since similar sentences have similar links, similar sentences have similar scores. As a result, there is a chance to select redundant sentences when we select sentences from higher score ones. Therefore, we need a mechanism to detect such redundant sentences and it is required to remove such redundant description[4].

In this research, we use similarity measure of two events to calculate redundancy of new description. Since we can describe same information by using different numbers of sentences, we do not compare sentences one by one. We decompose a sentence into a set of events and we check redundancy of a sentence by comparing with an extracted event set that is obtained from extracted sentence; i.e., we calculate weighted average of existence ratio of events in the sentence. As we discussed in section 2.2, we assume most important issues discussed in a sentence are located at the end of sentence, we set higher weight for an event with lower “Depth.”

By using this redundancy check mechanism, sentence extraction algorithm is as follows.

1. Construction of an initial extracted event set
   A sentence with highest importance is selected as an extracted sentence. An initial extracted event set is constructed from events in the selected sentence.

2. Redundancy check and addition of new description
   Our system tries to add new description from a sentence with higher importance. The system checks redundancy of the sentence and add it when it does not exceed predefined redundancy level. The system also adds events in the selected sentence to the extracted event set. This step reiterates to select a desired number of sentences.

In order to generate abstract by using extracted sentences, we need a method to reorder extracted ones and remove redundant description from them. So, we modify sentence extraction algorithm as follows.

1. Construction of an initial extracted event set
   This step is same as the one in sentence extraction algorithm.

2. Redundancy check and addition of new description.

   (a) Detection of redundant sentence
   Our system tries to add new description from a sentence with higher importance. The system checks redundancy of the sentence, when it exceed predefined redundancy level, the sentence is rejected from candidate sentence to include in an abstract.

(b) Reordering sentences
The system reorders sentence by using following criteria.

- The system keeps sentence order in one article.
- The system keeps date order ("ArticleDate").
- When there are two or more articles with same "ArticleDate," we set following order constraints based on similarity between sentences. When there is a sentence who has precedence sentence that is similar to a sentence of same "ArticleDate" to be included in the abstract, this sentence should be located after the similar sentence.
- When the system fails to solve the constraints among some sentences, the system sets order of these sentences by using an order of articles that is given to a system (the system uses alphabetical order of the file name to define this order).

(c) Text compaction

i. Construction of an initial extracted event set
The system selects first sentence and add it to an abstract. An initial extracted event set is constructed from events in the selected sentence.

ii. Construction of a sentence to add
New sentence is selected by using order defined in previous step. The system remove redundant description from the selected sentence by comparing event information of the sentence and the extracted event set. We introduce following criteria to remove elements in a sentence.

- An event that has similar events in the extracted event set is selected as a candidate one to remove.
- Words, which belong to candidate ones to remove, are candidate words to remove. However, when "Root" element is also an element of other event that is not candidates one to remove, a word corresponding to "Root" element is removed from candidate words to remove.
- Words that have dependency with removed elements are candidate words to remove.
- The system modifies date information by using "Date" or "ArticleDate" information and a range of "ArticleDate." For example, when there is a description about "10 日" (10th) in the artile with "ArticleDate" 981022 (22th October, 1998). We modify date information as follows. When all articles are not published in a same year, we modify this description to '1998 年 10 月 10 日" (10th October, 1998). When all articles are published in a same year and not in a same month, we modify this description to '10 月 10 日" (10th October).
- The system makes new sentence by removing the candidate words.

(d) Check character size
Count character size of a generated abstract. When it does not exceeds the limit of given one, step 2 reiterates to generate longer abstract. When it exceeds the limit, new added sentence are removed from extracted sentences and reiterates step 2. When the system fails this check third consecutive times, it stops this reiteration process.

Figure 2 shows an example of text compaction process.

3 Experiment and Discussion

We apply this system for the task of TSC-3. In TSC-3, there are two subtasks. One is sentence extraction and the other is abstraction.

In this report, we do not use "set of questions about important information of the document sets" given by task organizer.

In the formal run, we set $\alpha = 0.3$ and $\beta = 0.9$ for generating abstracts for submit.

3.1 Sentence Extraction

Table 1 shows the evaluation results calculated by the task organizer.

Table 1. Evaluation Results of Sentence Extraction Results

	Short	Long
coverage	0.308	0.339
precision	0.505	0.585

In the formal run, we implement system that uses event reference information and word sharing information. However in order to evaluate the effectiveness

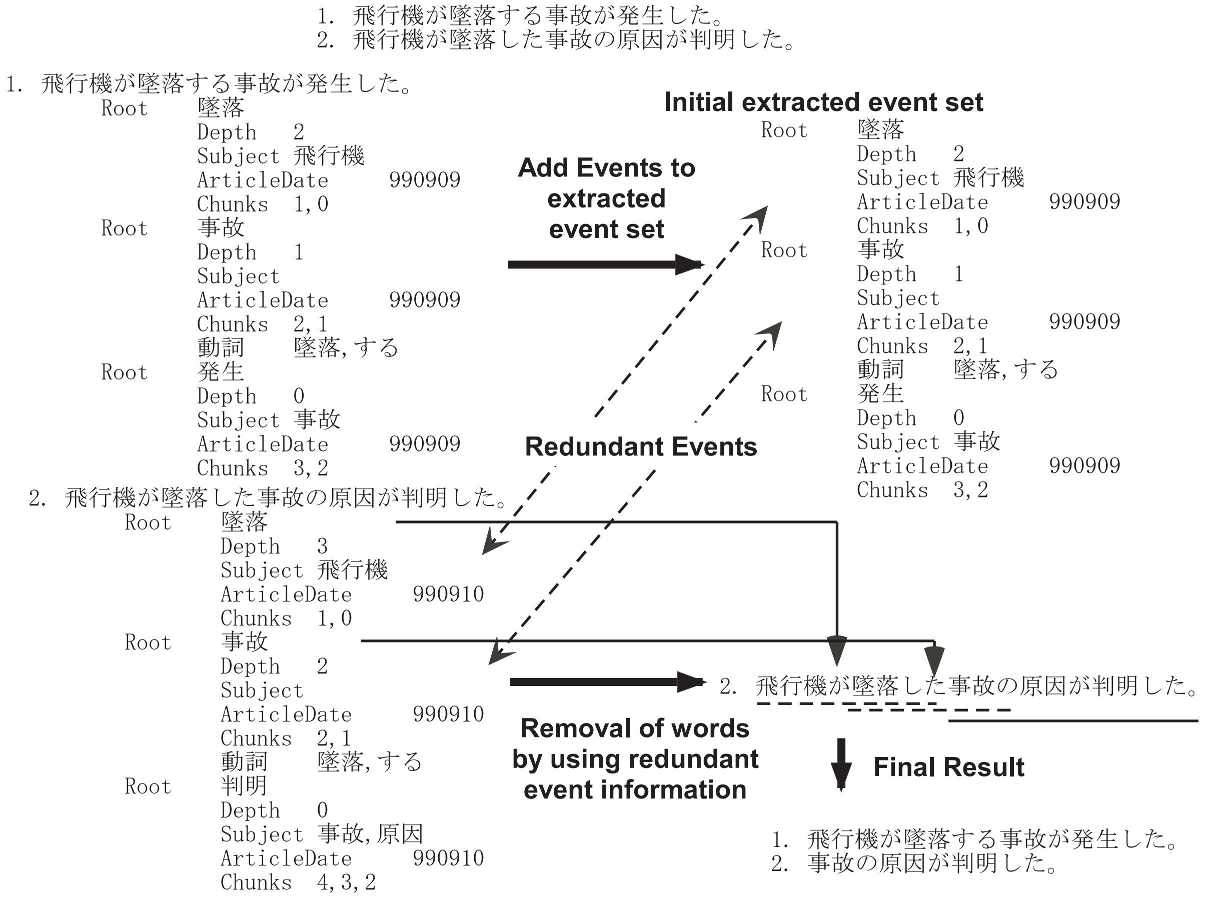

Figure 2. An Example of Text Compaction

of using event sharing information, we also conduct another experiments based on using transition matrix made only from event reference information and one only from word reference information. In addition, since LEAD method does not work well, we change $\alpha = 0.1$ to decrease effect of the sentence position. In addition, we also change $\beta = 0.3$ for better evaluation.

Table 2. Sentence Extraction Results with Different Information

		Short	Long
Event only	precision	0.325	0.313
	coverage	0.491	0.540
Word and Event	precision	0.323	0.341
	coverage	0.523	0.592
Word only	coverage	0.313	0.344
	precision	0.521	0.593

Table 2 shows a result of these experiments. From this result, calculation of importance by using event reference only has poorer performance than others.

Compared with links generated by word sharing information, we have less numbers of links generated by event reference information. Therefore, each link has more importance than that of word sharing one. However, this link information is not accurate at this moment; e.g., we still have a problem to identify similar events and we do not analyze anaphoric relation. We think that kind of inaccurate reference information may degrade a performance of the result.

3.2 Abstraction

Table 3 shows the content evaluation results calculated by the task organizer.

Table 3. Evaluation Results of Abstraction Results

	Short	Long
coverage	0.207	0.247
QA(exact)	0.390	0.356
QA(edit distance)	0.838	0.788

We compare our system with other system based on the readability evaluation results.

Following is a list of evaluation criteria that have better performance with average of other systems.

- q00: How many redundant or unnecessary sentences are there?
- q02: How many pronouns are there whose antecedents are missing?
- q04: How many expressions which have same meanings but different terms are there?
- q08: Does the summary have wrong chronological ordering?

From the evaluation result on q00, we confirm our method for removing redundant description works well. From the result on q08, we think reordering of sentences works well. We think better evaluation on q02 may be a side effect of this reordering. Since our method focuses on event reference by using surface information, when there are two or more variations to describe same events, most frequent description has most links. As a result, our system may tend to select similar description. We assume this is the reason why we have better evaluation on q04.

Following is a list of evaluation criteria that have worse performance with average of other systems.

- q01: How many places are there where (zero) pronouns of referring expressions to be used?
- q10: How many redundant verbs are there?

This problem comes from our method for text compaction. In the text compaction process 2c-2(c)ii, we remove redundant description based on event similarity. However in order to leave case elements for preserved event description, we left root elements as it is. However removal of such dependent elements means event related to this root word has already discussed. We assume this replication means that it is better to replace such root words as anaphoric word and as a result evaluation about q01 has worse evaluation result. For example, in the case of Figure 1, it is better to add “この” (this) infront of “事故” (accident). Our method of removing words from a sentence is too naive and that is a reason why we have worse evaluation on q10.

3.3 Discussion

There is another issue to discuss. Our method has better performance in “Long” compared with “Short.” Since our algorithm does not pay attention to the length of a sentence and longer sentence has more chance to have more links, a longer sentence tends to have higher importance. As a result, for the “Short” abstraction, such a longer sentence takes larger room and it becomes difficult to add another sentence. However for the “Long” abstraction, removal of redundant description may make new room to add another sentence. For the future work, it is necessary to have a mechanism that pays attention to the length of a sentence.

4 Conclusion

In this research, we propose a method to extract important sentences and to generate an abstract based on event reference information. We confirm our method has little bit better than average, but we need more effort to brush it up. In this evaluation, we confirm our method is good at detecting redundant description and ordering sentence extracting from multiple news articles.

Acknowledgment

We would like to thank organizer of TSC-3 for their effort to construct this test data. In this paper, we use news articles data of Mainichi newspaper and Yomiuri Newspaper on year 1997 and 1998.

References

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[2] T. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference*, 2002.

[3] T. Hirao, M. Okumura, T. Fukushima, and H. Nanba. Construction and evaluation of tsc3 corpus. In *Proceedings of The Tenth Annual Meeting of The Association for Natural Language Processing*, pages 588–591, 2004. (in Japanese).

[4] I. Mani. *Automatic Summarization*. John Benjamin Publishing Company, 2001.

[5] K. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, M. Y. Kan, B. Schiffman, and S. Teufel. Columbia multi-document summarization: Approach and evaluation. In *Proceedings of Document Understanding Conference 2001*, 2001.

[6] M. M.Haraguchi and M.Yoshioka. Towards an organization and access method of story databases. In *7th World Multiconference on Systematics, Cybernetics and Informatics (SCI2003) Vol. V*, pages 213–216, 2003.

[7] Y. M. Taku Kudo. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.