Two Different Summarization Methods at NTCIR3-TSC2: Coverage Oriented and Focus Oriented

Naoaki OKAZAKI [†] Yutaka MATSUO [‡] Naohiro MATSUMURA [†]
Hironori TOMOBE [†] Mitsuru ISHIZUKA [†]
[†]Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{okazaki, matumura, tomobe, ishizuka}@miv.t.u-tokyo.ac.jp

[‡]Cyber Assist Research Center
AIST Tokyo Waterfront
2-41-6 Aomi, Koto-ku, Tokyo 135-0064, Japan
y.matsuo@carc.aist.go.jp

Abstract

We are conducting research on multi-document summarization, participating in a competition of summarization, TSC (Text Summarization Challenge) task organized by NTCIR-3 project. In a dry run, we conceived a new extraction method for multi-document summarization which extracts a set of sentences that maximizes coverage of an original text and minimizes redundancy of a summary. Thinking over the result of the dry run, we decided to build another system for the formal run which generates a more focused summary. It employs a headline sentence and similarity of sentences to grasp the major points of original articles. In addition to them, we consider sentence ordering and reduction. We compare these summaries to discuss effectiveness of each method.

Keywords: *summarization, sentence extraction, cooccurrence relation, spreading activation, TSC*

1 Introduction

Information pollution driven by computerized documents leads to a problem of how to reduce the tedious burden of reading them. Automatic text summarization is one solution to the problem, providing users with a condensed version of an original text [4].

We frequently encounter related documents, for example, a collection of documents or web pages retrieved from a search engine through some queries, messages on an Internet discussion board or mailing list, collected papers on a certain research field, etc. A summary made by gathering summaries of each document has an adverse consequence that it will contain some redundant expressions or lack some important passages. Multi-document summarization, which is an extension of summarization of such related documents, has attracted attention in recent years.

The rest of the paper is organized as follows: The following section describes an overview of our summarization system in a dry run and its evaluation; and subsequent sections address the formal run and its evaluation. In Section 4, we discuss a comparison of the two systems. We discuss the future work and conclude this paper.

2 Summarization system in the dry run

2.1 Aim of summarization in the dry run

As related documents contain some similar expressions, extracting significant textual units often results in a redundant summary [8] . Therefore, we propose a new extraction method for multi-document summarization which aims at minimum inclusion of duplicate information as well as maximum coverage of original content. It uses a word cooccurrence graph and searches for an optimal combination of sentences by cost-based hypothetical reasoning [1].

2.2 Formulation of extracting sentences

We formulate the multi-summarization problem as follows.

First, we make an undirected graph of word cooccurrence from documents. In this paper, two terms in a sentence are considered to co-occur once. That is, we see each sentence as a “basket” and ignore term order and grammatical information except to extract word

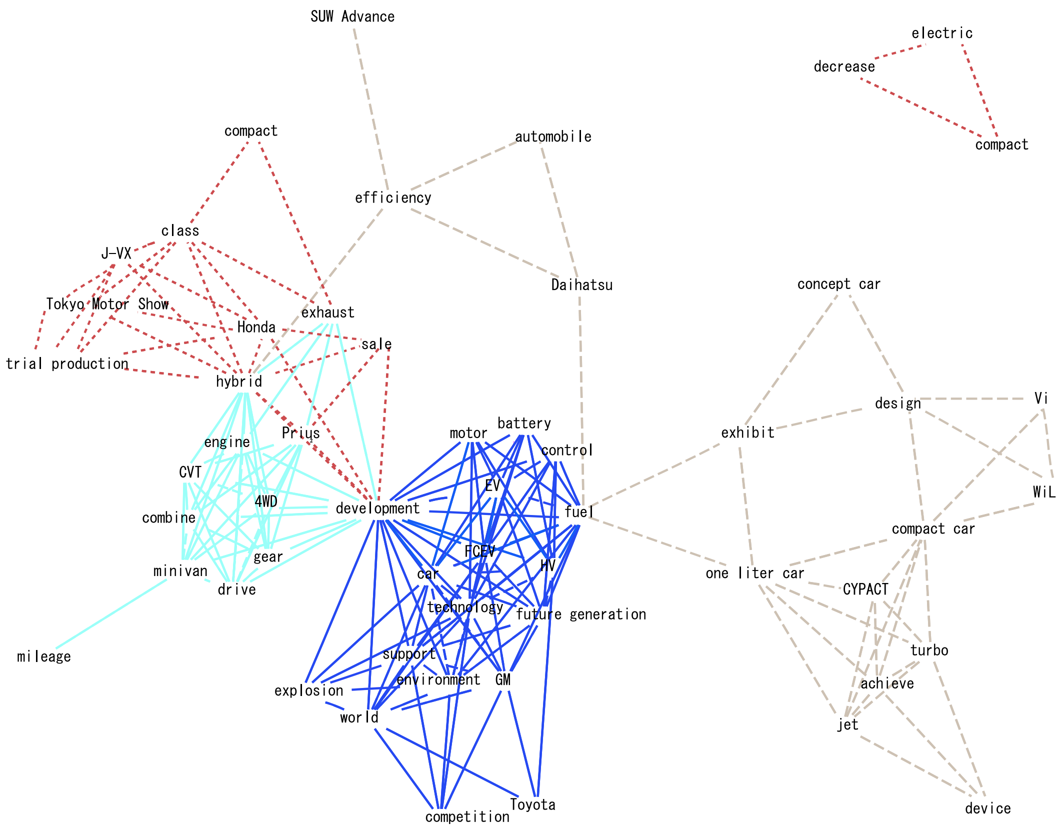

Figure 1. A word cooccurrence graph of a set of news articles. The source articles are a set of news articles about hybrid car development from the Mainichi newspaper (originally written in Japanese). The distance between nodes (terms) is roughly inversely proportional to the instances of cooccurrence. A line style corresponds to an article.

sequences. Fig. 1 shows a word cooccurrence relation between terms in a set of articles about “hybrid car.” A node represents a term; and we link nodes when a pair of terms appears in the same sentence more than twice.

What kind of sentences are characteristic in the graph? Each sentence in a document presents relations between terms [3]. That is equivalent in the graph to covering several links. As a consequence, we should choose a set of sentences that covers as many links as possible in the graph. It is useless, on the other hand, to choose a sentence which covers the same links as the previously selected sentence. Therefore, we obtain an edge covering problem defined as the following optimization problem,

$$\text{Min. } \mathrm{f} = \sum_{i \in K} cost_i x_i \tag{1}$$

$$\text{subject to } \sum s_j l_j \leq L, \tag{2}$$

where K is a set of links, $cost_i$ is a penalty cost when link i is not included in the summary, and x_i is a 0–1 boolean variable indicating whether link i is included (0) or not (1). s_j is a 0–1 variable indicating whether sentence j is added to the summary (1) or not (0); l_j is the number of letters in sentence j, and L is the limitation length of summary. If $s_j = 1$, all links in sentence j are to be selected.

2.3 Transformation of the optimization problem into cost-based hypothetical reasoning

We solve the optimization problem by applying cost-based hypothetical reasoning as follows. We denote k as the total number of links and m as the total number of sentences. We define goal G as representing *all links are taken into consideration* as follows.

$$G \leftarrow x_1, x_2, ..., x_k \tag{3}$$

A hypothesis h_{s_j} represents *sentence j is selected* and has no cost. For example, if a sentence has link#13, link#220, link#223, then we obtain the following rules.

$$x_{13} \leftarrow h_{s_1}, x_{220} \leftarrow h_{s_1}, x_{223} \leftarrow h_{s_1} \tag{4}$$

For unselected link i, on the other hand, we introduce hypothesis h_{emp_i} to represent *sentence i is not included in the summary* and the following rules.

$$x_i \leftarrow h_{emp_i}(i = 1, ..., k) \qquad (5)$$

We annotate h_{emp_i} with a penalty cost. The more this cost increases, the more link i is likely to be included into the summary.

Finally, we can describe the summarization problem which represent *selecting a set of sentences so as to minimizes the number of uncovered links*. However, the simplest solution to this problem is selecting all sentences with the sum of cost 0. We must introduce a constraint for outputting length, which is essential to the summarization task.

We use a fast hypothetical reasoning method [6] which solves a hypothetical reasoning problem quickly by transforming the problem into two continuous optimization problems. We can also describe some constraints among variables in free format with this method. So, we add the following constraint to represent (2):

$$39h_{s_1} + 77h_{s_2} + 54h_{s_3} + ... \leq 500 \qquad (6)$$

That is to say, the length of sentence 1 is 39 letters, sentence 2 is 77, sentence 3 is 54, ..., and summarization length must be within 500 letters.

In this way, we can decide a set of sentences by generating a knowledge base and finding a combination of sentences that proves goal G.

2.4 Implementation

First, we analyze the source text into a morpheme and identify the part of speech of each term by using Chasen. [1] Sorting nouns and verbs from terms, we enumerate cooccurrence between the terms in the same sentence. Then, we make and solve summarization problem described above.

2.5 Evaluation

In the dry run, 16 topics (sets of articles) were assigned to be summarized. Although the summaries are omitted due to space limitations, our summary of an article collection about "hybrid car" depicts various efforts of makers toward hybrid car development. Despite absence of a such heuristic as extracts the lead sentence, our system extracts them numerous times.

For an article collection about earning gold medals of Japanese athletes, in addition to prompt reports, our system includes some anecdotes about the victories.

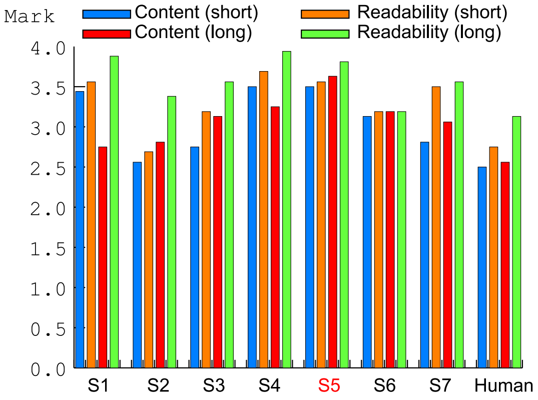

Figure 2. Subjective evaluation in the dry run. Sx stands for "System #x" and ours is S5. Lower mark is better.

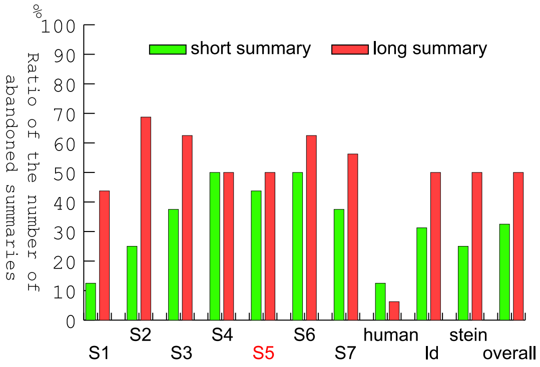

Figure 3. The number of abandoned summaries to correct in the dry run. Sx stands for "System #x" and ours is S5.

It is not novel in the summary that Japanese athletes won the games because articles were collected intentionally with queries, "*Nagano Olympics, Japan, gold, win.*" These queries often appear in the same sentence and have close cooccurrence relations. Because our summarization strategy tends not to bring such similar cooccurrence relations into a summary, it chose instead some "secret" stories which some users might not know.

Fig. 2 shows the average of subjective evaluation of summaries made by systems (S1–S7), and humans (Human). As can be seen from it, our system lost our popularity among subjects in terms of both content and readability, at either short or long summary.

Another subjective evaluation was correction of submitted summaries by a professional summarization. As shown in Fig. 3, which denotes how many summaries the corrector gave up; about half of our summaries were abandoned during correction.

The main operation for correcting our summaries

[1] A morphological analyzer by Computational Linguistics Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology (NAIST). Available at http://www.chasen.org/

is deleting sentences, which is to say that our method compiles numerous useless sentences. This suggests that it overemphasizes coverage of original content.

3 Summarization system in the formal run

3.1 Aim of summarization in the formal run

Thinking over the dry run result, we decided to build a new system for the formal run which generates a more focused summary. We use a headline sentence and similarity of sentences to grasp the major points of original articles.

In addition to extracting sentences, we consider sentence ordering to improve summary readability, detecting sub-topics within the original articles. We also try to eliminate redundancy within a sentence to improve the compression ratio.

3.2 Preprocessing

Sense disambiguation of words As we detail later, sentence extraction by spreading activation requires similarity of sentences.

Sentence similarity can be calculated from lexical relations between terms appearing in a sentence and others. That is, we have to calculate similarity of terms for extracting sentences.

For calculating similarity of terms, we use a Japanese lexical dictionary, *Nihongo Goi Taikei*, [2] It consists of three sub dictionaries, “lexical system”, “word system”, and “syntactical system”. The “noun lexical system” maps nouns into a tree structure, which consists of 2,710 nodes of semantic attributes. Because the tree has the property that a node connotes semantic attributes of descendant nodes, we can estimate similarity of terms by distance between terms on the semantic tree. Therefore, our only task is to identify the attribute to which the terms belong.

Although a human can determine immediately the meaning of a term in the context of a text which has a number of meanings, computers do not have such an ability. We can not calculate similarity of terms without identifying the meanings. We formulate the word-sense disambiguation problem as follows.

We define $\mathbf{T} = (t_1, t_2, ..., t_n)$ as a noun term which appears in an article. We introduce $\mathbf{A_i}$ to enumerate the possible semantic attributes of term t_i, consulting the dictionary, *Nihongo Goi Taikei*. For example, for a word ’system’, five attributes #362 (organization), #962 (machine), #1155 (institution), #2498 (structure), #2595 (unit) are found,

$$t_1 = \text{‘system’}, \mathbf{A_1} = \{362, 962, 1155, 2498, 2595\} \tag{7}$$

When t_i has no entry in the dictionary (i.e. unidentified terms), we leave $\mathbf{A_i}$ as empty,

$$\mathbf{A_i} = \{\} \tag{8}$$

Then, we choose $a_i \in \mathbf{A_i}$ for each i that maximizes the following $score$,

$$score = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \min\{4 - \mathrm{distance}(a_i, a_j), 0\}, \tag{9}$$

where $\mathrm{distance}(a_i, a_j)$ is the distance between node #a_i, #a_j on the semantic tree. We define $\mathrm{distance}(a_i, a_j) = \infty$ in case of $\mathbf{A_i} = \{\}$ or $\mathbf{A_j} = \{\}$.

Through optimization, in other words, we determine an attribute of each term adopting lexical cohesion as a context of original articles [7].

Clustering articles The collection of articles retrieved with a query, ”Great Taiwan Earthquake”, for example, contains articles which make a quick report of an earthquake occurring, reports of the earthquake center and magnitude, on-the-spot reports from the scene of the disaster, reports of support from other countries, and so on. In such a case, we should order extracted sentences along the sub-topics to improve summary readability.

We can assume a newspaper article to be written for one topic. Hence, to classify sub-topics in a summary, all we have to do is classify articles by their topics. We apply the nearest neighbor method [2] for clustering after measuring distance or similarity between two articles, i and j, as follows:

$$\mathrm{distance}(\mathbf{D}_i, \mathbf{D}_j) = 1 - \mathrm{Sim}(\mathbf{D}_i, \mathbf{D}_j) \tag{10}$$

$$\mathrm{Sim}(\mathbf{D}_\mathrm{i}, \mathbf{D}_\mathrm{j}) = \frac{\mathbf{D}_\mathrm{i} \cdot \mathbf{D}_\mathrm{j}}{|\mathbf{D}_\mathrm{i}||\mathbf{D}_\mathrm{j}|}, \tag{11}$$

where $\mathbf{D}_i$ is a vector of article i, whose element is term frequency.

$$\mathbf{D}_i = (\mathrm{tf}_1^i, \mathrm{tf}_2^i, \mathrm{tf}_3^i, ..., \mathrm{tf}_n^i)^t \tag{12}$$

where tf_t^i is the number of occurrences of term t in the article i. In this way we merge a pair of clusters when their minimum distance is lower than 0.4.

3.3 Sentence extraction by spreading activation

Sentence selection by headline Headline is a straightforward summary of an article by the author. Words occurring in the headline serves many uses as keywords that represent the article. Also, it is manifested that articles for summarization are retrieved from the only one news source in TSC2. Headline words in such a situation are not the only representatives of the article but also something characteristic to

[2] NTT Communication Science Laboratories, Iwanami Shoten.

the article because the author clarifies difference between this article and previous articles to attract the reader interest.

For that reason, we extract all sentences which contain a term occurring in the headline of each article.

Spreading activation through the similarity of sentences Because sentence selection by headline is a process of passing over those which are irrelevant to the thrust, a great deal of sentences still remains as summary candidates.

We have represented that the goal of extraction in the formal run was drawing up a centered summary. Therefore, we rank sentences by spreading activation with the assumption that, *"Sentences which are relevant to ones of significance are also significant."* Our method differs from some studies such as [5] in that ours ranks sentences directly by spreading activation with the use of sentence similarity.

First, for all pairs of sentences, we calculate sentence similarity by the following formula.

$$\mathrm{sim}(S_i, S_j) = \sum_{t_i \in S_i} \sum_{t_j \in S_j} \frac{0.5^{\mathrm{distance}(t_i, t_i)}}{\sqrt{|S_i||S_j|}} \qquad (13)$$

$|S_i|$, $|S_j|$ are the numbers of indexing terms in sentences S_i, S_j respectively. $\mathrm{distance}(t_i, t_i)$ stands for the semantic distance between term t_i and t_j defined as:

$$\mathrm{distance}(t_i, t_j) = \begin{cases} 0 & (t_i, t_j \text{ are identical}) \\ \mathit{length} + 1 & (\mathit{length} < 4) \\ \infty & (\mathit{length} \geq 4), \end{cases} \qquad (14)$$

where length is the distance between term t_i and t_j from the viewpoint of semantic tree. We assume that terms t_i and t_j are similar when t_i and t_j are identical or close on the semantic tree.

Next, we link a pair of sentences S_i and S_j if $\mathrm{sim}(S_i, S_j) > 0$. In this way, we make a network graph which indicates the similarity relationship of sentences. Then, we continue spreading activation by the following formula.

$$\mathbf{A}^{(k)} = \alpha \mathbf{I} + (1 - \alpha)\mathbf{R} \cdot \mathbf{A}^{(k-1)} \qquad (15)$$

$\mathbf{A}^{(k)}$ is a n-vector whose element is an activation after k steps, $\mathbf{I}$ is a n-identity matrix, $\mathbf{R}$ is a spreading matrix($n \times n$) which shows similarity. $\mathbf{R}_{ij}$(an element of $\mathbf{R}$) represents strength of similarity between sentences S_i and S_j:

$$\mathbf{R}_{ij} = \begin{cases} \frac{\mathrm{sim}(S_i, S_j)}{\text{the number of links of } S_j} & (\text{if } i \neq j) \\ 0 & (\text{if } i = j) \end{cases} \qquad (16)$$

α is a parameter which determines activation inserted to the network.

パイロットらの賃金制度改定をめぐる全日空の労使対立で、同社の乗員組合は６日午前０時から国際線の一部路線で無期限ストライキに突入した。**パイロットらの賃金制度改定をめぐる全日空の労使対立で、** 乗員組合は６日、７日も国際線の一部路線でストライキを継続することを決めた。

(English Translation)

Due to labor-management difficulties involved in revision of pilots' wage plan of All Nippon Airways Co., Ltd., the crew union went on strike indefinitely on some of international airlines at 0 a.m. of the 6th. **Due to labor-management difficulties involved in revision of pilots' wage plan of All Nippon Airways Co., Ltd.,** the crew union, on the 6th, decided to keep on strike on some of international airlines of the 7th.

Figure 4. A typical example of duplication (with rough English translation). The boldface clause is a repeated expression.

In the network model, we set an injection parameter α to be 0.15 and initialize $\mathbf{A}^{(k)}$ with a given value. Then, we apply the formula (15) until convergence, normalizing $\mathbf{A}^{(k)}$ for each step to satisfy this:

$$\sum_i \mathbf{A}_i^{(k)} = 1 \qquad (17)$$

3.4 Eliminating similar clauses

We can acquire a set of important sentences by extracting highly activated sentences up to a specified summarization length. This can be a good summary which centers on several key points because we do spreading activation with the assumption that *"Sentences which are relevant to the ones of significance are also significant."* On the other hand, this may also lead to extraction of a set of sentences which may contain many redundancies. Related newspaper articles often contains a pair of sentences like these in Fig. 4, which have a lot in common but describe slightly separate subjects. Eliminating such a repeated expressions has also been an issue of multi-document summarization.

In order to achieve this, breaking up each sentence into several units (or clauses), we delete some redundant units. We use KNP [3] for identifying clause-like units in a sentence and delete units which are similar to previously-included content.

Concerning calculation of similarity of clauses, we can reuse the method to calculate that of sentences. However, this may result in inaccurate estimation because of fewer pairs of terms for comparison. Consequently, we employ another method in which we weight terms of a clause according to how a term contributes to the gist of the clause and compare with the

[3] Language Media Laboratory, Graduate School of Informatics, the University of Kyoto.

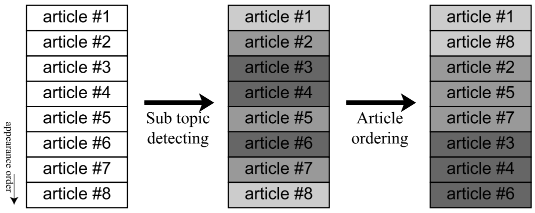

Figure 5. The strategy of sentence (article) ordering. There are three sub topics detected (half-tone dot meshed) in these articles.

weights. Presuming that a more close term to the root on the semantic tree contributes more to the gist of the clause, we attribute higher weight to it.

3.5 Post-processing

Sentence ordering Having classified articles by their sub topics in the preprocessing phase, we order the extracted sentence so as not to lose the thread of the argument. Sub topics are sorted by the order of “time stamp of sub topic”, which represents the date of the oldest article in the sub topic. Articles in each sub topic are sorted by date on which the article was written (i.e. Fig. 5); we do not change sentence order in each article.

Entreatment of reference term Extracting gathers sentences from all over the source text. Although some are extracted adjacently along with previous sentences, some are isolated.

When an extracted sentence has a reference term, we exceptionally nominate the sentence just before it as a candidate for summary sentences.

Normalization of verbs Here is a sentence made up of three clauses, “**A B C.**”; and the clause **B** is similar to a clause coming before. Clause **B** is fated to be deleted as a consequence of eliminating similar clauses. After deleting it, we normalize the conjugation form of clause **A** to generate two sentences, “**A.**” and “**C.**” Normalization is not applied to the last clause in a sentence (i.e. clause **C**).

Deletion of a quote When a newswriter quotes someone in an article, he or she will append a summary after someone’s long statements in a sentence. In this case, we can compress the sentence by blacking out the section concerning the quotational phrase. So we recognize a quotational phrase which begins at the open quote and which ends at the closing quote or its successive adverb phrase. When there are less than eight letters remaining after the quotational phrase, we leave the phrase as it is so as not to make the sentence incomprehensible. We also leave a quotation phrase before a reverse conjunction.

> 小渕恵三首相は６日、広島市で記者会見し、金融機関の不良債権処理のための金融再生トータルプラン関連６法案について、**「与野党でこだわって問題の一つが解決しなければすべて法律が成立しないというのは国民のためにならない。必ずや国会で考え方の一致を見て法律が成立すると期待している」と述べ、**６法案成立のためには野党との修正協議に柔軟に応じ、早期成立を図る考えを示した。
>
> *(English Translation)*
>
> On the 6th at a press conference held in Hiroshima, prime minister Keizo Obuchi, concerning financial reconstruction total plan related six bills for handling the bad debts of financial institution, **said “It does not benefit the nation that no legislation is enacted before the resolution of an issue in which ruling and opposition parties are absorbed. I hope the legislation will be enacted in the Diet session with their consent.” and** revealed his idea that he had a flexible attitude over changes in the legislation with the opposition in order to pass the bills early.

Figure 6. An example of quote deletion (with rough English translation). The boldface segment to be deleted.

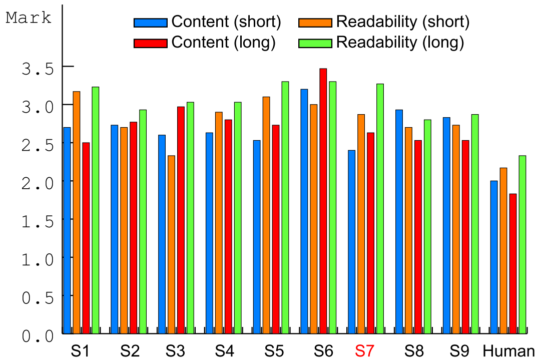

Figure 7. Subjective evaluation in the formal run. Sx stands for “System #x” and ours is S7. Lower mark is better.

Miscellaneous We delete a conjunction at the beginning of a sentence, phrases in parentheses, and so on.

3.6 Evaluation

In the formal run, 30 topics (sets of articles) were assigned for summarization. Subjective evaluation of our system was improved noticeably. Figures 2 and 7 show that our summary in the formal run got a more favorable impression (by about 1 mark for content, 0.5 for readability) than that of the dry run. Although we cannot simply look at these results, because the summary by human was also improved, it can readily be

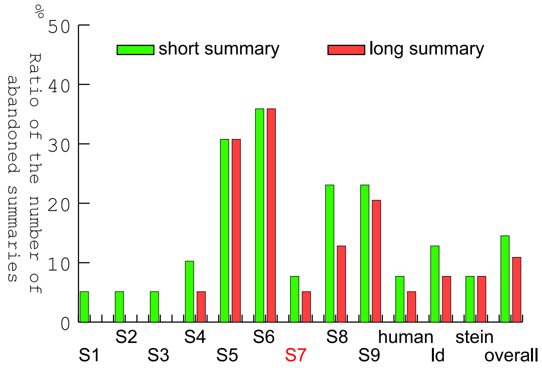

Figure 8. The number of abandoned summaries to correct in the formal run. Sx stands for "System #x", and ours is S7.

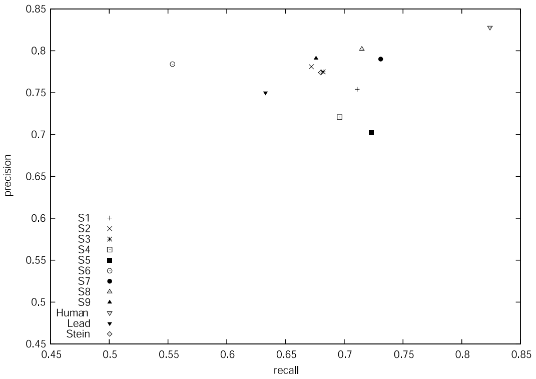

Figure 9. Precision-recall-like evaluation for short summaries.

said that, for content in short summary, our system contended for first place in return.

This can be seen from Fig. 8 as well. The number of abandoned summaries is decreased from about 50% to 7% or 8% while that of human remained unchanged. From the fact that the probability of rejection is identical to that of human, our summary in formal run seems to be acceptable to the corrector.

Figures 9 and 10 are precision-recall-like evaluation of each summarization length. Precision and recall in this evaluation are defined as follows:

$$\text{precision} = 1.0 - (\text{sum of deletion ratio}) \qquad (18)$$

$$\text{recall} = 1.0 - (\text{sum of insertion ratio}) \qquad (19)$$

The sum of deletion ratio denotes how many letters are deleted in the process of correction, and the sum of insertion does so correspondingly.

Strictly speaking, they are different from usual usage in that deletion or insertion ratios are not given to abandoned summaries. The more summaries of a system the corrector gives up, the lower the effective

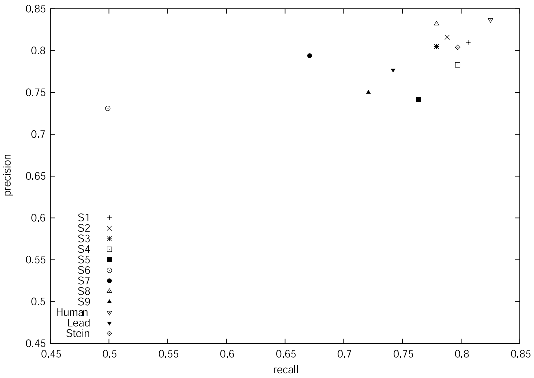

Figure 10. Precision-recall-like evaluation for long summaries.

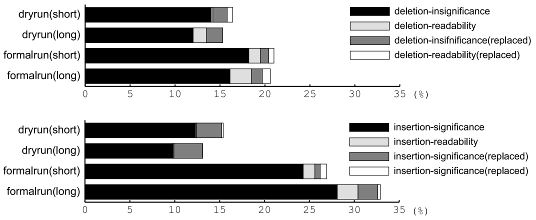

Figure 11. Detail of why deletion and insertion took place. The ratios of corrected letters to summary length are shown.

precision and recall may be because it can be estimated that deletion and insertion ratio of abandoned summaries would have been very high.

Even from Fig. 9, we can see that our system takes one of the leads for short summary. For the long summary (Fig. 10), on the other hand, ours does not seems to perform well, especially owing to the recall. This shows it is prone to including similar content and disregarding something unusual. Limitation of space at shorter summary leads us to disregard this bad habit since summaries with a few centers are enough for short summaries. Compared to this situation, at longer summaries, it is expected that it includes not only a few centers but more key points.

4 Discussion

In this section, we continue to discuss results of the dry run and the formal run and illuminate features of each method.

Fig. 11 shows how much and why deletion and insertion took place in correcting. It indicates the different nature of the two methods.

Note that we cannot say that the method in the dry run is superior to that in the formal run only because the amount of corrections in the dry run is fewer. As noted earlier with Figs. 3 and 8, the number of abandoned summaries in the dry run was extremely large.

We can see that from Fig. 11, in the formal run, the number of deleted letters for insignificance is lower than that of inserted letters for significance. This shows that the readers wanted to get some other sentences (i.e. content) as they approved that sentences in the summaries had their existence values. Regarding the comparison with summarization length, we also notice that the ratio of insertion gets larger for long summaries. This also reflects that our method in the formal run finds difficulty in extracting valuable sentences for longer summaries.

In the dry run, on the other hand, the number of deleted letters for insignificance is greater than that of inserted letters for significance. It would appear that readers were satisfied with the pool of information, but not well with the value of information.

These discussions can be summarized as follows: The summarization method in the dry run was actually a coverage oriented method and that in the dry run was focus oriented.

5 Conclusion

We introduced two summarization methods. The former is sentence extraction based on a word cooccurrence graph and edge covering problem. It has the unique feature of coverage of original content and inclusion of some novel stories which some users might not know. Because such a summary results in a rambling text, it requires improved visualization to capitalize on the feature.

The latter aims at a more focused and acceptable summary. It employs spreading activation through similarity of sentences for ranking sentences followed by sentence selection with headlines. In addition to extracting sentences, we considered sentence ordering to improve summary readability detecting sub-topics within the original articles. We also tried to eliminate redundancy within sentences to improve the compression ratio.

Summaries in the formal run won the title against those in the dry run according to the rating of TSC2. However, we will continue to research both methods while considering what should be done to combine their features.

References

[1] E. Charniak and S. E. Shimony. Cost-based abduction and map explanation. *Artificial Intelligence*, 66:345–374, 1994.

[2] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13:21–27, 1967.

[3] M. A. K. Halliday and R. Hasan. *Cohesion in English*. Longman, 1976.

[4] I. Mani. *Automatic Summarization*. John Benjamins, 2001.

[5] I. Mani and E. Bloedorn. Summarizing similarities and differences among related documents. *Information Retrieval*, 1(1-2):35–67, 1999.

[6] Y. Matsuo and M. Ishizuka. Two transformations of clauses into constraints and their properties for cost-based hypothetical reasoning. In M. Ishizuka and A. Sattar, editors, *PRICAI 2002: Trends in Artificial Intelligence (Proc. 7th Pacific Rim Int'l Conf. on AI, Tokyo)*, pages 118–127. Springer, 2002.

[7] M. Okumura and T. Honda. Word sense disambiguation and text segmentation based on lexical cohesion. In *Proceedings of COLING-94*, volume 2, pages 755–761, 1994.

[8] M. Okumura and H. Nanba. Automated text summarization: A survey. *Journal of Natural Language Processing*, 6(6):1–26, 1999.