PAPER *Special Issue on Text Processing for Information Access*

Sentence Extraction by Spreading Activation through Sentence Similarity

Naoaki OKAZAKI[†], ***Student Member***, **Yutaka MATSUO**[††], **Naohiro MATSUMURA**[†], ***Nonmembers, and*** **Mitsuru ISHIZUKA**[†], ***Member***

SUMMARY Although there has been a great deal of research on automatic summarization, most methods rely on statistical methods, disregarding relationships between extracted textual segments. We propose a novel method to extract a set of comprehensible sentences which centers on several key points to ensure sentence connectivity. It features a similarity network from documents with a lexical dictionary, and spreading activation to rank sentences. We show evaluation results of a multi-document summarization system based on the method participating in a competition of summarization, TSC (Text Summarization Challenge) task, organized by the third NTCIR project.

key words: *summarization, extraction, sentence similarity, spreading activation*

1. Introduction

Information pollution driven by computerized documents presents the problem of how to reduce the tedious burden of reading such texts. Automatic text summarization [1] is one solution to the problem, providing users with a condensed version of an original text.

There are two major types of summaries (or extracts), *a reading material* and *a run of items*. A summary shown by a run of items is a set of claused sentences or phrases. We should generate both a wide and shallow panoramic view of an original text when readers are content with itemization of essential parts. We should perform myriad processes (e.g., clustering and ordering items) to elucidate relationships among claused textual units because such a claused sentence or phrase gives fragmentary information.

On the other hand, a summary as a reading material is not only a collection of major points, but a well-formed text. When readers expect this kind of summary, we should provide an easy-to-read summary. They may find a poorly organized summary very hard to read; in the worst case, they may lose interest in the original document. However, it is very difficult for computers to work on the text to improve wording and generate a well-organized text. For that reason, we often subject the original sentences to minimal correction.

We have developed a novel method to extract a set of comprehensible sentences that centers on several key points with the intention of generating a summary as such a reading material. It features a similarity network generated from documents with a lexical dictionary and spreading activation through the similarity network to rank sentences.

The rest of the paper is organized as follows. The following section describes an overview of automatic summarization and related research using textual similarity for extracting. We propose our extraction method by spreading activation through sentence similarity with a lexical dictionary in Section 3. The subsequent section (Section 4) addresses shortly the second TSC (Text Summarization Challenge) of NTCIR Workshop and an implementation of a multi-document summarization system based on our method. After we evaluate our method in Section 5, we discuss future work and conclude this paper.

2. Summarization and usages of similarity measure

There has been a great deal of research on automatic summarization. The basic process of extraction is to find characteristic sentences by statistical methods such as term frequency [2], [3], cue phrases [4], titles [4], or sentence location [4]. However, extraction by statistical methods disregards the relationships between extracted textual units such as terms, sentences, and passages. It often yields an incomprehensible summary by agglomerating textual units recommended through statistical methods. Some methods are proposed to improve sentence connectivity.

Mani et al. [5] proposed a summarizing method based on a graphic representation of related documents. By exploiting meaningful relations between units based on an analysis of text cohesion and the context, it finds topic-related text regions using spreading activation, filters activated regions by segment finding, and extracts textual fragments instead of sentences. This method requires deep analysis of the original text.

Nagao et al. [6] proposed a similar approach. However, their approach uniquely introduces GDA (Global Document Annotation). Spreading activation

Manuscript received December 04, 2002.

Manuscript revised May 02, 2003.

[†] Graduate School of Information Science and Technology, the University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

[††] Cyber Assist Research Center, AIST Tokyo Waterfront, 2-41-6 Aomi, Koto-ku, Tokyo 135-0064, Japan

is performed in the network through use of an intra-document network, in which nodes correspond to terms and link to the semantic relations which are defined naturally by a GDA tagged document. It generates summary sentences directly from the semantic network, adding highly activated elements into the resultant summary. It becomes an effective method if GDA-tagged documents are given.

Fukumoto [7] proposed a method that first chooses a sentence that contain a query term of user input and sentences which have a strong similarity to the previously selected sentence. As it decides to extract a sentence one-by-one by comparing similarity, it does not consider overall network topology of sentence similarity. A reader must give a query term to determine a point (sentence) where the extraction process starts. When the reader does not have adequate knowledge of source documents, he or she may miss important sentences that have no connection with the query, or be at a loss for the query.

Salton et al. [8] suggested passage (paragraph) extraction from a document based on *intra-document links* between paragraphs. It yields a *text relationship map* from intra-document links, which indicate that the linked texts are semantically related. It proposes three strategies from the text relationship map: *bushy path*, *depth-first path*, and *segmented bushy path*, which determine important paragraphs.

A bushy path is constructed out of highly bushy nodes on the map, where the bushiness of a node on a graph is defined as the number of links connecting it to other nodes on the graph. A highly bushy paragraph is likely to discuss topics covered in many other paragraphs since it has an overlapping vocabulary with other paragraphs. A depth-first path, which starts at an important (highly bushy) node and visits the next most similar node at each step, aims at a readable extract. A segmented bushy path, which is a bushy path within a text segment, is expected to solve a problem by which the bushy and depth-first strategies tend to extract a slanted topic in the document.

Although the method proposed by Salton et al. [8] is similar to the one proposed by Fukumoto [7], Salton's method automatically chooses a passage where the extraction process starts and considers multiple subtopics within a document. However, the depth-first path strategy relies on similarities so heavily that it is hard to escape unimportant regions once it enters there. The problem with the depth-first path strategy becomes obvious when we apply this method to generate a summary constructed by fragmentary textual units (e.g., sentences). Also, it does not consider both sentence similarity and sentence importance simultaneously.

3. Proposed Method

Against the background of these studies, we propose a novel extraction method that ranks sentences by spreading activation with an assumption that *"Sentences which are relevant to many ones of significance are also significant."* This assumption derives from PageRank [13], which judges importance of web pages based on a recursizely defined assumption that *"Pages which are linked (voted) from many ones of significance are also significant."* We analogously determine a sentence to be a web page and sentence similarity to be a link on the web; we try to find important sentences in the "web" of similarity relation.

Where addresses conventional methods [7], [8] depend only on sentence similarity to choose important sentences, spreading activation through sentence similarity considers how much we can trust a similarity to vote another important sentence from the importance of the sentence forming the similarity relation. It adjusts sentence importance and similarity at the same time. Our method is also circumspect of the overall network topology of sentence similarity.

A sentence vector may become too sparse to supply sufficient amount of hints for measuring similarity because a sentence contains much less indexing terms than a paragraph or document does. In addition, two similar sentences do not always share the exact same terms because a human often uses paraphrasing. Therefore, we introduce a lexical dictionary to refine a similarity measure of sentences.

3.1 Sentence Similarity

Sentence extraction by spreading activation, as we detail later, requires sentence similarity. Sentence similarity can be calculated from lexical relations between terms appearing in a sentence and others. When we estimate sentence similarity, we must consider two problems: *how to estimate term similarity*; and *how to calculate sentence similarity from it.*

3.1.1 Estimation of term similarity

We use a Japanese lexical dictionary, *Nihongo Goi Taikei* [†] for estimating similarity of terms. It consists of three sub-dictionaries: "lexical system", "word system", and "syntactical system". The "noun lexical system" maps nouns into a tree structure which comprises 2,710 nodes that represent semantic attributes.

Figure 1 is an abridgement of the semantic tree of the noun lexical system. We can estimate similarity of terms by the distance between terms on the semantic tree because the tree has the property that a node

[†] NTT Communication Science Laboratories, Iwanami Shoten.

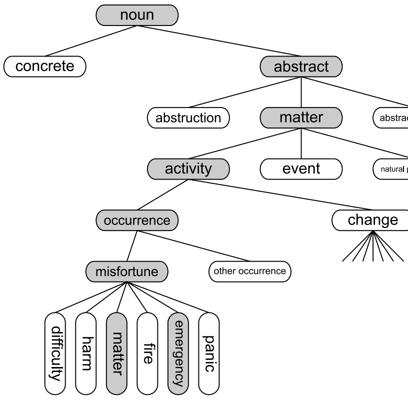

Fig. 1 Semantic tree of "noun lexical system" in *Nihongo Goi Taikei*. This figure is an abridgement for explanation from the manual.

connotes semantic attributes of descendant nodes. For example, we find that the noun 'incident' belongs to an attribute *matter* and the noun 'terrorism' to an attribute *emergency* by consulting a dictionary. Figure 1 indicates that the attributes *matter* and *emergency* have a sibling relationship; we can admit that the words *incident* and *terrorism* have a close relation.

Examining the semantic tree carefully, we notice that the number of terms that exist along a path from one attribute to another increases exponentially in proportion to the path length on the tree. In other words, the relationship between two terms is inversely exponential to path length since the number of terms on the path increases exponentially. Hence, we should define similarity of two attributes by the exponential function. We define similarity of two terms t_i and t_j as

$$\mathrm{sim}(t_i, t_j) = \gamma^{\mathrm{distance}(t_i, t_j)}, \tag{1}$$

where $\mathrm{distance}(t_i, t_j)$ is the path length between the terms, and an attenuation factor γ ranges $0 < \gamma < 1$. We determine γ to be 0.5 vaguely, as similarity of two terms belonging to the same semantic attribute will be 0.5 since they do not always have a synonymous relation. Finally, we define semantic distance of two terms, t_i and t_j, as the following.

$$\mathrm{sim}(t_i, t_j) = 0.5^{\mathrm{distance}(t_i, t_j)} \tag{2}$$

When t_i and t_j are identical, we define $\mathrm{distance(t_i, t_j)}$ to be 0.

$$\mathrm{distance}(t_i, t_i) = 0 \tag{3}$$

In cases where t_i and t_j are not identical, let two terms t_i and t_j belong to attributes a_i and a_j, respectively. We define $\mathrm{distance(t_i, t_j)}$ as the following,

$$\begin{aligned}&\mathrm{distance}(t_i, t_j)\\ &= \begin{cases} \mathrm{path\_length}(a_i, a_j) + 1 & (\mathit{length} < 4) \\ \infty & (\mathit{length} \geq 4) \end{cases},\end{aligned} \tag{4}$$

where $\mathrm{path\_length}(a_i, a_j)$ is the path length between attributes a_i and a_j on the semantic tree. In the above example, we calculate distance('incident', 'terrorism') as follows,

$$\begin{aligned}&\mathrm{distance}(\text{'incident'}, \text{'terrorism'})\\ &= \mathrm{path\_length}(\mathit{matter}, \mathit{emergency}) + 1\\ &= 3.\end{aligned} \tag{5}$$

We do not admit semantic relation of terms whose path length exceeds four. For cases where neither t_i nor t_j has an entry in the dictionary,

$$\mathrm{distance}(t_i, t_j) = \infty. \tag{6}$$

When t_i and t_j are identical, the similarity defined by (2) is 1. When we find t_i and t_j in the dictionary, and it turns out that both of them possess the same semantic attributes (e.g., a synonymous relation), the similarity becomes 0.5.

In this way, we can estimate semantic similarity of terms. However, we encounter another problem. How can we choose a semantic attribute properly when a term has some possible attributes?

Nihongo Goi Taikei considers by design that words may have multiple meanings: most nouns possess several semantic attributes. Although a human can determine correctly and immediately the meaning of a term which has a number of meanings in the context of a text, computers do not have such ability. We cannot calculate similarity of terms without identifying 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 a document. We introduce A_i to enumerate possible semantic attributes of term t_i, consulting the dictionary, *Nihongo Goi Taikei*. For example, five attributes are found for the word 'system': *organization*, *machine*, *institution*, *structure*, and *unit*.

$$\begin{aligned} t_1 &= \text{'system'},\\ A_1 &= \{\mathit{organization}, \mathit{machine}, \mathit{institution},\\ &\quad \mathit{structure}, \mathit{unit}\}\end{aligned} \tag{7}$$

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

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

In this way, the word-sense disambiguation problem is transcribed into a combinational optimization problem that decides $\{a_1, a_2, ..., a_n\}$ where $a_1 \in A_1$, $a_2 \in A_2$, ..., and $a_n \in A_n$.

For an objective of the optimization, we apply lexical chains [9] which consist of a succession of lexical cohesions (semantically related terms) and create a context and contribute to the continuity of meaning. We seek a combination of $\{a_1, a_2, ..., a_n\}$ that constructs the densest lexical chains of a document.

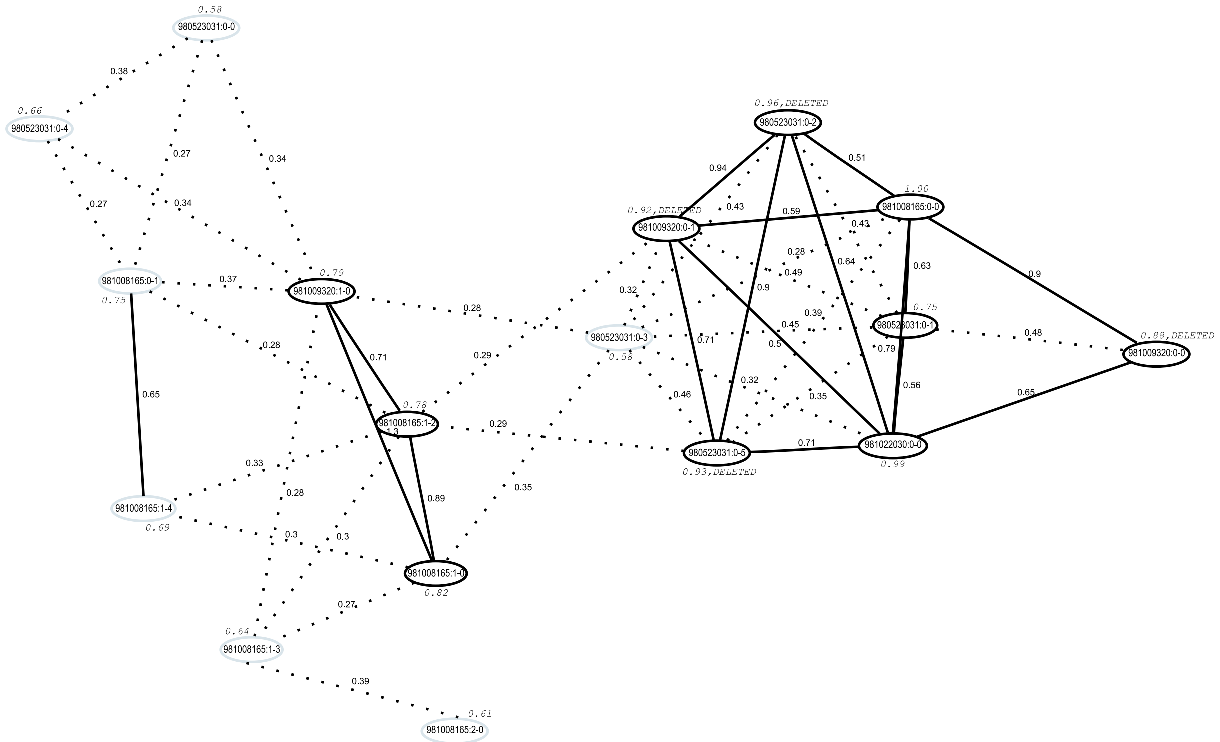

Fig. 2 A similarity network of sentences. A node is a sentence in source documents (e.g., "981008165:0-0" stands for the first sentence in the first paragraph of #165 article in the paper written on October 8th, 1998). An edge with a value represents sentence similarity by two line styles: a dotted line (similarity larger than 0.25) and a solid line (similarity larger than 0.50). A figure around the node shows its activation normalized as the maximum activation will be 1.

We define cohesiveness between two attributes a_i and a_j as the following:

$$\max\{4 - \mathrm{path\_length}(a_i, a_j), 0\}, \tag{9}$$

where $\mathrm{path\_length}(a_i, a_j)$ is the path length between attributes a_i and a_j on the semantic tree. The maximum value 4 indicates that attributes a_i and a_j are identical and have close relation. The value decreases to minimum value 0 as $\mathrm{path\_length}(a_i, a_j)$ increases to 4.

We maximize overall cohesiveness (i.e., choose a combination of $\{a_1, a_2, ..., a_n\}$, where $a_1 \in A_1$, $a_2 \in A_2$, ..., and $a_n \in A_n$) as the following:

$$\text{Maximize}$$
$$f = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \max\{4 - \mathrm{path\_length}(x_i, x_j), 0\},$$
$$\text{subject to } x_i \in A_i (i = 1, ..., n). \tag{10}$$

In other words, we determine an attribute of each term adopting the lexical chains as a context of original articles through optimization [10], [11].

3.1.2 Calculation of sentence similarity

For all pairs of sentences, we calculate similarity of sentences by the following formula,

$$\mathrm{Sim}(S_i, S_j) = \sum_{t_i \in S_i} \sum_{t_j \in S_j} \frac{\mathrm{sim(t_i, t_j)}}{\sqrt{|S_i||S_j|}}, \tag{11}$$

where $|S_i|$, $|S_j|$ are the numbers of indexing terms in sentences S_i and S_j, respectively. This formula counts all possible lexical relations in inter-sentences and normalizes the sum by the geometrical mean to satisfy similarity of the same sentences to be at least 1.

3.2 Sentence extraction by spreading activation

Finally, we rank sentences by spreading activation [12], [13] with the assumption that *"Sentences which are relevant to many ones of significance are also significant."*

First, for all pairs of sentences, we calculate similarity of sentences by formula (11). In this way, we make a network graph which indicates similarity relationship of sentences. Figure 2 is an example of a similarity network of sentences. A node represents a sentence (e.g., "981008165:0-0" stands for the first sentence in the first paragraph of #165 article in a paper written on October 8th, 1998); an edge with a value shows sentence similarity.

Then, we continue spread activation by the following formula:

$$\mathbf{A}^{(k)} = \alpha\mathbf{I} + (1 - \alpha)\mathbf{R} \cdot \mathbf{A}^{(k-1)}, \tag{12}$$

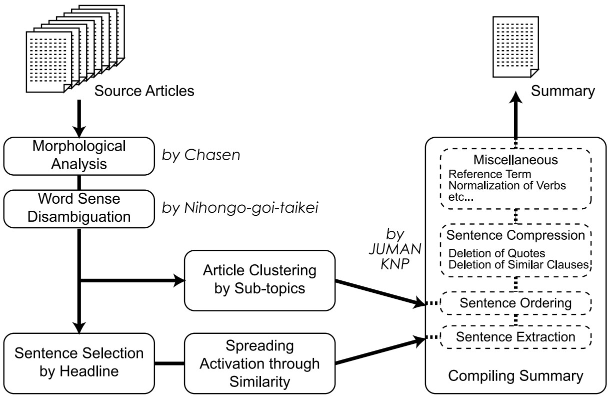

Fig. 3 Summarization system overview.

where: $\mathbf{A}^{(k)}$ is an n-vector whose element is an activation after k steps; n represents the number of nodes (sentences); $\mathbf{I}$ is an n-identity matrix; and $\mathbf{R}$ 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:

$$\mathbf{R}_{ij} = \begin{cases} \frac{\mathrm{sim}(S_i,S_j)}{\text{the number of links of } S_j} & (\text{if } i \neq j) \\ 0 & (\text{if } i = j) \end{cases} . \quad (13)$$

Finally, α is a parameter which determine activation to be inserted to the network.

In the network model, we set injection parameter α to be 0.15 and initialize $\mathbf{A}^{(k)}$ with $1/n$. Then, we apply formula (12) until convergence, normalizing $\mathbf{A}^{(k)}$ for each step to satisfy:

$$\sum_i \mathbf{A}_i^{(k)} = 1. \quad (14)$$

In this way, we can acquire a list of important sentences with the activations. The more highly a sentence is activated, the more important the sentence turns out to be.

4. Summarization system for NTCIR3-TSC2

4.1 Evaluation at NTCIR3-TSC2

To evaluate our method, we made a multi-document summarization system for Japanese newspaper articles, participating in a workshop of summarization, TSC2 (Text Summarization Challenge) task [15] organized by NTCIR-3 project [14]. TSC2 used Mainichi Newspaper articles as a source document set. For more information about NTCIR project and TSC task, refer to the workshop proceedings [14], [15].

4.2 System overview

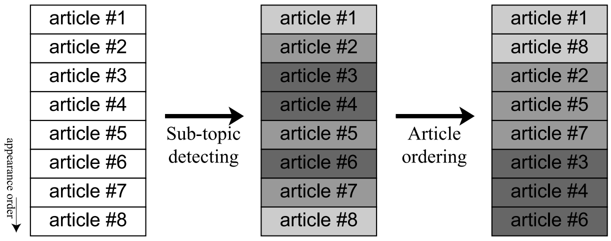

Fig. 4 The strategy of sentence (article) ordering. There are three sub-topics detected (half-tone dot meshed) in these articles.

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.

Fig. 5 A typical example of duplication (rough English translation). The boldface clause is a repeated expression to be deleted.

We participated in a multi-document summarization task to inquire into behavior (i.e., how much this method encompasses important contents of the original articles; or how much this method includes redundant contents) when source documents have a sufficient amount of similar sentences. We introduce some other components (Fig. 3) to build a multi-document summarization system.

We employ a headline heuristic to catch major points of each article. Because it is manifested that articles for summarization are retrieved from the only one news source in TSC2, headline words are not only representative of the article, but also something characteristic to the article. We extract all sentences which contain a term occurring in the headline of each article; this is equivalent to a process of passing over those which are irrelevant to the thrust. A spreading activation algorithm is applied to candidate sentences by this phase.

In addition to extracting sentences, we consider sentence ordering to improve summary readability by detecting sub-topics within the original articles. Since we can find some sub-topics in source documents that are collected for some topic, we should order extracted sentences along sub-topics to improve overall quality of summary [17]. We assume a newspaper article to be written for one topic; we apply the nearest neighbor method [18] for document vectors whose element represents term frequency. After classifying articles by their sub-topics, we order the extracted sentence so as not to lose the thread of the argument (Fig. 4).

We also try to eliminate redundancy within a sentence to improve the compression ratio. We acquire a set of key sentences by extracting highly activated sentences up to a specified summarization length. This can be a good summary which centers on several key

$^{980523031:0-1}$ **大学入試センター試験の出願受け付けは今年１０月９日から同２１日まで。**$^{980523031:0-2}$ **試験は来年１月１６、１７の両日。**$^{980523031:0-5}$ 試験日は２０００年１月１５、１６の両日とした。 $^{981008165:0-0}$ **来年１月１６、１７日に実施される１９９９年度大学入試センター試験の願書受け付けが９日から東京・駒場の大学入試センターで始まる。**$^{981008165:1-0}$ **試験を利用するのは国立９５校、公立６１校と私立２１７校の計３７３校。**$^{981008165:1-2}$ **私立全体の４９%が試験を利用することになる。**受験シーズンの幕開けとなる１９９９年度大学入試センター試験の願書受け付けが９日、東京・駒場の大学入試センターで始まった。$^{981009320:0-1}$ 試験は来年１月１６、１７日に行われる。 $^{981009320:1-0}$ **センター試験を利用する大学は前年度より４１校増える。**$^{981022030:0-0}$ **１９９９年1月１６、１７の両日に行われる大学入試センター試験の願書受け付けが２１日、締め切られた。**

A summary in Japanese

$^{980523031:0-1}$ **An application of National Center Test for University Admissions will be received from Oct. 9 to 21 this year.** $^{980523031:0-2}$ **Students will have the test on Jan. 16 and 17.** $^{980523031:0-5}$ Students will have the test on Jan. 15 and 16, 2000. $^{981008165:0-0}$ **On the 9th, Natinal Center for University Entrance Examination at Komaba will begin to receive applications of National Center Test for University Admissions held on Jan. 16 and 17, 1999.** $^{981008165:1-0}$ **95 national, 61 public and 217 private universities, 373 universities in total, use the test.** $^{981008165:1-2}$ **This figure shows that 49% of public universities use the test.** $^{981009320:0-0}$ On the 9th, Natinal Center for University Entrance Examination at Komaba started to receive applications of National Center Test for University Admissions; the season of entrance exam in 1999 came. $^{981009320:0-1}$ Students will have the test on Jan. 16 and 17 next year. $^{981009320:1-0}$ **The number of universities that use National Center Test is 41 larger than those in the last year.** $^{981022030:0-0}$ **On the 21th, an application of National Center Test for University Admissions held on Jan. 16 and 17 in 1999 was closed.**

A rough translation in English

Fig. 6 An example of extracted sentences (in Japanese and rough English). The source document is written about National Center Test for University Admissions. Only sentence in boldface are actually included into the summary; other sentences are deleted by similar clause elimination.

points. However, this may also lead to extraction of a set of sentences which may contain many redundancies. Related newspaper articles often contain a pair of sentences like those in Fig. 5, which have a lot in common, but describe slightly separate subjects. In order to reduce redundancy, breaking up each sentence into several textual units (or clauses), we delete units which are similar to previously-included content.

For more details about the summarization system, refer to a paper [16] in the Workshop proceedings.

5. Evaluation

5.1 An example of activations and summaries

Figure 6 is a extracted sentences to generate a summary within 250 characters from the similarity network

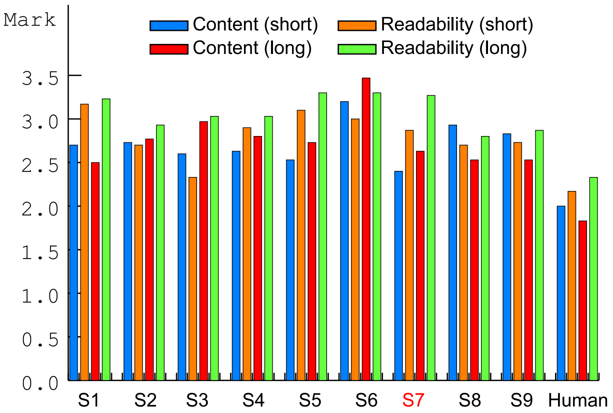

Fig. 7 Subjective evaluation by ranking. Sx stands for "System #x" and ours is S7. A lower mark is better.

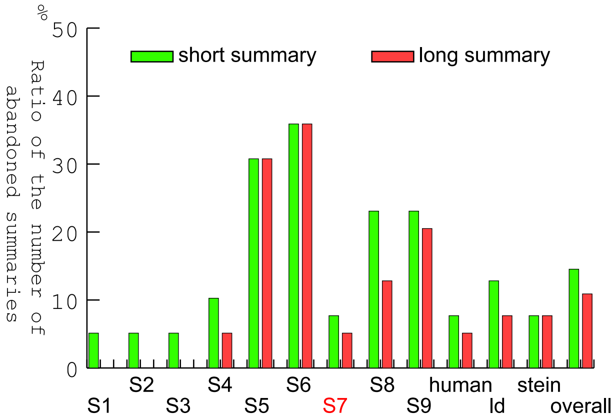

Fig. 8 The number of abandoned summaries to revise. Sx stands for "System #x" and ours is S7.

shown in Fig. 2; the source articles (four articles) report on application of National Center Test for University Admissions. Figures around the nodes in Fig. 2 show their activations; deep black nodes (sentences) are actually extracted to generate the summary.

We can see some clusters in Fig. 2: three clusters are found when we identify clusters by solid lines. Even though we use no method of clustering to choose important sentences, we can see extracted sentences belonging to either of the two clusters.

As expected, spreading activation through sentence similarity tends to focus on several key points (e.g., the application date, schedule of National Center Test, and the number of universities that use the test), but it often includes a pair of sentences that present the same thing. It is necessary to cut down redundant expression after extracting importance sentences; sentences not in boldface are actually deleted as a result of similar clause elimination.

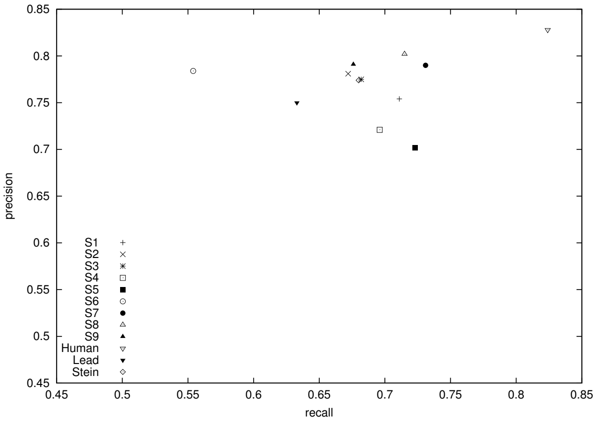

Fig. 9 Precision-recall like evaluation for short summaries.

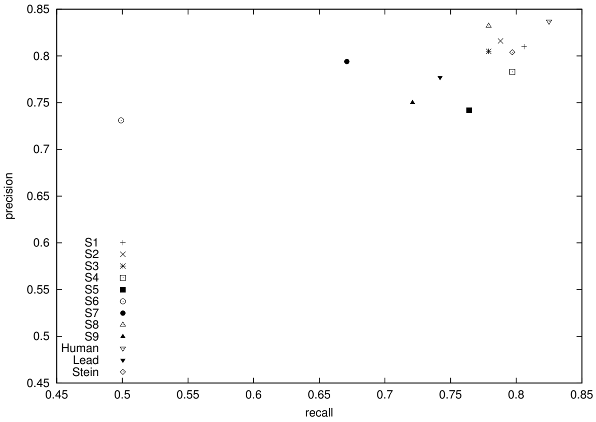

Fig. 10 Precision-recall like evaluation for long summaries.

5.2 Evaluation by rating

Figure 7 shows a rating made by human judges.[†] Our system, shown as S7, is rated: 2.40 (content of shorter summary); 2.87 (readability of shorter summary); 2.63 (content of longer summary); and 3.27 (readability of longer summary).[††] Our summary got a favorable impression from readers, contending for first place especially in terms of content of shorter summaries. In contrast to shorter summaries, we can see that content of longer summaries degrades to an average; and readability of longer summaries is worse than average.

5.3 Evaluation by revision

Our summaries were revised by human correctors[†] to examine the tendency of our summary. The correctors can give up revising a summary in case it is far from an acceptable one. The number of abandoned summaries can be seen in Fig. 8. The ratio of rejection is about 7 or 8%: equal to that of humans and slightly better than these of baseline methods (denoted by 'lead' and 'stein' [19]). It turns out that our summary was acceptable for readers.

We evaluate our method by precision-recall-like metrics from human's revision. Figures 9 and 10 are precision-recall like evaluations of each summarization length. We define precision and recall as follows:

$$\text{precision} = 1.0 - (\text{sum of deletion ratio}) \tag{15}$$

$$\text{recall} = 1.0 - (\text{sum of insertion ratio}) \tag{16}$$

Precision and recall in this definition differ from usual usage in two aspects. Conventional precision-recall metrics is not suitable for summarization because it assumes there exists only one correct answer for a summary. On the other hand, this metrics method virtually requires human's corrector to prepare a correct answer for each summary in his or her mind. Another aspect is that deletion or insertion ratios are not given to abandoned summaries. The more summaries of a system the corrector abandons, the lower the effective precision and recall may be; it has been estimated that the deletion and insertion ratio of abandoned summaries has been very high.

Figure 9 shows that our system is one of the best for short summary. As might be expected, our summary is worse than a human summary, but it is better than baseline methods. On the other hand, ours does not seem to perform well for longer summaries (Fig. 10); it is worse than the stein method and rivals the lead method.

Particularly, the recall of a long summary is lower than that of a short summary while precision keeps up. Degression of recall shows that our method is prone to including similar content and disregarding something unusual; readers wanted to get some other sentences (i.e. contents) as they approved that sentences in the summaries had their existence values. Limitation of space in shorter summary leads us to disregard this bad tendency since summaries with a few centers are sufficient. Compared to this situation, at longer summary, it is expected that it includes not only a few centers, but more peripheral sentences.

6. Conclusion

We introduced a novel summarization method that ranks sentences by spreading activation through sentence similarity in order to archieve a comprehensive summary. Our method is proven effective for a short

[†] TSC asked human judges to evaluate and rank system summaries on a 1 to 4 scale (1 is best, and 4 is worst) in terms of content (How much a system summary encompasses important contents of the original articles) and readability (How readable the system summary is).

[††] A short summary is only half the length of a long summary.

[†] Correctors read the original texts and revise system summaries in terms of content and readability. Revision is restricted to three editing operations, insertion, deletion, and replacement.

summary, but future work to improve the recall for a long summary remains. A summary as reading material should be not only a collection of major points, but a well-formed text. Hence, the reader prefers a shorter summary to a longer summary since it requires little burden to read a short summary. Our method will match well when readers want a short summary in the form of a text.

Acknowledgement

We used Mainichi Newspaper articles and Summarization Task Data, participating in a competition of summarization, TSC (Text Summarization Challenge) task organized by NTCIR-3 project. We also used Chasen as a morphological analyzer, JUMAN as a morphological analyzer, and KNP as a syntactic parser. We wish to thank reviewers for furnishing useful comments.

References

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

[2] H. P. Luhn, “The automatic creation of literature abstracts”, *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 159–165, 1958.

[3] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of are Information by Computer.* Addison-Wesley, 1989.

[4] H. P. Edmundson, “New methods in automatic extracting”, *Journal of the Association for Computing Machinery*, 16(2), pp. 264–285, 1969.

[5] I. Mani and E. Bloedorn, “Summarizing similarities and differences among related documents”, *Information Retrieval*, Vol. 1, No. 1–2, pp. 35–67, 1999.

[6] K. Nagao and K. Hashida, “Automatic text summarization based on the Global Document Annotation”, Proc. 17th International Conf. on Computational Linguistics (COLING-ACL ’98), Montreal, Quebec, Canada, Aug. 1998.

[7] J. Fukumoto, “Extraction of sentences from a Japanese textusing inter-sentential connectivity”, PACLING ’97, Japan. 1997.

[8] G. Salton, A. Singhal, M. Mitra, and C. Buckley, “Automatic text structuring and summarization”, *Information Processing and Management*, Vol. 32, No. 2, pp. 53–65, 1997.

[9] J. Morris and G. Hirst, “Lexical cohesion computed by thesaural relations as an indicator of the structure of text”, *Computional Linguistics*, Vol. 17, No. 1, pp. 21–48, 1991.

[10] M. Okumura and T. Honda, “Word sense disambiguation and text segmentation based on lexical cohesion”, Proc. of COLING-94, Vol. 2, pp. 75–761. 1994.

[11] R. Barzilay and M. Elhadad, “Using lexical chains for text summarization”, Proc. of The ACL/EACL Workshop on Intelligent Scalable Text Summarization, pp. 10–17, 1997.

[12] A. Collins and E. Loftus, “A spreading activation theory of semantic processing”, *Psychological Review*, Vol. 82, pp. 407–428, 1975.

[13] S. Brin and L. Page, “The anatomy of a large-scale hypertextual Web search engine”, *Computer Networks and ISDN Systems*, Vol. 30, No. 1–7, pp. 107–117, 1998.

[14] N. Kando, “Overview of the Third NTCIR Workshop”, In working notes of the 3rd NTCIR Workshop Meeting, Part. I: Overview, pp. 1–16, National Institute of Informatics, Tokyo, Japan. Oct. 2002.

[15] T. Fukushima and M. Okumura, “Text Summarization Challenge 2: Text summarization evaluation at NTCIR Workshop 3”, In working notes of the 3rd NTCIR Workshop Meeting, Part. I: Overview, pp. 87–91, National Institute of Informatics, Tokyo, Japan. Oct. 2002.

[16] N. Okazaki, Y. Matsuo, N. Matsumura, H. Tomobe, and M. Ishizuka “Text Summarization Challenge 2 Text summarization evaluation at NTCIR Workshop 3”, In working notes of the 3rd NTCIR Workshop Meeting, Part. V: Text Summarization Challenge 2 (TSC2), pp. 39–46, National Institute of Informatics, Tokyo, Japan. Oct. 2002.

[17] R. Barzilay, N. Elhadad and K. McKeown, “Inferring Strategies for Sentence Ordering in Multidocument Summarization”, *Journal of Artificial Intelligence Research*, Vol. 17, pp. 35–55, 2002.

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

[19] G. C. Stein, T. Strzalkowski and G. B. Wise, “Interactive, Text-Based Summarization of Multiple Documents”, *Computational Intelligence*, Vol. 16, No. 4, pp. 606–613, 2000.

Naoaki OKAZAKI received his B.S. and M.S. in Information and Communication Engineering from the University of Tokyo in 2001 and 2003. He is a Ph. D. candidate at Graduate School of Information Science and Technology, the University of Tokyo at present. His research interests include text mining and natural language processing.

Yutaka MATSUO is a researcher at Cyber Assist Research Center, National Institute of Advanced Industrial Science and Technology. He received his B.S. and Ph.D. in Information and Communication Engineering from University of Tokyo in 1997 and 2002. His research interests include automated inference, natural language processing and information theory.

Naohiro MATSUMURA received his B.S. and M.S. in Engineering Science from Osaka University in 1998 and 2000, and Ph.D. in Electronic Engineering from the University of Tokyo in 2003. Currently, he is a postdoctoral fellow at the Dept. of Information and Communication Engineering, School of Information Science and Technology, the University of Tokyo. His research interests include community mining, text mining, natural language processing, and chance discovery.

Mitsuru ISHIZUKA is a professor at the Dept. of Information and Communication Eng., School of Information Science and Thechnology, the Univ. of Tokyo. Prior to this position, he worked at NTT Yokosuka Lab., the Institute of Industrial Science, the Univ. of Tokyo, and Purdue Univ.(visiting assoc. prof.). He earned his B.S., M.S. and Ph.D. in electronic engineering from the Univ. of Tokyo. His research areas include artificial intelligence, Internet/WWW intelligence, and multimodal lifelike agents. He is a member of IEEE, AAAI, IPS Japan, Japanese Society for AI, etc.