Sentence Extraction by Spreading Activation with Refined Similarity Measure

Naoaki Okazaki and **Yutaka Matsuo*** and **Naohiro Matsumura** and **Mitsuru Ishizuka**
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract

Although there has been a great deal of research on automatic summarization, most methods are based on a statistical approach, disregarding relationships between extracted textual segments. To ensure sentence connectivity, we propose a novel method to extract a set of comprehensible sentences that centers on several key points. This method generates a similarity network from documents with a lexical dictionary and applies spreading activation to rank sentences. Also, 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 (NII-NACSIS Test Collection for IR Systems) project.

Introduction

Information pollution driven by computerized documents presents to the 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.

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 consists of a set of claused sentences or phrases. When readers are content with itemization of essential parts, we should generate a summary to give widely and shallowly a panoramic view of an original text. Since such claused sentences or phrases give fragmentary information, we should perform myriad processes (e.g., clustering and ordering items) to elucidate relationships among claused textual units.

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. If the summary is not well-organized, they may find it very hard to read; as a worst case, they may lose their 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 minimum revision.

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

Summarization toward comprehensible text

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 (Luhn 1958; Salton 1989), cue phrases (Edmundson 1969), titles (Edmundson 1969), or sentence location (Edmundson 1969).

However, extraction by statistical methods disregards relationships between extracted textual units (i.e., terms, sentences or 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. (Mani & Bloedorn 1999) proposed a summarizing method based on a graph representation of related documents. By exploiting meaningful relations between units based on an analysis of text cohesion and 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 an unusually deep analysis of an original text.

Nagao, et. al. (Nagao & Hasida 1998) proposed a similar approach. However, their approach uniquely introduces GDA (Global Document Annotation). Through use of an intra-document network, in which nodes correspond to terms and link the semantic relations which are defined naturally by a GDA tagged document, spreading activation is performed in the network. It generates summary sentences directly from the semantic network, adding highly activated elements into resultant summary. It may be an effective method if GDA-tagged documents are given.

Salton, et. al. (Salton *et al.* 1997) suggest a passage extraction from a document based on *intra-document* links between passages. It generates intra-document links from similarity of passage vectors. Once a similarity network is gen-

*Now at Cyber Assist Research Center, AIST Tokyo Waterfront, 2-41-6 Aomi, Koto-ku, Tokyo 135-0064, Japan

erated, the method chooses important passages by judging from the *bushiness* [1] of a node (passage), depth-first path, and segmented bushy path in the network.

Fukumoto (Fukumoto 1997) proposed a method that first chooses sentences that contain a query term of user input and sentences which have a strong similarity to the previously selected sentences. As it decides to extract sentences one-by-one by comparing similarity, it does not consider the 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 an 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. As with Salton's method (Salton *et al.* 1997), it uses simple vector cosine distance for measuring sentence similarity; it neglects synonym relations.

Proposed Method

Against the background of these studies, we propose a novel extraction method that ranks sentences by spreading activation with the assumption that *"Sentences which are relevant to ones of significance are also significant"*. It produces a comprehensive summary even when a reader requires a short summary. Our method differs from some studies e.g., (Mani & Bloedorn 1999; Nagao & Hasida 1998) in that ours ranks sentences directly by spreading activation through sentence similarity; it does not require a deep analysis of original text. Our method also differs from others (Salton *et al.* 1997; Fukumoto 1997) by introducing refined similarity measure of sentences.

Sentence Similarity

Sentence extraction by spreading activation, as we detail later, requires similarity of sentences. Sentence similarity can be calculated from lexical relations between terms appearing in a sentence and others. When we estimate similarity of sentences, we must consider three problems: *how to estimate similarities of terms*; *how to identify the meaning of terms*; and *how to calculate sentence similarity from them*.

Estimation of term similarity For estimating similarity of terms, we use a Japanese lexical dictionary, *Nihongo Goi Taikei* [2] to take synonyms or other relations into consideration. Examining the semantic tree carefully, we notice that the number of terms that exist along the path from one term to another increases exponentially in proportion to path length. 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 terms, t_i and t_j, by the exponential function,

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

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

When t_i and t_j are identical, we define $\mathrm{distance}$ to be 0; $\mathrm{sim}(\mathrm{t_i}, \mathrm{t_i})$ will be 1, consequently. In cases where t_i and t_j are not identical, introducing a_i and a_j to represent attributes to which term t_i and t_j belong respectively, we define $\mathrm{distance}$ as the following.

$$\mathrm{distance}(t_i, t_j) = \begin{cases} \mathrm{length_p}(a_i, a_j) + 1 & (length < 4) \\ \infty & (length \geq 4) \end{cases} \quad (2)$$

$\mathrm{length_p}(a_i, a_j)$ is the path length between nodes $\#a_i$ and $\#a_j$ on the semantic tree. In case either t_i or t_j has no entry in the dictionary, $\mathrm{distance}$ is defined as ∞.

Sense disambiguation of terms 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 can not 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, for a word 'system', five attributes are found: #362 (organization), #962 (machine), #1155 (institution), #2498 (structure), #2595 (unit).

$$t_1 = \text{'system'}, A_1 = \{362, 962, 1155, 2498, 2595\}. \quad (3)$$

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

Then, we choose a combination of $a_i \in A_i$ (i.e. search optimal $\{a_1, a_2, ..., a_n\}$, where $a_1 \in A_1$, $a_2 \in A_2$, ..., and $a_n \in A_n$) so that it maximizes the following $score$,

$$score = \sum_{k=1}^{n} \sum_{l=k+1}^{n} \min\{4 - \mathrm{distance}(a_k, a_l), 0\}, \quad (4)$$

where $\mathrm{distance}(a_i, a_j)$ is the same as in Equation (2). In other words, we determine an attribute of each term adopting lexical cohesion as context of original articles through optimization (Okumura & Honda 1994).

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

$$\mathrm{Sim}(S_i, S_j) = \sum_{t_i \in S_i} \sum_{t_j \in S_j} \frac{\mathrm{sim(t_i, t_j)}}{\sqrt{|S_i||S_j|}}, \quad (5)$$

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

[1] The bushiness of a node on a graph is defined as the number of links connecting it to other nodes on the graph.

[2] NTT Communication Science Laboratories, Iwanami Shoten. *Nihongo Goi Taikei* 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 that represent semantic attributes. Because the tree has the property that a node connotes semantic attributes of descendant nodes, we can estimate similarity of terms by the distance between terms on the semantic tree.

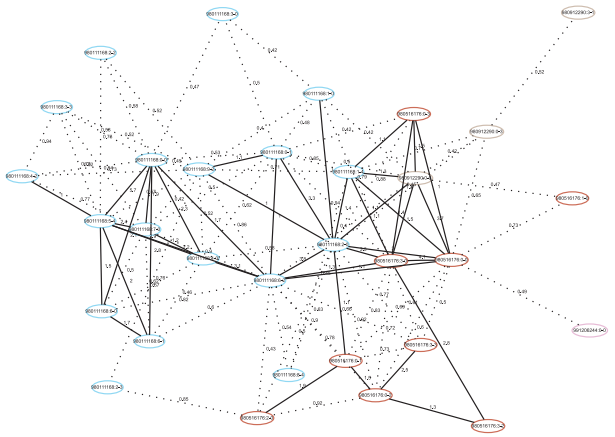

Figure 1: Similarity network of sentences.

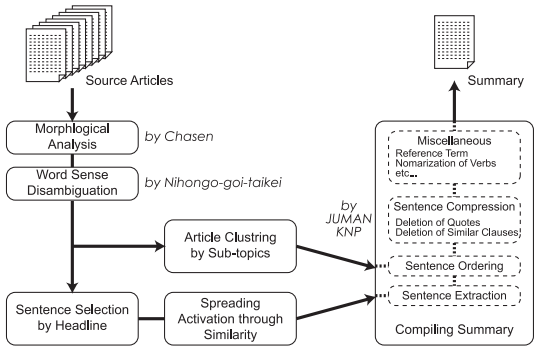

Figure 2: Summarization system overview.

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 3: A typical example of duplication (rough English translation). The boldface clause is a repeated expression.

Sentence extraction by spreading activation

Finally, we rank sentences by spreading activation (Collins & Loftus 1975) with an assumption that *"Sentences which are relevant to ones of significance are also significant."*

First, we link a pair of sentences S_i and S_j if $\mathrm{Sim}(S_i, S_j) > 0$ to make a network graph, which indicates similarity relationship of sentences. Figure 1 is an example of similarity network of sentences. A node represents a sentence [3], and an edge with a value shows similarity of sentences.

Then, we continue spreading activation by the following formula.

$$\mathbf{A}^{(k)} = \alpha\mathbf{I} + (1-\alpha)\mathbf{R}\cdot\mathbf{A}^{(k-1)} \qquad (6)$$

$\mathbf{A}^{(k)}$ is a n-vector whose element is an activation after k steps; $\mathbf{I}$ is a 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:

$$\mathbf{R}_{ij} = \begin{cases} \dfrac{\mathrm{sim}(S_i,S_j)}{\text{the number of links of } S_j} & (\text{if } i \neq j) \\ 0 & (\text{if } i = j) \end{cases} . \qquad (7)$$

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

In the network model, we set injection parameter α to be 0.15 and initialize $\mathbf{A}^{(0)}$ with a given value. Then, we apply the formula (6) until convergence, normalizing $\mathbf{A}^{(k)}$ for each step to satisfy the sum of activations to be 1. In this way, we can acquire a list of important sentences with their activations. The more a sentence is activated highly, the more important the sentence turns out to be.

Implementation

To ensure effectiveness of our method, we made a multi-document summarization system (Figure 2) for Japanese newspaper articles, participating in a competition of summarization, TSC (Text Summarization Challenge) task [4] organized by NTCIR-3 (NII-NACSIS Test Collection for IR Systems) project [5]. We participated in a multi-document summarization task. A summary made by gathering summaries of each document has an adverse consequence that it will contain some redundant expressions or lack some important passage (McKeown *et al.* 1999). To build a multi-document summarization system, we introduce some other components.

Sentence selection by headline We extract all sentences which have one or more terms with relation to a term occurring in the headline of each article. It 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.

Eliminating similar clauses We can acquire a set of key sentences by extracting highly activated sentences up to a specified summarization length. Although this can be a good summary which centers on several key points, this may also lead to extract a set of sentences which may contain many redundancies. Related newspaper articles often contain a pair of sentences like those in Figure 3, which has a lot in common but describes slightly separate subjects. In order to eliminate such repeated expressions, breaking up each sentence into several textual units (or clauses), we delete units

[3] "980111168:0-0" stands for the first sentence in the first paragraph of article #168 in a paper written on January 11, 1998

[4] http://lr-www.pi.titech.ac.jp/tsc/index-en.html

[5] http://research.nii.ac.jp/ ntcadm/index-en.html

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 4: An example of quote deletion (rough English translation). The boldface segment is to be deleted.

to be considered as redundant. We use KNP [6] for identifying clause-like units in a sentence and delete units which are similar to previously-included content.

Deletion of quotes A newswriter quoting someone in an article will append a summary after someone's long statement in a sentence like Figure 4. We recognize a quotational clause which begins at the open quote and which ends at the closing quote or its successive adverb phrase to compress such sentences by blacking out the section concerning the quotational clause.

Sentence ordering by clustering articles We can find some sub-topics in documents collected for some topic. In such case, we should order extracted sentences along the sub-topics to improve overall summary quality (Barzilay, Elhadad, & McKeown 2002).

We can assume a newspaper article to be written for one topic. Hence, to classify sub-topics in a summary, we classify articles by their topics. We apply the nearest neighbor method (Cover & Hart 1967) for clustering after measuring cosine distance between two article vectors whose element is term frequency. We merge a pair of clusters when their minimum distance is lower than 0.4. After classifying the articles by their sub topics, we order the extracted sentences so as not to lose the thread of the argument.

Evaluation

After participants in TSC2 send their summaries to TSC, TSC evaluates summaries in a common way and returns evaluation results. TSC2 evaluation of summaries is done by two intrinsic methods, using summaries prepared by humans as reference data for evaluation. In the formal run, 30 topics (sets of articles) were assigned for summarization with two specified lengths (long and short). [7]

Evaluation by ranking

The first evaluation is done by ranking participating systems (summaries). They ask human judges, who are experienced

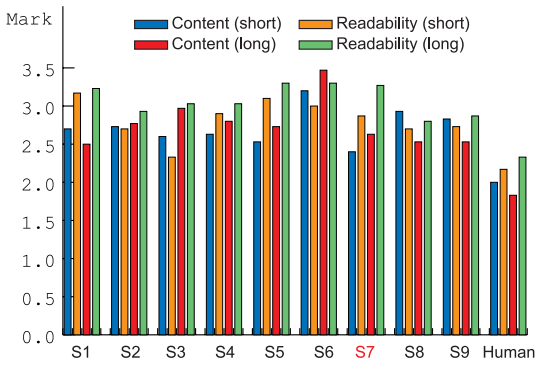

Figure 5: Subjective evaluation by ranking. Sx stands for "System #x"; ours is S7. Lower mark is better.

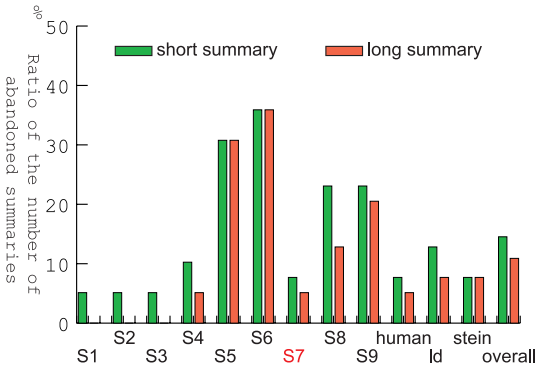

Figure 6: The number of abandoned summaries to revise. Sx stands for "System #x"; ours is S7.

in producing summaries, to evaluate and rank system summaries on a 1 to 4 scale (1 is the best, 4 is the worst) in terms of *content*, and *readability*.

Figure 5 shows evaluation results of summaries made by participating systems (S1–S9) and a human. Our system is shown as S7. The ranking of humans implies the upper bounds of the evaluation. It is shown that our summary got a favorable impression from readers. Our system contended for the first place especially in terms of content of the shorter summary.

Evaluation by revision

The second evaluation is done by measuring revision degree to summaries. Correctors read the original texts and revise system summaries in terms of content and readability. The revision are restricted to three editing operations, insertion, deletion, and replacement. 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 Figure 6. The ratio of rejection is about 8%, equal to that of humans. It turns out that our summary is acceptable for readers.

We evaluate our method by precision-recall-like metrics from the evaluation by human's revision as well. Figures 7

[6] Japanese syntactic parser by Language Media Laboratory, Graduate School of Informatics, the University of Kyoto.

[7] Short summary is only half the length of a long summary.

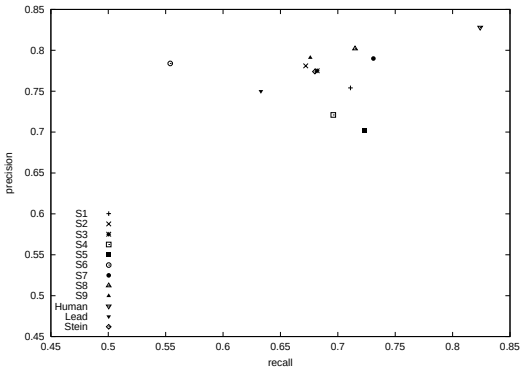

Figure 7: Precision-recall-like evaluation (short summaries).

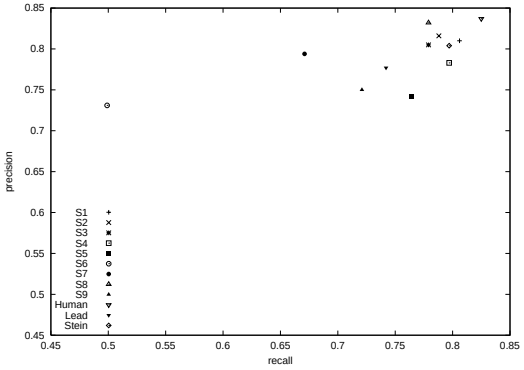

Figure 8: Precision-recall-like evaluation (long summaries).

and 8 are precision-recall-like evaluations of each summarization length. Precision and recall in this evaluation are defined as follows: [8]

$$\text{precision} = 1.0 - (\text{sum of deletion ratio}) \quad (8)$$

$$\text{recall} = 1.0 - (\text{sum of insertion ratio}) \quad (9)$$

Sum of deletion ratio denotes how many letters are deleted in the process of revision and the sum of insertion does so correspondingly.

Figure 7 shows that we can see that our system takes one of the leads for short summary. For the long summary (Figure 8), on the other hand, ours seems to perform poorly, especially owing to recall. This shows it is prone to including similar content and disregarding something unusual. One of the main reasons is precision of activation degrades to no appreciable difference as we pick up more sentences. Limitation of space at a shorter summary leads us to disregard this bad habit since summaries with a few centers are enough. Compared to this situation, a longer summary is expected to include not only a few centers, but more key points.

[8] Strictly speaking, they differ 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 precision and recall may be because it has been estimated that the deletion and insertion ratio of abandoned summaries has been very high.

Conclusion

We introduced a novel summarization method that ranks sentences by spreading activation with refined similarity measure of sentences in order to archive a comprehensive summary. Although future work remains to improve the recall for the long summary, it is proven that our method is effective for the short summary. Our method will match well when readers want a short summary in 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.

References

Barzilay, R.; Elhadad, N.; and McKeown, K. 2002. Inferring strategies for sentence ordering in multidocument summarization. *Journal of Artificial Intelligence Research* 17:35–55.

Collins, A., and Loftus, E. 1975. A spreading activation theory of semantic processing. *Psychological Review* 82:407–428.

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

Edmundson, H. P. 1969. New methods in automatic extracting. *Journal of the Association for Computing Machinery* 16(2):264–285.

Fukumoto, J. 1997. Extraction of sentences from a Japanese text using inter-sentential connectivity. In *Proc. of PACLING'97*.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM journal of Research and Development* 2(2):159–165.

Mani, I., and Bloedorn, E. 1999. Multi-document summarization by graph search and matching. In *Proc. of AAAI-97*, 622–628.

McKeown, K.; Klavans, J.; Hatzivassiloglou, V.; Barzilay, R.; and Eskin, E. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proc. of 16th National Conference on Artificial Intelligence*, 453–460.

Nagao, K., and Hasida, K. 1998. Automatic text summarization based on the Global Document Annotation. In *Proc. of COLING-ACL '98*.

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

Salton, G.; Singhal, A.; Mitra, M.; and Buckley, C. 1997. Automatic text structuring and summarization. *Information Processing and Management* 32(2):53–65.

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