Similarity Calculation of Segment Retrieval for Aid in reading Related Documents

Tatsunori Mori and **Shun'ichi Tanaka** and **Hiroshi Nakagawa**
Division of Electrical and Computer Engineering,
Yokohama National University
79-5 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, JAPAN
{mori,tanashun}@forest.dnj.ynu.ac.jp, nakagawa@naklab.dnj.ynu.ac.jp

Abstract

It is difficult to appropriately retrieve the documents which are truly needed by users at that moment, because we have a lot of available documents recently.

In this paper, we propose a new similarity calculation method among parts of documents. The calculation is the main part of the system which allows readers of documents to obtain related parts of other documents.

We introduce two types of information, the co-occurrence of words and the lexical chain of words, to improve accuracy of the baseline method of similarity calculation, which is based on the $tf \cdot idf$ method and the vector space model. We also shows that it is effective to combine those types of information.

According to our experiment, in which the similarity among segments of related manuals are calculated, our methods outperforms the baseline $tf \cdot idf$ method in terms of the precision values at the same recall rate.

1 Introduction

Document retrieval gives us a good starting point to read documents relevant to a topic among a very large document base. However some other ways of reading aid are required when we read a particular document.

For example, suppose that a user wants to obtain other information related to the part of document he/she is reading. Conventional information retrieval systems are not suitable for the need because they recommend not a relevant part of document but a whole document relevant to the topic. Even if passage retrieval systems are used, users have to input troublesome requests, or queries. Systems suitable for such a task should have abilities like the function to retrieve parts of documents segment by segment, the capability to make links among related segments, and so forth. Suppose that the user wants to read only the segments related to the current part of document. For such a user, the system should be a useful reading aid if the system has the retrieval performance enough to get only truly relevant parts of documents, because the user can concentrate his/her attention on the current document and a small amount of supplementary explanations retrieved.

Based on the discussion described above, we propose the methods to improve the performance of segment retrieval from a set of documents. Here, the term *segment* represents a certain unit of small part of document like a section, a subsection and so forth, and accordingly one document consists of a series of segments. The phrase *current segment* denotes the segment which the user is reading now.

The aim of segment retrieval is to obtain the other segments which are *similar* to the current segment. The *similarity* is usually measured as the degree of overlap of segments from the viewpoint of the contents. Therefore we suppose that the relevant information can be obtained by looking around in other documents where the similar segments are located. Note that users may not obtain new information from such *similar* segments themselves, because they may have the almost same contents as the original one. Even in such a case, however, new relevant information should be gotten from the adjacent segments.

Computing the degree of overlap in terms of semantic representations is one of the methods to calculate similarity. However, the cost of the semantic interpretation is very high and the semantic representations may have some kind of ambiguity. Therefore we adopt the method based on the statistics of words, namely the $tf \cdot idf$ method and the vector space model, which are usually used in the conventional information retrieval(Salton and Buckley, 1988; Salton et al., 1975). Unfortunately, the $tf \cdot idf$ method does not have enough performance we needed. Our task requires that the relevant segments should surely appeared as higher ranks in the similarity calculation method. Namely, the result of ranking by the method should have high precision in the low-recall area of the recall-precision curve. Accordingly, we introduce the information of word co-occurrence and lexical chains to improve the performance of similarity calculation. We also propose the method to combine information of them.

2 Baseline method

Our basic system is illustrated in Figure 1. Firstly, the term extraction module analyzes

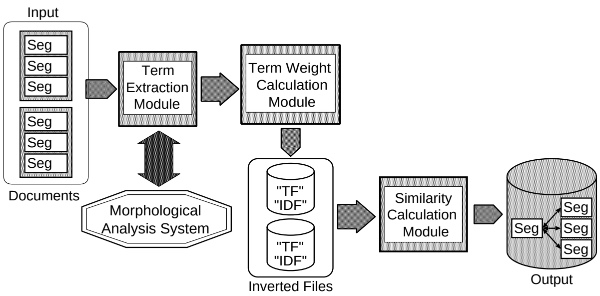

Figure 1: Basic system

the input documents morphologically to extract content words like nouns, verbs(Matsumoto et al., 1996). Secondly, the term weight calculation module computes $tf \cdot idf$ values for each term. Here, the tf value of each term is the term frequency in a segment and the idf value is the logarithm of inverse of the term's segment frequency. Thirdly, the similarity calculation module computes the similarity of each pair of segments based on the vector space model. In the model, a segment is represented as a vector in a vector space, each of whose dimension corresponds to each term. A vector's value of each dimension is the $tf \cdot idf$ value of the term. We call this method *the baseline method*, or *B-method* for short, hereafter.

3 Improvement of the performance of baseline method using the information of relation among terms

Because our system treats the retrieval of segments, which are parts of documents, we can take account of not only the information of terms within a segment but also the information over more than one segment.

In this section, we propose introducing the following two types of the relations among terms to improve the performance of similarity calculation.

- Co-occurrence of two terms in a sentence, as the information of intra-segment.
- Lexical chain (reputation) of a term, as the information of inter-segment.

Roughly speaking, those types of information can be used to increase the weight of important terms.

3.1 Utilization of Term Co-occurrence

Takaki et al.(Takaki and Kitani, 1996) introduce *the importance score of co-occurrence of term pairs*, *cw*, and show that the score contributes to improve the IR performance of Japanese newspaper articles. We adopt the

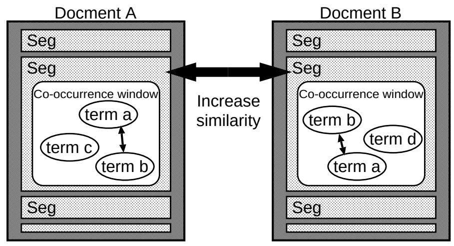

Figure 2: Use of co-occurrence of term pairs in similarity calculation

score cw with some minor modifications. When term pairs appear in two segments d_A and d_B, the similarity of the segments is increased by adjusting the tf values of terms in the term pairs.

Suppose the term t_k occurs f times in the segment d_A. The modified tf value, $tf'(d_A, t_k)$, is calculated from the original tf value, $tf(d_A, t_k)$, as the following formula:

$$tf'(d_A, t_k) = tf(d_A, t_k) + \sum_{p=1}^{f} \sum_{t_c \in T_c(t_k, p, d_A, d_B)} cw(d_A, t_k, p, t_c) \qquad (1)$$

where the variable p denotes the p-th occurrence of t_k in the segment d_A, and $T_c(t_k, p, d_A, d_B)$ is a set of terms which co-occur with the p-th t_k.

Here, we formally define *the co-occurrence of two terms* in terms of the distance between the terms. If the distance of two terms is less or equal to a threshold, the terms are regarded as in co-occurrence.

The value of cw is defined by the following formula:

$$cw(d_A, t_k, p, t_c) = \frac{\alpha(d_A, t_k, p, t_c) \cdot \beta(t_k, t_c) \cdot \gamma(t_k, t_c)}{M(d_A)} \quad (2)$$

where $\alpha(d_A, t_k, p, t_c)$ is the function expressing how near t_k and t_c occur. $\beta(t_k, t_c)$ is the normalized frequency of co-occurrence of t_k and t_c. $\gamma(t_k, t_c)$ is the inverse segment frequency of t_c which co-occurs with t_k. $M(d_A)$ is the length of the segment d_A counted in word. They are defined as follows:

$$\alpha(d_A, t_k, p, t_c) = \frac{d(d_A, t_k, p) - dist(d_A, t_k, p, t_c)}{d(d_A, t_k, p)} \quad (3)$$

$$\beta(t_k, t_c) = \frac{rtf(t_k, t_c)}{atf(t_k)}, \quad (4)$$

$$\gamma(t_k, t_c) = log(\frac{N}{df(t_c)}). \quad (5)$$

Here, the function $dist(d_A, t_k, p, t_c)$ is the distance between p-th t_k and t_c counted in word. $d(d_A, t_k, p)$ is the threshold of distance of two words in co-occurrence. Since, in our system, we only focus on co-occurrences within a sentence, $d(d_A, t_k, p)$ is the number of words in the sentence we focus on. $atf(t_k)$ is the total number of t_k's occurrences within the document which includes the segment d_A. $rtf(t_k, t_c)$ is the total number of co-occurrences of t_k and t_c. N is the total number of segments in the document, $df(t_c)$ is the number of segments in which t_c occurs.

Hereafter, we call the method *the term co-occurrence method*, or *C-method* for short.

3.2 Utilization of lexical chains

In general, a *lexical chain* denotes a series of relevant words, which are in some lexical cohesion. The lexical cohesion is usually recognized by using some linguistic resources like thesauri (Green, 1996). In related documents, however, it is expected that a term has only one meaning and a concept is represented by one term. Therefore, in this paper we adopt the simplest version of lexical chain, namely, the repetitions of a term illustrated in Figure 3. Since a lexical chain may be the repetition of a term which goes through more than one segments, it is expected that we can capture the *current global topic* by it.

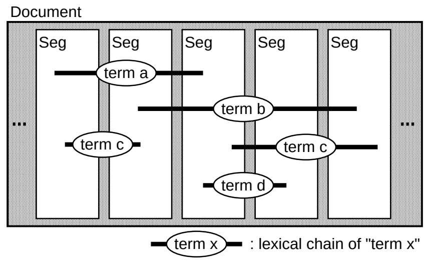

Figure 3: Lexical chains as repetitions of terms

By lexical chains, we would like to detect important terms over a series of segment. Therefore, a lexical chain of which length is extremely long or short is not a good clue for it. Accordingly, we introduce three thresholds, *the maximum gap length*(Th_{gap}), *the maximum length*(Th_{max}) and *the minimum length*(Th_{min}) to filter out such meaningless lexical chains as follows.

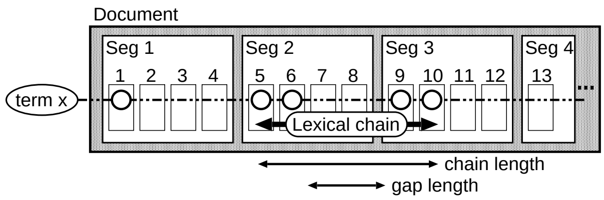

Figure 4: Detection of lexical chains

1. Decompose a segment into a series of small units. The unit in this paper is a sentence. For example, the segment 2 in Figure 4 consists of four units, namely 5, 6, 7 and 8.
2. For each term, do the following procedure.
   (a) Mark the all units in which the term appears.
   (b) If the number of non-marked units between two marked units is less than or equal to Th_{gap}, the marked units are parts of a lexical chain of the term.
   (c) Filter out the lexical chains of which length is less than Th_{min} or more than Th_{max}.

Lexical chains are took into account to calculate similarity among segments as adjustments of the tf values.

Note that compound words themselves scarcely make lexical chains (repetitions) because they usually have low frequencies. Especially, it is problematic in technical documents, because they have many compound words. Of course, each constituent word of compound word relatively makes a lexical chain easily. According to our preliminary experiment, on the contrary, in the condition that a compound word is treated as one term, the baseline $tf \cdot idf$ method is more effective than in the case that a compound word is decomposed into a series of constituent words. In order to take advantage of both treatments, we treat each compound word as follows:

1. A compound word is treated as one word in the $tf \cdot idf$ calculation.
2. In the case of detecting lexical chains, the compound word is decomposed into a series of constituent words.
3. All lexical chains of the constituent words are take into account in adjusting the tf value of the compound word.

That is, the tf value of the term t_k in the segment d_i is revised as the following formula:

$$tf'(d_i, t_k) = tf(d_i, t_k) \cdot (1 + \sum_{t_m \in T_m(t_k)} f_c(d_i, t_m)) \qquad (6)$$

where $T_m(t_k)$ is the set of constituent words of the term t_k, and $f_c(d_i, t_m)$ is the predicate function which returns 1 if the term t_m makes a lexical chain at segment d_i, otherwise returns 0.

Hereafter, we call the method *the lexical chain method*, or *L-method* for short.

4 Combination of two methods

Since the two method in the last section are based on the different relations of terms, the combination is expected to contribute to the improvement of IR effectiveness.

In this section, we propose the two ways to combine them.

4.1 Macro-combination method

The first method is called *the macro-combination method* or *MaC-method* for short. It combines two scores of similarity, $sim_C(d_A, d_B)$ and $sim_L(d_A, d_B)$, which are calculated independently by the C and L-method respectively. That is, the similarity $sim_{MaC}(d_A, d_B)$ of two segment d_A and d_b is defined by the following formula:

$$sim_{MaC}(d_A, d_B) = sim_C(d_A, d_B) + sim_L(d_A, d_B). \qquad (7)$$

4.2 Micro-combination method

The second method is called *the micro-combination method* or *MiC-method* for short. It combines two methods at the adjustments of the tf values. That is, the tf value of term t_k in segment d_i is defined by the following formula.

$$tf'(d_i, t_k) = tf(d_i, t_k) + r_{coc}(d_i, t_k) + r_{lex}(d_i, t_k) \qquad (8)$$

where $r_{coc}(d_i, t_k)$ and $r_{lex}(d_i, t_k)$ are the adjustment terms of the tf in (1) and (6) respectively.

5 Experimental Results

5.1 Evaluation of our methods

In this section , we will evaluate our methods, or, the C-method, the L-method, the MaC-method and the MiC-method in terms of effectiveness in segment retrieval. In IR systems, the effectiveness is usually measured with the recall-precision curve. In the retrieval of segments relevant to a segment, the number of relevant segments varies considerably segment by segment[1]. Therefore, our evaluation is based on not the similarity ranking of segments relevant to a certain segment, but on the similarity ranking of all segment pairs. If the relevant segment pairs move to the higher ranks by introducing a certain method, we may consider the (average) performance of the new method to be more effective than the old one.

In order to validate the advantage of our method precisely, we use not only the average precision in the recall-precision curve but also the Wilcoxon matched-pairs signed-ranks test(MPSR test), which is a kind of non-parametric statistical test (Hull, 1993).

5.2 Sample Documents

As for the baseline method described in Section 2, we have already examined the performance with large documents(Mori et al., 1998). Since the documents are too large to check all truly relevant pairs of segments, we examined a subset of segment pairs as an approximation.

[1] As for our example documents, it varies from zero to six.

In this paper, however, we use the documents of medium size instead of the large documents, because we can manually check the relevance of all pairs of segments and it is expected that we can examine the details of our methods. As such medium size documents, we select three technical documents, more precisely, three instruction manuals of VCRs. Although such combination of documents is slightly different from the real situation we aimed at, it should be enough to check the performance of systems which do not deeply analyze the documents but treat the documents as a series of terms.

In order to investigate dependence of performance on the degree of overlaps of vocabulary, we use two documents, M_1 and M_2, which are relatively resemble each other in terms of content, and one document, M_3, which is much more different from M_1 and M_2. The relevancy among segments are carefully judged by two graduate school students.

5.3 Results

Table 1 shows the comparison of B, C, L, MaC, and MiC-method, where the numerical values represent the average values of precision, and the relations in parentheses are the result of MPSR test. As an example, the recall-precision curves for the combination $M_2 \Leftrightarrow M_3$ are shown in Figure 5, 6 and 7.

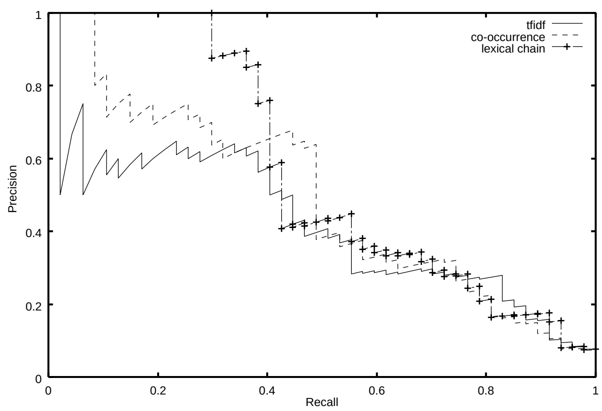

Figure 5: Recall-precision curve (B,C and L-method) of $M_2 \Leftrightarrow M_3$

The results are summarized as follows:

- All of our methods outperform the baseline $tf \cdot idf$ method (B-method).
- In the case that the documents are close to each other (e.g. the combination $M_1 \Leftrightarrow M_2$), even the baseline method, it achieves high performance.
- The MaC-method is not so effective. It is outperformed by the L-method.
- The most effective methods are the MiC-method and the L-method. Although the average performance of the MiC-method is almost same as L-method, the recall-precision curves show that the MiC-method outperforms the L-method in the low-recall area.

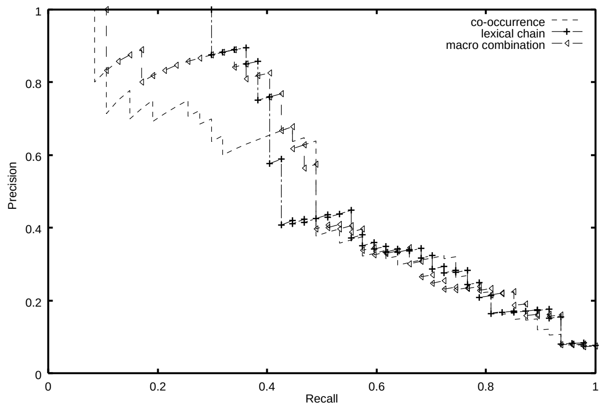

Figure 6: Recall-precision curve (C,L and MaC) of $M_2 \Leftrightarrow M_3$

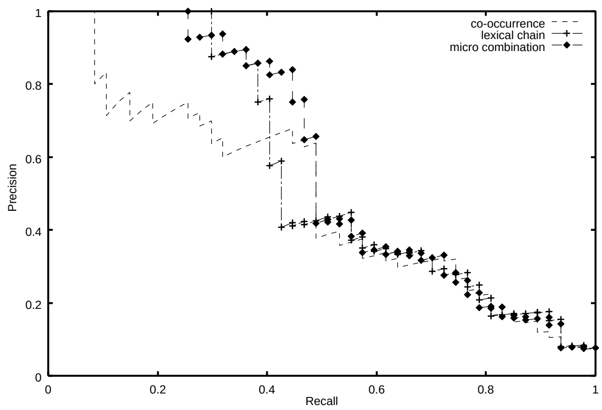

Figure 7: Recall-precision curve (C,L and MiC) of $M_2 \Leftrightarrow M_3$

Those results show that the MiC-method successfully combines the two elementary methods complementarily and achieves the good effectiveness. On the other hand, in the MaC-method the combined result is the average of two methods and could be worse than the L-method because the L-method outperforms the

Table 1: Comparison of our methods

Method	B	C	L	MaC	MiC
$M_1 \Leftrightarrow M_2$	0.680	0.671 (C > B)	0.709 (L > B)	0.698 (MaC < L)	0.707 (MiC $\simeq$ L)
$M_1 \Leftrightarrow M_3$	0.497	0.535 (C > B)	0.599 (L > B)	0.585 (MaC < L)	0.598 (MiC $\simeq$ L)
$M_2 \Leftrightarrow M_3$	0.407	0.478 (C > B)	0.550 (L > B)	0.533 (MaC < L)	0.575 (MiC $\simeq$ L)

C-method at almost all points in the recall-precision curve.

6 Related Works

From the viewpoint of fact that the retrieved objects are not documents themselves but the parts of documents, our system shares fundamentals with *passage retrieval*. A passage is a small part of document and corresponds to a segment in our system. Passage retrieval is firstly proposed by Salton et al.(Salton et al., 1993). Mochizuki et al.(Mochizuki et al., 1998) utilize the lexical chains in passage retrieval in order to determine the effective passages dynamically according to queries. Although they examine several types of lexical chains, they do not use the co-occurrence information to calculate the similarity, which is combined with the lexical chains in our method successfully.

Green(Green, 1996) proposes the method to generate hyper-links among newspaper articles by detecting lexical chains. He introduces the WordNet to detect semantic lexical chains. His method calculates the similarity of articles only with the information of lexical chains. He reports that the method does not outperform the $tf \cdot idf$ method. On the other hand, our method outperforms the $tf \cdot idf$ method by blending the lexical chains and term co-occurrence information with the basic $tf \cdot idf$ method.

7 Conclusion

In this paper, we proposed several similarity calculation methods which are intended to use in reading aid of related documents. We introduced two types of term relations, namely the term co-occurrence and the lexical chains, into the basic system which adopts the $tf \cdot idf$ method and the vector-space model. Our experimental result shows that the micro combination method we proposed here has good performance.

However, the combination of documents used in our experiment is slightly different from the real situation. We have to prepare other types of documents and perform further experiments in our future works.

References

Stephen J. Green. 1996. Using lexical chains to build hypertext links in newspaper articles. In *AAAI 96 Workshop on Internet-based Information Systems.*

David Hull. 1993. Using statistical testing in the evaluation of retrieval. In *Proceedings of SIGIR '93: 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338.

Yuji Matsumoto, Osamu Imaichi, Tatsuo Yamashita, Akira Kitauchi, and Tomoaki Imamura, 1996. *Japanese Morphological Analysis System ChaSen Manual (version 1.0b4).* Nara Institute of Science and Technology, November. (in Japanese).

Hajime Mochizuki, Makoto Iwayama, and Manabu Okumura. 1998. Passage-level document retrieval using lexical chains. IPSJ SIG-NL Notes NL-127-6, IPS Japan, September. (in Japanese).

Tatsunori Mori, Hiroshi Nakagawa, Nobuyuki Omori, and Jun Okamura. 1998. Hypertext authoring for linking relevant segments of related instruction manuals. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL 98)*, pages 929–933, August.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523.

Gerard Salton, A. Wong, and C.S. Yang. 1975. A vector space model for automatic indexing. *Communicatons of the ACM*, 18:613–620.

Gerard Salton, J. Allan, and Christopher Buckley. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of SIGIR '93: 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58.

Toru Takaki and Tsuyoshi Kitani. 1996. Relevance ranking of documents using query word co-occurrences. IPSJ SIG-FI Notes 96-FI-41-8, IPS Japan, April. (in Japanese).