Hypertext Authoring for Linking Relevant Segments of Related Instruction Manuals

Hiroshi Nakagawa and **Tatsunori Mori** and **Nobuyuki Omori** and **Jun Okamura**
Department of Computer and Electronic Engineering, Yokohama National University
Tokiwadai 79-5, Hodogaya, Yokohama, 240-8501, JAPAN
E-mail: nakagawa@naklab.dnj.ynu.ac.jp,{mori,ohmori,jun}@forest.dnj.ynu.ac.jp

Abstract

Recently manuals of industrial products become large and often consist of separated volumes. In reading such individual but related manuals, we must consider the relation among segments, which contain explanations of sequences of operation. In this paper, we propose methods for linking relevant segments in hypertext authoring of a set of related manuals. Our method is based on the similarity calculation between two segments. Our experimental results show that the proposed method improves both recall and precision comparing with the conventional $tf \cdot idf$ based method.

1 Introduction

In reading traditional paper based manuals, we should use their indices and table of contents in order to know where the contents we want to know are written. In fact, it is not an easy task especially for novices. Recent years, electronic manuals in a form of hypertext like *Help* of *Microsoft Windows* became widely used. Unfortunately it is very expensive to make a hypertext manual by hand especially in case of a large volume of manual which consists of several separated volumes. In a case of such a large manual, the same topic appears at several places in different volumes. One of them is an introductory explanation for a novice. Another is a precise explanation for an advanced user. It is very useful to jump from one of them to another of them directly by just clicking a button of mouse in reading a manual text on a browser like *NetScape*. This type of access is realized by linking them in hypertext format by hypertext authoring.

Automatic hypertext authoring has been focused on in these years, and much work has been done. For instance, Basili et al. (1994) use document structures and semantic information by means of natural language processing technique to set hyperlinks on plain texts.

The essential point in the research of automatic hypertext authoring is the way to find semantically relevant parts where each part is characterized by a number of key words. Actually it is very similar with information retrieval, IR henceforth, especially with the so called passage retrieval (Salton et al., 1993). J.Green (1996) does hypertext authoring of newspaper articles by word's lexical chains which are calculated using WordNet. Kurohashi et al. (1992) made a hypertext dictionary of the field of information science. They use linguistic patterns that are used for definition of terminology as well as thesaurus based on words' similarity. Furner-Hines and Willett (1994) experimentally evaluate and compare the performance of several human hyper linkers. In general, however, we have not yet paid enough attention to a full-automatic hyper linker system, that is what we pursue in this paper.

The new ideas in our system are the following points:

1. Our target is a multi-volume manual that describes the same hardware or software but is different in their granularity of descriptions from volume to volume.
2. In our system, hyper links are set not between an anchor word and a certain part of text but between two segments, where a segment is a smallest formal unit in document, like a subsubsection of LaTeX if no smaller units like subsubsubsection are used.
3. We find pairs of relevant segments over two volumes, for instance, between an introductory manual for novices and a reference manual for advanced level users about the same software or hardware.
4. We use not only $tf \cdot idf$ based vector space model but also words' co-occurrence information to measure the similarity between segments.

2 Similarity Calculation

We need to calculate a semantic similarity between two segments in order to decide whether two of them are linked, automatically. The most well known method to calculate similarity in IR is a vector space model based on $tf \cdot idf$ value. As for idf, namely inverse document frequency, we adopt a segment instead of document in the definition of *idf*. The definition of *idf* in our system is the following.

$$idf(t) = \log \frac{\sharp \text{ of segments in the manual}}{\sharp \text{ of segments in which } t \text{ occurs}} + 1$$

Then a segment is described as a vector in a vector space. Each dimension of the vector space consists of each term used in the manual. A vector's value of each dimension corresponding to the term t is its $tf \cdot idf$ value. The similarity of two segments is a *cosine* of two vectors corresponding to these two segments respectively. Actually the *cosine* measure similarity based on $tf \cdot idf$ is a baseline in evaluation of similarity measures we propose in the rest of this section.

As the first expansion of definition of $tf \cdot idf$, we use case information of each noun. In Japanese, case information is easily identified by the case particle like *ga*(nominal marker), *o*(accusative marker), *ni*(dative marker) etc. which are attached just after a noun. As the second expansion, we use not only nouns (+ case information) but also verbs because verbs give important information about an action a user does in operating a system. As the third expansion, we use co-occurrence information of nouns and verbs in a sentence because combination of nouns and a verb gives us an outline of what the sentence describes. The problem at this moment is the way to reflect co-occurrence information in $tf \cdot idf$ based vector space model. We investigate two methods for this, namely,

1. Dimension expansion of vector space, and
2. Modification of tf value within a segment.

In the following, we describe the detail of these two methods.

2.1 Dimension Expansion

This method is adding extra-dimensions into the vector space in order to express co-occurrence information. It is described more precisely as the following procedure.

1. Extracting a case information (*case particle* in Japanese) from each noun phrase. Extracting a verb from a clause.
2. Suppose be there n noun phrases with a case particle in a clause. Enumerating every combination of 1 to n noun phrases with case particle. Then we have $\sum_{k=1}^{n} {}_n\mathrm{C}_k$ combinations.
3. Calculating $tf \cdot idf$ for every combination with the corresponding verb. And using them as new extra dimensions of the original vector space.

For example, suppose a sentence "An end user learns the programming language." Then in addition to dimensions corresponding to every noun phrase like "end user", we introduce the new dimensions corresponding to co-occurrence information such as:

- (VERB, learn) (NOMNINAL end user) (ACCUSATIVE programming language)
- (VERB, learn) (NOMNINAL end user)
- (VERB, learn) (ACCUSATIVE programming language)

We calculate $tf \cdot idf$ of each of these combinations that is a value of vector corresponding to each of these combinations. The similarity calculation based on *cosine* measure is done on this expanded vector space.

2.2 Modification of tf value

Another method we propose for reflecting co-occurrence information to similarity is modification of tf value within a segment. (Takaki and Kitani, 1996) reports that co-occurrence of word pairs contributes to the IR performance for Japanese news paper articles.

In our method, we modify tf of pairs of co-occurred words that occur in both of two segments, say d_A and d_B, in the following way. Suppose that a term t_k, namely noun or verb, occurs f times in the segment d_A. Then the modified $tf'(d_A, t_k)$ is defined as the following formula.

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

where cw and cw' are scores of importance for co-occurrence of words, t_k and t_c. Intuitively, cw and cw' are counter parts of $tf \cdot idf$ for co-occurrence of words and co-occurrence of (noun case-information), respectively. cw is defined by the following formula.

$$\begin{aligned} & cw(d_A, t_k, p, t_c) \\ & = \frac{\alpha(d_A, t_k, p, t_c) \times \beta(t_k, t_c) \times \gamma(t_k, t_c) \times C}{M(d_A)} \end{aligned}$$

where $\alpha(d_A, t_k, p, t_c)$ is a function expressing how near $t_k$and t_c occur, p denotes that pth t_k's occurrence in the segment d_A, and $\beta(t_k, t_c)$ is a normalized frequency of co-occurrence of t_k and t_c. Each of them is defined as follows.

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

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

where the function $dist(d_A, t_k, p, t_c)$ is a distance between pth t_k within d_A and t_c counted by word. $d(d_A, t_k, p)$ shows the threshold of distance within which two words are regarded as a co-occurrence. Since, in our system, we only focus on co-occurrences within a sentence, $\alpha(d_A, t_k, p, t_c)$ is calculated for pairs of word occurrences within a sentence. As a result, $d(d_A, t_k, p)$ is a number of words in a sentence we focus on. $atf(t_k)$ is a total number of t_k's occurrences within the manual we deal with. $rtf(t_k, t_c)$ is a total number of co-occurrences of t_k and t_c within a sentence. $\gamma(t_k, t_c)$ is an inverse document frequency (in this case "inverse segment frequency") of t_c which co-occurs with t_k, and defined as follows.

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

where N is a number of segments in a manual, and $df(t_c)$ is a number segments in which t_c occurs with t_k.

$M(d_A)$ is a length of segment d_A counted in morphological unit, and used to normalize cw. C is a weight parameter for cw. Actually we adopt the value of C which optimizes 11point precision as described later.

The other modification factor cw' is defined in almost the same way as cw is. The difference between cw and cw' is the following. cw is calculated for each noun. On the other hand, cw' is calculated for each combination of noun and its case information. Therefore, cw' is calculated for each (noun, case) like (user, NOMINAL). In other words, in calculation of cw', only when (noun-1, case-1) and (noun-2, case-2), like (user NOMINAL) and (program ACCUSATIVE), occur within the same sentence, they are regarded as a co-occurrence.

Now we have defined cw and cw'. Then back to the formula which defines tf'. In the definition of tf', $T_c(t_k, d_A, d_B)$ is a set of word which occur in both of d_A and d_B. Therefore cws and cw's are summed up for all occurrences of t_k in d_A. Namely we add up all cws and cw's whose t_c is included in $T_c(t_k, d_A, d_B)$ to calculate tf'.

3 Implementation and Experimental Results

Our system has the following inputs and outputs.

Input is an electronic manual text which can be written in plain text,LaTeXor HTML)

Output is a hypertext in HTML format.

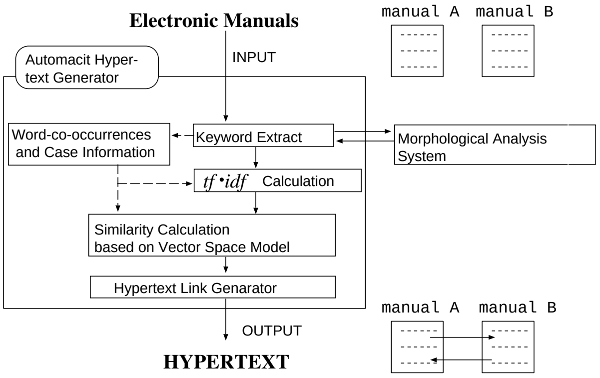


Figure 1: Overview of our hypertext generator

We need a browser like *NetScape* that can display a text written in HTML. Our system consists of four sub-systems shown in Figure 1.

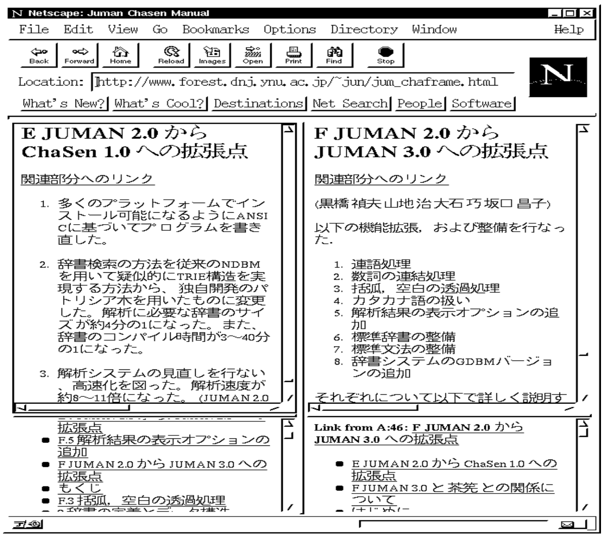
Keyword Extraction Sub-System In this sub-system, a morphological analyzer segments out the input text, and extract all nouns and verbs that are to be keywords. We use Chasen 1.04b (Matsumoto et al., 1996) as a morphological analyzer for Japanese texts. Noun and Case-information pairs are also made in this sub-system. If you use the dimension expansion described in 2.1, you introduce new dimensions here.

tf · idf Calculation Sub-System This sub-system calculates $tf \cdot idf$ of extracted keywords by Keyword Extraction Sub-System.

Similarity Calculation Sub-System This sub-system calculates the similarity that is represented by *cosine* of every pair of segments based on $tf \cdot idf$ values calculated above. If you use modifications of tf values described in 2.2, you calculated modified tf, namely tf' in this sub-system.

Hypertext Generator This sub-system translates the given input text into a hypertext in which pairs of segments having high similarity, say high *cosine* value, are linked. The similarity of those pairs are associated with their links for user friendly display described in the following .

We show an example of display on a browser in Figure 2. The display screen is divided into four parts. The upper left and upper right parts show a distinct part of manual text respectively. In the lower left (right) part, the title of segments that are relevant to the segment displayed on the upper left (right) part are displayed in descending order of



http://www.forest.dnj.ynu.ac.jp/~jun/jum_chaframe.html

E JUMAN 2.0 から ChaSen 1.0 への拡張点

関連部分へのリンク

1. 多くのプラットフォームでインストール可能になるようにANSI Cに基づいてプログラムを書き直した。
2. 辞書検索の方法を従来のNDBMを用いて疑似的にTRIE構造を実現する方法から、独自開発のパトリシア木を用いたものに変更した。解析に必要な辞書のサイズが約4分の1になった。また、辞書のコンパイル時間が3〜40分の1になった。
3. 解析システムの見直しを行ない、高速化を図った。解析速度が約8〜11倍になった。(JUMAN 2.0

- 拡張点
- F.5 解析結果の表示オプションの追加
- F JUMAN 2.0 から JUMAN 3.0 への拡張点
- もくじ
- F.3 括弧，空白の透過処理

F JUMAN 2.0 から JUMAN 3.0 への拡張点

関連部分へのリンク

(黒橋 禎夫 山地 治 大石 巧 坂口 昌子)

以下の機能拡張，および整備を行なった.

1. 連語処理
2. 数詞の連結処理
3. 括弧，空白の透過処理
4. カタカナ語の扱い
5. 解析結果の表示オプションの追加
6. 標準辞書の整備
7. 標準文法の整備
8. 辞書システムのGDBMバージョンの追加

それぞれについて以下で詳しく説明す

Link from A:46: F JUMAN 2.0 から JUMAN 3.0 への拡張点

- E JUMAN 2.0 から ChaSen 1.0 への拡張点
- F JUMAN 3.0 と 茶筅 との関係について

Figure 2: The use of this system

similarity. Since these titles are linked to the corresponding segment text, if we click one of them in the lower left (right) part, the hyperlinked segment's text is instantly displayed on the upper right (left) part, and its relevant segments' title are displayed on the lower right (left) part. By this type of browsing along with links displayed on the lower parts, if a user wants to know relevant information about what she/he is reading on the text displayed on the upper part, a user can easily access the segments in which what she/he wants to know might be written in high probability.

Now we describe the evaluation of our proposed methods with recall and precision defined as follows.

$$recall = \frac{\sharp \text{ of retrieved pairs of relevant segments}}{\sharp \text{ of pairs of relevant segments}}$$

$$precision = \frac{\sharp \text{ of retrieved pairs of relevant segments}}{\sharp \text{ of retrieved pairs of segments}}$$

The first experiment is done for a large manual of APPGALLARY(Hitachi, 1995) which is 2.5MB large. This manual is divided into two volumes. One is a tutorial manual for novices that contains 65 segments. The other is a help manual for advanced users that contains 2479 segments. If we try to find the relevant segments between ones in the tutorial manual and ones in the help manual, the number of possible pairs of segments is 161135. This number is too big for human to extract all relevant segment manually. Then we investigate highest 200 pairs of segments by hand, actually by two students in the engineering department of our university to extract pairs of relevant segments. The guideline of selection of pairs of relevant segments is:

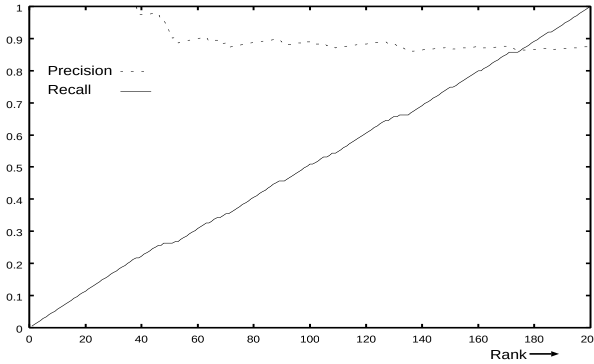


Figure 3: Recall and precision of generated hyperlinks on large-scale manuals

Table 1: Manual combinations and number of right correspondences of segments

pair of manuals	A⇔ B	A ⇔ C	B ⇔ C
♯ of all pairs	1056	896	924
♯ of relevant pairs	65	60	47

1. Two segments explain the same operation or the same terminology.
2. One segment explains an abstract concept and the other explains that concept in concrete operation.

Figure 3 shows the recall and precision for numbers of selected pairs of segments where those pairs are sorted in descending order of *cosine* similarity value using normal $tf \cdot idf$ of all nouns. This result indicates that pairs of relevant segments are concentrated in high similarity area. In fact, the pairs of segments within top 200 pairs are almost all relevant ones.

The second experiment is done for three small manuals of three models of video cassette recorder(MITSUBISHI, 1995c; MITSUBISHI, 1995a; MITSUBISHI, 1995b) produced by the same company. We investigate all pairs of segments that appear in the distinct manuals respectively, and extract relevant pairs of segment according to the same guideline we did in the first experiment by two students of the engineering department of our university. The numbers of segments are 32 for manual A(MITSUBISHI, 1995c), 33 for manual B(MITSUBISHI, 1995a) and 28 for manual C(MITSUBISHI, 1995b), respectively. The number of relevant pairs of segments are shown in Table 1.

We show the 11 points precision averages for these methods in Table 2. Each recall-precision curve, say Keyword, dimension N, cw+cw' tf, and Normal Query, corresponds to the methods described in the previous section. We describe the more precise definition of each in the following.

Table 2: 11 point average of precision for each method and combination

Method	A ⇔ B	A ⇔ C	B ⇔ C
Keyword	0.678	0.589	0.549
cw+cw' tf	0.683	0.625	0.582
C	0.1	0.6	1.3
dimension N	0.684	0.597	0.556
Normal Query	0.692	0.532	0.395

Keyword: Using $tf \cdot idf$ for all nouns and verbs occuring in a pair of manuals. This is the baseline data.
dimension N: Dimension Expansion method described in section 2.1. In this experiment, we use only noun-noun co-occurrences.
cw+cw' tf: Modification of tf value method described in section2.2. In this experiment, we use only noun-verb co-occurrences.
Normal Query: This is the same as **Keyword** except that vector values in one manual are all set to 0 or 1, and vector values of the other manual are $tf \cdot idf$.

In the rest of this section, we consider the results shown above point by point.

The effect of using $tf \cdot idf$ information of both segments

We consider the effect of using $tf \cdot idf$ of two segments that we calculate similarity. For comparison, we did the experiment **Normal Query** where $tf \cdot idf$ is used as vector value for one segment and 1 or 0 is used as vector value for the other segment. This is a typical situation in IR. In our system, we calculate similarity of two segments already given. That makes us possible using $tf \cdot idf$ for both segments. As shown in Table 2, **Keyword** outperforms **Normal Query**.

The effect of using co-occurrence information

The same types of operation are generally described in relevant segments. The same type of operation consists of the same action and equipment in high probability. This is why using co-occurrence information in similarity calculation magnifies similarities between relevant segments. Comparing dimension expansion and modification of tf, the latter outperforms the former in precision for almost all recall rates. Modification of tf value method also shows better results than dimension expansion in 11 point precision average shown in Table 2 for A-C and B-C manual pairs. As for normalization factor C of modification of tf value method, the smaller C becomes, the less tf value changes and the more similar the result becomes with the baseline case in which only tf is used. On the contrary, the bigger C becomes, the more incorrect pairs get high similarity and the precision deteriorates in low recall area. As a result, there is an optimum C value, which we selected experimentally for each pair of manuals and is shown in Table 2 respectively.

4 Conclusions

We proposed two methods for calculating similarity of a pair of segments appearing in distinct manuals. One is Dimension Expansion method, and the other is Modification of tf value method. Both of them improve the recall and precision in searching pairs of relevant segment.This type of calculation of similarity between two segments is useful in implementing a user friendly manual browsing system that is also proposed and implemented in this research.

References

Roberto Basili, Fabrizio Grisoli, and Maria Teresa Pazienza. 1994. Might a semantic lexicon support hypertextual authoring? In *4th ANLP*, pages 174–179.

David Elhs. Jonathan Furner-Hines and Peter Willett. 1994. On the measurement of inter-linker consistency and retrieval effectiveness in hypertext databases. In *SIGIR '94*, pages 51–60.

Hitachi, 1995. *How to use the APPGALLERY, APPGALLERY On-Line Help*. Hitachi Limited.

Stephen J.Green. 1996. Using lexcal chains to build hypertext links in newspaper articles. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases, Portland, Oregon.*

S. Kurohashi, M. Nagao, S. Sato, and M. Murakami. 1992. A method of automatic hypertext construction from an encyclopedic dictionary of a specific field. In *3rd ANLP*, pages 239–240.

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, Nov.

MITSUBISHI, 1995a. *MITSUBISHI Video Tape Recorder HV-BZ66 Instruction Manual.*

MITSUBISHI, 1995b. *MITSUBISHI Video Tape Recorder HV-F93 Instruction Manual.*

MITSUBISHI, 1995c. *MITSUBISHI Video Tape Recorder HV-FZ62 Instruction Manual.*

Gerard Salton, J. Allan, and Chris Buckley. 1993. Approaches to passage retrieval in full text information systems. In *SIGIR '93*, pages 49–58.

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