Cross-Lingual Information Retrieval based on LSI with Multiple Word Spaces

Tatsunori Mori Tomoharu Kokubu Takashi Tanaka
Div. of Electrical and Computer Eng., Yokohama National University
79-5 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan
{mori,kokubu,tanaka}@forest.dnj.ynu.ac.jp

Abstract

In this paper, we report the utilization of a large-scaled bilingual corpus in Cross-Language Latent Semantic Indexing(CL-LSI). When we construct one monolithic word space with a large-scaled corpus, we encounter problems such as the increase in ambiguity of word translation, the difficulty in singular value decomposition, which is the important process in LSI. In order to cope with the problems, we introduce the method in which the large bilingual corpus is divided into smaller sub-corpora according to the similarity among documents in it, and from each of them one word sub-space is created. By placing each document in the word sub-space, which is made from the sub-corpus most similar to the document, ambiguity of translation is expected to decrease. In the searching process, the query is placed into every word sub-spaces, and similarity between the query and the documents are calculated.

1 Introduction

As described in literatures[11, 7, 8], many kinds of methods of cross-lingual information retrieval make use of trans-lingual dictionaries. Those dictionaries are usually compiled and examined by hand. The accuracy of translation of those systems should be high, although the accuracy depends on not only the scale and quality of the dictionaries but also the way to use them.

As antithesis of this kind of methods, in this paper we examine a corpus based method in terms of the following points:

- How precisely (or inaccurately) systems can perform cross-language information retrieval (CLIR) for a large scaled document database without dictionaries?
- Can we full-automatically construct translation information from a large scaled existing corpora?

Of course, the corpus-based systems, which does not use dictionaries, is supposed to be less effective than systems with dictionaries compiled by hand. On the contrary, Carbonell et al.[2] shows by middle-scaled experiments with 1134 dual-language documents that an example-based Machine Translation establishing corpus-based term equivalences outperforms Machine-Readable-Dictionary-based query translation. However, it still is not clear how precisely (or inaccurately) the systems can do CLIR for a large scale document database without dictionaries.

Accordingly, in the NTCIR2 evaluation, we make experiments of CLIR only with the bilingual corpus of NTCIR1. Although we should also take account of the cost to compile and maintain the bilingual corpus, some kinds of multi-lingual corpora are growing larger year by year without considerable labor, like summaries of technical papers written in more than one language.

Among variations of corpus-based methods, we pay attention to a method called 'Cross Language Latent Semantic Indexing' (CL-LSI), which uses a set of dual-language documents as the resource for language translation. In this paper, we propose a method to apply CL-LSI to large scale multi-lingual corpora by introducing plural word spaces(LSI spaces). We also evaluate the method in the experiments of the NTCIR2 Japanese-English CLIR evaluation. In the experiment, we use a set of dual-language (Japanese-English) summaries of technical papers, which is distributed in the NTCIR1 evaluation.

2 Cross Language Latent Semantic Indexing

Cross-language LSI (CL-LSI) is a fully automatic method for cross-language document retrieval in which no query translation is required [4, 5]. Queries in one language can retrieve documents in other languages as well as the original language. This is accomplished by a method that automatically constructs a multi-lingual semantic space using Latent Semantic Indexing (LSI). For the CL-LSI method to be used, an initial sample of documents is translated by humans or, perhaps, by machine. From these translations, we produce a set of dual-language documents (i.e., documents consisting of parallel text from both languages) that are used to "train" the system. An LSI analysis of these training documents results in a dual-language semantic space in which terms from both languages are represented. Standard mono-lingual documents

are then "folded in" to this space on the basis of their constituent terms. Queries in either language can retrieve documents in either language without the need to translate the query because all documents are represented as language-independent numerical vectors in the same LSI space.

2.1 Latent Semantic Indexing

Most information retrieval methods depend on exact matches between words in users' queries and words in documents. Such methods will, however, fail to retrieve relevant materials that do not share words with users' queries. One reason for this is that the standard retrieval models (e.g.,Boolean, standard vector, probabilistic) treat words as if they are independent, although it is quite obvious that they are not. A central theme of LSI is that term-term inter-relationships can be automatically modeled and used to improve retrieval[3]. This is critical in cross-language retrieval since direct term matching is of little use. LSI examines the similarity of the "contexts" in which words appear, and creates a reduced-dimension feature space in which words that occur in similar contexts are near each other. LSI uses a method from linear algebra, singular value decomposition (SVD)[10], to discover the important associative relationships. It is not necessary to use any external dictionaries, thesauri, or knowledge bases to determine these word associations because they are derived from a numerical analysis of existing texts. The learned associations are specific to the domain of interest, and are derived completely automatically. The singular-value decomposition (SVD) technique is closely related to eigenvector decomposition and factor analysis. For information retrieval and filtering applications we begin with a large term-document matrix, in much the same way as vector or Boolean methods do. The (i, j) element of the matrix is the frequency of the term i in the document j. This term document matrix is decomposed into a set of k, typically 200-300, orthogonal factors from which the original matrix can be approximated by linear combination. This analysis reveals the "latent" structure in the matrix that is obscured by variability in word usage.

Traditional vector methods represent documents as linear combinations of orthogonal terms. In contrast, LSI represents terms as continuous values on each of the orthogonal indexing dimensions. Terms are not independent. When two terms are used in similar contexts (documents), they will have similar vectors in the reduced-dimension LSI representation. LSI partially overcomes some of the deficiencies of assuming independence of words, and provides away of dealing with synonymy automatically without the need for a manually constructed thesaurus. The result of the SVD is a set of vectors representing the location of each term and document in the reduced k-dimension LSI representation. Retrieval proceeds by using the terms in a query to identify a point in the space. Technically, the query is located at the weighted vector sum of its constituent terms. Documents are then ranked by their similarity to the query, typically using a cosine measure of similarity.

New documents (or terms) can be added to the LSI representation using a procedure we call "folding in". This method assumes that the LSI space is a reasonable characterization of the important underlying dimensions of similarity, and that new items can be described in terms of the existing dimensions. A document is located at the weighted vector sum of its constituent terms.

2.2 Cross-Language Retrieval Using LSI

LSI could easily be adapted to cross-language retrieval as shown in Figure 1. An initial sample of documents is translated by human or, perhaps, by machine, to create a set of dual-language training documents.

A set of training documents is analyzed using LSI, and the result is a reduced dimension semantic space in which related terms are near each other. Because the training documents contain both terms of two languages, the LSI space will contain terms from both languages, and the training documents. This is what makes it possible for the CL-LSI method to avoid query or document translation. Words that are consistently paired will be given identical representations in the LSI space, whereas words that are frequently associated with one another will be given similar representations.

The next step in the CL-LSI method is to add (or "fold in") documents in just one language. This is done by locating a new document at the weighted vector sum of its constituent terms. The result of this process is that each document in the database, whether it is in one of two language, has a language-independent representation in terms of numerical vectors. Users can now pose queries in one of those languages and get back the most similar documents regardless of language.

3 Issues in making LSI spaces from a huge set of dual-language documents

The CL-LSI can be considered as the method which is effective mainly for document database in a certain specific domain. If the database includes documents from diverse domains, we have to collect a large number of dual-language documents in order to make a huge LSI space which has enough vocabulary for the document database. When we would like to obtain such an LSI space, we face the problem in the process of SVD. Since SVD is a kind of operation for matrices, the time and space complexity of computation will increase for larger data. Thus, if we use a huge set of dual-language documents, the process will break down because of the shortage of computer memory.

For example, we can find about 180 thousand dual-language summaries in the NTCIR1 corpus and 370 thousand words in it. The document-word matrix for the summaries will have 67G elements. It can not be stored in the memory of computers except for super computers, even if the matrix is rather sparse.

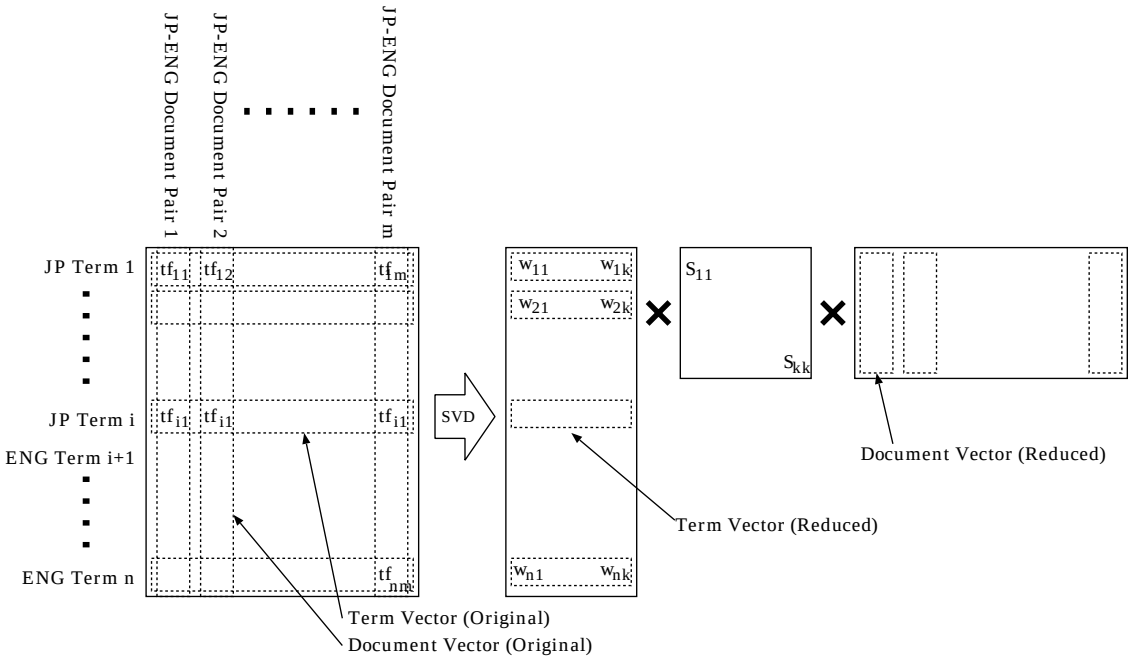

Figure 1. Cross Language Latent Semantic Indexing

Therefore, we introduce a method in which the large bilingual corpus is divided into smaller sub-corpora according to the similarity among documents in it, and from them separate LSI sub-spaces are created. Figure 2 shows the over view of our scheme. We expect the introduction of the multiple LSI sub-spaces to contribute toward the following objectives.

- SVD can be performed to make LSI spaces.
- The ambiguity in translation will be decreased if the area associated with each sub LSI space is appropriately restricted.

In order to adopt the method, we have to consider the following points:

- How can we divide a corpus into sub corpora?
- How can we place (new and mono-lingual) documents in the set of LSI spaces.
- How can we retrieve the documents in the set of LSI spaces.

We also have to study the following point:

- How can we extract linguistic expressions (e.g., words, phrases, etc) from documents to index the documents.

If a set of documents is limited to a certain area, the context in the documents will be restricted and the variations of translation of words would be also decreased. Therefore, it would be effective in CLIR to divide a set of dual-language documents into subsets according to the similarity among documents. Although clustering algorithms are usually used to do that, it costs a huge amount of computational resources to apply an ordinary clustering algorithm to a huge document set. We also have to adjust the size of subsets of documents manually according to computational resources, even if a certain clustering algorithm can be used.

On the other hand, in real situations, documents are accompanied with some useful information to guess the areas of documents. For example, each technical paper usually includes some information about 'are name' like the name of society. Therefore, in this paper we adopt the following approximated way of clustering in which the dual-language documents of the same area name treated as one document group and some of document groups are merged or divided according to the limit of size of document group.

1. Classify dual-language documents into area groups according to the area name of each document.
2. For each dual-language document, make a tfidf-based document vector $(tfidf_1, \ldots, tfidf_i, \ldots, tfidf_n)$, where the weight $tfidf_i$ is the TFIDF value of the term i in the document.
3. For each area group, calculate the 'area vector' by averaging all of the document vector in th group.
4. Select several large area groups manually. We call the groups 'major area groups'.
5. For each of other area groups, find the most similar major area group and marge it to the major

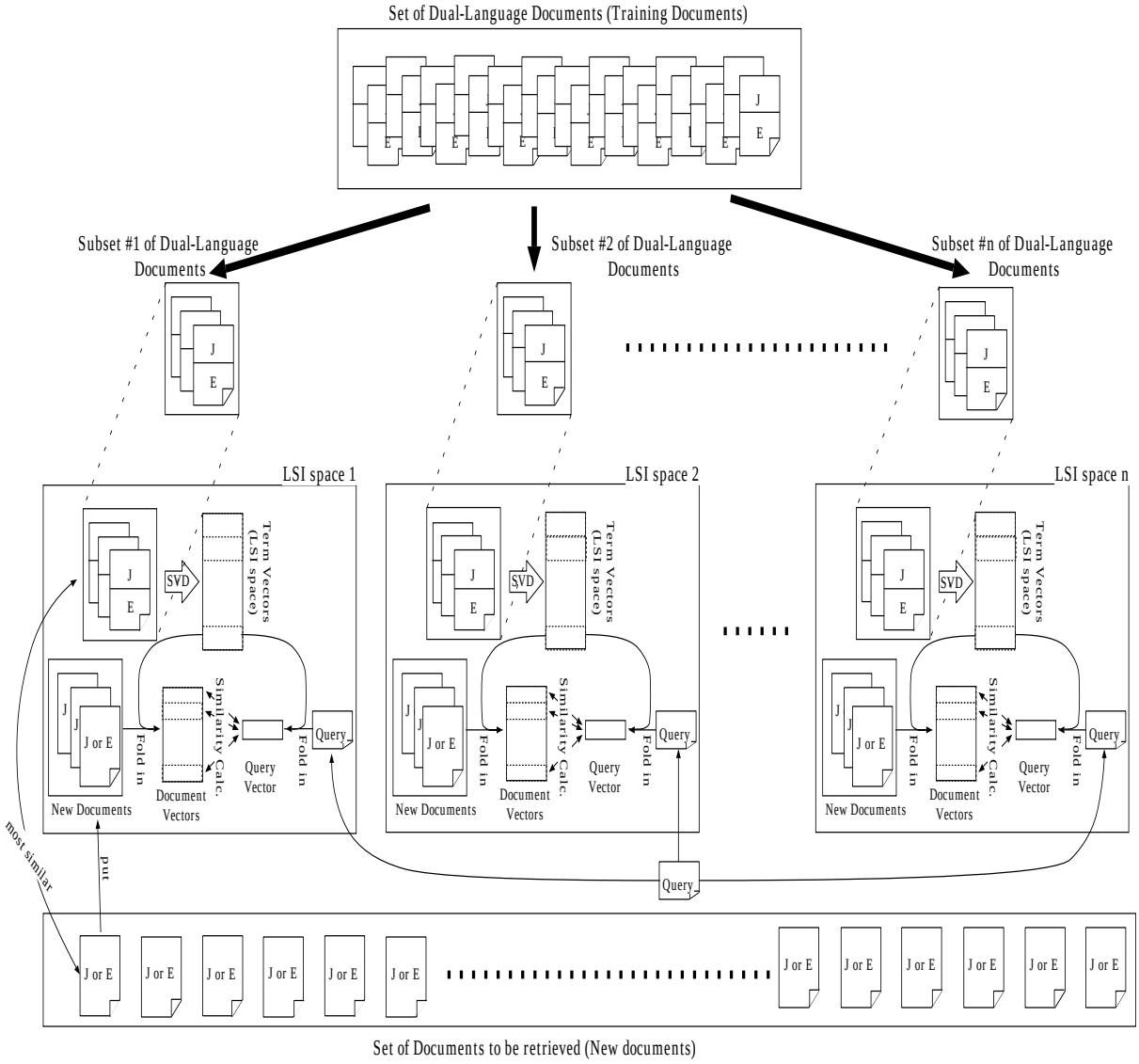

Figure 2. Overview of our Scheme

area group according to similarity among are vectors. The cosine value is adopted as the similarity of vectors.

6. For each area group the size of which exceeds a certain limit, divide it into sub groups of the required size. The limit is determined according to computational resources.

7. Update the area vectors of existing area groups.

In our experiment, as the set of dual-languaged documents, we extract about 180 thousand pairs of summaries written in Japanese and English from the NTCIR1 corpora. Those summaries come from technical papers of 57 scientific societies. In the step 4 of the above algorithm, we select six societies as major area groups. Then other area groups are merged to one of major area groups. According to our computational resources, each group of the four largest area groups is divided into two sub groups, and we finally obtained ten area groups. Each area group has from 14 to 26 thousand pairs of documents and from 78 to 115 thousand different terms. Total number of different terms in the corpus is 380 thousand.

For each area group, we obtain a set of term vectors (i.e. an LSI space) by the method described in section 2. The dimension of each LSI space is about 450 (from 430 to 463). We use the program `las2` in SVDPACKC[1] to perform SVD.

4 Storing Documents

In the scheme of CL-LSI, the (mono-lingual) documents to be retrieved are different from the dual-lingual documents which is used to make the LSI space. Thus, we have to "fold in" all of mono-lingual documents to the LSI space by using term vectors. Since we have plural LSI spaces corresponding to area groups, the structure of document vectors depends on what LSI space is selected.

When we have plural LSI spaces, several ways would be supposed to make document vectors as follows.

(a) Each document is placed in every LSI space as different vectors.

(b) Each document is placed in one selected LSI space.

Since all of translation information is took into account, the method (a) would be expected to be more effective than the method (b), However, the method (a) requires a huge storage system because each LSI space has the full set of document vectors. Accordingly, we adopt the realistic method (b).

In the 4, we have to consider the way to select one 'suitable' LSI space for each document. Because of accuracy of translation, it is desirable that a document is put into the LSI space which made from (training) documents in the same area as the target. Therefore, each document is placed to the LSI space the area vector of which is most similar to the tfidf-based vector of the document.

In order to "fold in" the mono-lingual documents to the LSI spaces, we use the following formula. It is different from the original method because of use of IDF values.

$$\mathbf{D} = \sum_{T_i \in D} tf(T_i, D) idf(T_i) \mathbf{T}_i, \qquad (1)$$

where

$\mathbf{D}$:	Document vector of document D in the LSI space.
$tf(T_i, D)$:	Frequency of term T_i in document D.
$idf(T_i)$:	IDF value of term T_i, $\log \frac{N}{df(T_i)} + 1$, where $df(T_i)$ is the document frequency of T_i in database.
$\mathbf{T}_i$:	Term vector of term T_i in LSI space.

5 Document Retrieval with Plural LSI Spaces

In the CL-LSI method, each query is also represented as a vector in the LSI space. According to the similarity between the query vector and each document vector, all documents are ranked. The retrieval of documents is performed based on the ranking information.

Since we have several LSI spaces in our scheme, we retrieve the documents by the following procedure:

1. Make one query vector for each LSI space by (1), in order to compare the query with all of documents.

2. In each LSI space, calculate the similarity between the query vector and each document vector.

3. Rank all documents in all LSI spaces according to their similarity value.

6 Problem of Unknown Words arising from Dividing Bilingual Corpus

In the CL-LSI method, the unknown words, which do not appear in the set of dual-language documents, are totally ignored because we cannot obtain the translation information of them. Thus, the accuracy of retrieval will be degraded when there are a number of words which appear not in dual-language documents but in documents to be retrieved. It is an inevitable problem. Unfortunately, we have another unknown-words problem in our scheme. It is caused by the division of corpus.

When we divide corpus, there may be words which do appear not in some sub-corpora but in the other sub-corpora. Since each LSI space has a different set of unknown words from others, in some cases we can not obtain desired results in the retrieval of documents.

For instance, let us consider the case where with three terms T_a, T_b and T_c in the query the system retrieves documents in the LSI spaces TS_1 and TS_2. We

suppose that the space TS_1 has T_a but does not have T_b and T_c, and the document D_1 with T_a is placed into the space. On the other hand, we also suppose that the space TS_2 has T_a, T_b and T_c, and the document D_2 with T_a and T_b is placed into the space.

In this situation, the document D_2 is more preferable to D_1 as a retrieved document, and we expect that the similarity between D_2 and the query is larger than the similarity between D_1 and the query. However, in reality the similarity about D_1 is larger than that about D_2. The reason is as follows. Since T_a is only considered in the process of the similarity calculation in TS_1, the query is substantially regarded as T_a and consequently D_1 is accidentally supposed to have all the terms in the query. Thus, D_1 has a high similarity value. On the other hand, the similarity calculation in TS_2 are based on all the the terms T_a, T_b and T_c. The similarity between D_2 and the query is lower even if D_2 has T_a and T_b, because the document does not have T_c, which is in the query.

In order to make the similarity calculation more preferable, we have to properly treat unknown words in the query as the factor of discounting similarity in every LSI space, instead of just discarding them. As one of ways to do that, we propose the introduction of one new dimension into each LSI space to represent unknown words. This method adjust the similarity between documents and the query in terms of unknown words by expanding each LSI space and treat unknown words as one vector which is orthogonal with all other term vectors as follows.

Suppose that an LSI space is an n-dimensional space where a term is represented as a vector $(w_1, ..., w_n)$. We introduce one new dimension into the space and obtain an (n+1)-dimensional space. In the new LSI space, each existing vector is represented as $(w_1, \ldots, w_n, 0)$, on the other hand, all unknown words in the query are represented as one vector $(0, \ldots, 0, 1)$.

The similarity in the new reconstructed space is discounted according to unknown words in the query. The adjusted similarity $sim(\mathbf{D}', \mathbf{Q}')$ between the document vector $\mathbf{D}'$ and the query vector $\mathbf{Q}'$ in the new LSI space is given by the following formula.

$$sim(\mathbf{D}', \mathbf{Q}') = \frac{\mathbf{D}' \cdot \mathbf{Q}'}{|\mathbf{D}'||\mathbf{Q}'|} \quad (2)$$

In (2), we have the following relations:

$$\begin{aligned} \mathbf{D}' \cdot \mathbf{Q}' &= \mathbf{D} \cdot \mathbf{Q} \\ |\mathbf{D}'| &= |\mathbf{D}| \\ |\mathbf{Q}'| &= |\mathbf{Q} + \mathbf{Q_u}| \\ &= \sqrt{|\mathbf{Q}|^2 + |\mathbf{Q_u}|^2} \\ &= \sqrt{|\mathbf{Q}|^2 + \sum_{T_i \in Q_u} (tf(T_i, Q_u) idf(T_i))^2}, \end{aligned}$$

where

- $\mathbf{D}$: Document vector before the adjustment
- $\mathbf{D}'$: Document vector after the adjustment
- $\mathbf{Q}$: Query vector before the adjustment
- $\mathbf{Q}'$: Query vector after the adjustment
- Q_u: List of unknown words in the query Q
- $\mathbf{Q_u}$: Vector of Q_u.

From those formulas, we obtain the similarity calculation as the following formula.

$$sim(\mathbf{D}', \mathbf{Q}') = \frac{\mathbf{D} \cdot \mathbf{Q}}{|\mathbf{D}| \sqrt{|\mathbf{Q}|^2 + \sum_{T_i \in Q_u} (tf(T_i, Q_u) idf(T_i))^2}} \quad (3)$$

7 Extraction of Index Words from Documents

In our experiment, we use simple words and compound words for indexing. We adopt the C-value[6] based method to extract compound words, although there are other ways to do that, like the method which use POS information and rules to make compound words. The method based on C-value has the good feature that it can be used uniformly for any languages.

7.1 Extraction of Simple Words

Japanese documents are tokenized by Japanese morphological analyzer JUMAN 3.61[9]. The analyzer also tags each word with POS information. We select nouns, adjectives, verbs, noun modifiers, adverbs, English words, katakana words as simple index words.

For English documents, each words are stemmed and useless words are eliminated with a stop word list.

8 Extraction of Compound Words

Our process of extracting compound nouns consists of the following two steps.

1. Make a suffix array of word sequences in dual-language documents in order to obtain the frequencies of word sequences. Discard word sequences the frequency of which is less than the threshold TH_f. The remaining word sequences are candidates of compound words.

2. Calculate the C-value for each candidate. Discard candidates the C-value of which is less than the threshold TH_c. The remaining candidate is regarded as selected compound words.

In our experiment, we divided the set of dual-language documents into eleven subsets because of limitation of our program. We process each subset by the above procedure with the condition, $TH_f = 5$ and $TH_c = 5$.

9 Experimental Results on Dividing LSI Space

First of all, we examine the effect of division of an LSI space by mate retrieval in the following conditions[1].

[1]The mate retrieval is the one of evaluation method for CLIR. The one language part of each dual-language document is submit-

We selected 6000 dual-language documents from the NTCIR1 corpus, and constructed two types of CL-LSI systems. First one has a monolithic LSI space, which is made from the whole of the document set. Second one has three LSI sub-spaces. As for the system, the document set was divided into three subsets according to area names and each LSI sub-space was made from each subset. In both of these systems, the dimension of each LSI space is about 150. We also selected the other 3000 dual-language documents for the evaluation by mate retrieval.

As the result in Table 1 shows, the system with plural LSI spaces has almost same or a little bit higher effectiveness than the system with monolithic LSI space. We also confirm that the adjustment of unknown words is effective.

Table 1. Plural Spaces V.S. Monolithic Space

	Rank 1(%)	Within Rank 3(%)
Plural Spaces	47.8	63.9
Plural Spaces with Adjustment	59.4	78.2
Monolithic Space	58.2	75.7

10 Experimental Results of NTCIR2 J-E and E-J tasks

Our experimental results of NTCIR2 J-E and E-J tasks are shown in Table 2. In this tables, the label 'Desc' means that the DESCRIPTION field of topic is only used as query. On the other hand, the label 'Desc-Nar' means that both of the DESCRIPTION and NARRATIVE fields are used. The label 'Submitted' means that it is one of results submitted to NTCIR2 committee. The label 'Bug-fixed' means that it is one of results after the coding mistake in our system are corrected. Because of the bug, our system incorrectly ignored the sequence of alphabet in Japanese documents. The label 'Adjustment' shows that it is one of results after we introduce the adjustment proposed in Section 6.

Since LSI is sensitive to unknow words in queries, we also examine the result of topics without unknown words as shown in Table 3. All keywords extracted from those topics can be found in dual-language documents used for building LSI spaces.

ted as a query. Then, we examine the retrieval rank of its 'mate', namely, the paired document of it. If the average rank of retrieval is high, the method can be regarded as effective.

Table 2. Experimental Results of all Topics

	Average precision	R-Precision
J-E-Desc Submitted	0.0367	0.0408
J-E-Desc Bug-fixed	0.0533	0.0635
J-E-Desc Adjustment	0.0666	0.0786
Gain by Adjustment	24.9 pt	23.8 pt
J-E-Desc-Nar Submitted	0.0682	0.0852
J-E-Desc-Nar Bug-fixed	0.0868	0.1031
J-E-Desc-Nar Adjustment	0.0940	0.1096
Gain by Adjustment	8.3 pt	6.3 pt
E-J-Desc Submitted	0.0399	0.0575
E-J-Desc Bug-fixed	0.0512	0.0705
E-J-Desc Adjustment	0.0610	0.0839
Gain by Adjustment	19.1 pt	19.0 pt
E-J-Desc-Nar Submitted	0.0495	0.0757
E-J-Desc-Nar Bug-fixed	0.0609	0.0876
E-J-Desc-Nar Adjustment	0.0736	0.1018
Gain by Adjustment	20.9 pt	16.2 pt

Table 3. Experimental Result of Topics with no Unknown Words

	# of Queries without unknown words	Average precision	R-precision
J-E-Desc Bug-fixed	43	0.0600	0.0704
J-E-Desc Adjustment	43	0.0743	0.0870
Gain by Adjustment		23.8 pt	23.6 pt
J-E-Desc-Nar Bug-fixed	31	0.1032	0.1206
J-E-Desc-Nar Adjustment	31	0.1094	0.1307
Gain by Adjustment		6.0 pt	8.4 pt
E-J-Desc Bug-fixed	43	0.0579	0.0786
E-J-Desc Adjustment	43	0.0692	0.0942
Gain by Adjustment		19.5 pt	19.8 pt
E-J-Desc-Nar Bug-fixed	39	0.0738	0.1025
E-J-Desc-Nar Adjustment	39	0.0872	0.1187
Gain by Adjustment		18.2 pt	15.8 pt

11 Discussion

In the viewpoint of absolute effectiveness of information retrieval, we have to say that our system, which is only based on a set of dual-language documents, is less effective than other systems which would be based on translation dictionaries. The best result in NTCIR2 participating systems is above 0.3 in the average precision, while our method achieves only about 0.1. However, we reconfirm that we can construct a large scale CLIR system without dictionaries, if we have a enough number of dual-language documents.

We also confirm that the our adjusting method for unknown words is effective. For example, the average precision of 'J-E-Desc' in Table 3 is improved in 24.9 point. The average precision of 'J-E-Desc-Nar' also rises in 8.3 point. The precision of the retrieval with a query made from the DESCRIPTION field only is improved more than the case that both DESCRIPTION and NARRATIVE fields are used as a query. The reason is that shorter queries were relatively more affected by unknown words.

12 Concluding Remarks

In this paper, we studied the CL-LSI method where a set of dual-language documents is only required to construct translation information for information retrieval. We proposed a way to apply it to a large set of dual-language documents by dividing the set into several subsets and constructing plural LSI spaces. We also study the decline in accuracy of retrieval, which is caused by difference in vocabularies of the LSI spaces. We showed that our solution is effective to solve the problem.

In the viewpoint of absolute effectiveness of retrieval, we have to conclude that our system is less effective than other systems which is based on translation dictionaries. However, we reconfirm that we can construct a large scale CLIR system without dictionaries, if we have a enough number of dual-language documents.

The following problems will be parts of our future works.

- Confirmation of improvement of accuracy by introducing plural LSI spaces

  The experimental result of mate retrieval shows that the division of word spaces is effective to improve the precision of retrieval. However, it is not obvious how effective it is in real retrieval situations like NTCIR2. Additional experiments are needed to confirm the effectiveness.

- Comparison our scheme with other similar methods like Generalized Vector Space Model (GVSM) in the same condition.

  Carbonell et al.[2] show that the effectiveness of CLIR by GVSM, which does not need the SVD process, is comparable to that of CL-LSI method in the middle-scaled experiments with 1134 dual-language documents. We take an interest in the performance of GVSM with the large-scaled dual-language corpus like the corpus of NTCIR1. In order to apply GVSM to a large-scaled corpus, we have to examine how to maintain a large term-document matrix, which is also used in CL-LSI.

- Estimating vector of unknown words

  In the original LSI scheme, a method to estimating vector of unknown words from new documents. In the method, firstly the vectors of new documents are created. In the process all unknown words are ignored. Secondly, the vector of each unknown word is made by sum up the vectors of documents in which the unknown word appears. It would be possible to introduce the estimation to our method.

References

[1] M. Berry, T. Do, G. O'Brien, V. Krishna, and S. Varadhan. *SVDPACKC (Version 1.0) User's Guide*. Computer Science Department, University Tennessee, 1993.

[2] J. G. Carbonell, Y. Yang, R. E. Frederking, R. D. Brown, Y. Geng, and D. Lee. Translingual information retrieval: A comparative evaluation. In *Proceedings of International Joint Conference on Artificial Intelligence '97 IJCAI '97*, 1997.

[3] S. Deerwester, S. T. Dumais, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[4] S. T. Dumais, T. K. Landauer, and M. L. Littman. Automatic cross-linguistic information retrieval using latent semantic indexing. In *SIGIR '96 – Workshop on Cross-Linguistic Information Retrieval*, pages 16–23, 1996.

[5] S. T. Dumais, T. A. Letsche, M. L. Littman, and T. K. Landauer. Automatic cross-language retrieval using latent semantic indexing. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, Mar. 1997.

[6] K. Frantzi and A. S. Extracting nested collocations. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING 96)*, pages 41–46, Aug. 1996.

[7] D. A. Hull and G. Grefenstette. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of SIGIR '96: 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–57, 1996.

[8] G. Kikui. Retrieving Documents Across Language-Barriers. *Journal of Japanese Society for Artificial Intelligence*, 15(4):550–558, July 2000.

[9] T. Kurohashi and M. Nagao. *Japanese Morphological Analysis System JUMAN version 3.61 Manual*. Kyoto University, 1998. (in Japanese).

[10] S. Leach. Singular valued decomposition — a primer. Department of Computer Science, Brown University.

[11] D. W. Oard and B. J. Dorr. A survey on multilingual text retrieval. Technical Report UMIACS-TR-96-19 CS-TR-3615, University of Maryland, Apr. 1996.