Cross-Language Information Retrieval at ULIS

Atsushi Fujii **Tetsuya Ishikawa**

University of Library and Information Science
1-2 Kasuga Tsukuba 305-8550, JAPAN
{fujii,ishikawa}@ulis.ac.jp

Abstract

This paper evaluates the latest version of our cross-language information retrieval (CLIR) system. Our system first translates a given query into the target language, and then retrieves documents relevant to the translated query. We use bilingual dictionaries to derive possible translations and collocational statistics to resolve translation ambiguity. To enhance the query translation, we use three types of dictionaries. We also compare two different retrieval engines. Our experimental results show that enhancements of query translation and retrieval engine individually improve the CLIR performance.

Keywords: cross-language information retrieval, query translation, multiple dictionaries, transliteration, vector space model, the NACSIS collection

1 Introduction

We have recently developed a Japanese/English cross-language information retrieval (CLIR) system [6, 7, 8], where the user presents queries in one language (J or E) to retrieve documents in another language (E or J)[1]. This paper extensively compares variations of our latest system, using the official version of the "NACSIS" test collection [15].

Since queries and documents are in different languages, CLIR needs a translation phase along with the monolingual retrieval phase. To cope with this problem, three types of CLIR systems have been proposed, i.e., query translation [1, 2, 3, 10, 14, 19], document translation [9, 18] and interlingual representation [2, 5, 20, 23] approaches. Among these approaches, we currently adopt the query translation approach, because we can combine a query translation module and existing monolingual retrieval engines with minimal cost.

[1]Some of our recent publications are available from `http://www.ulis.ac.jp/~fujii/publication.html`.

Consequently, our focus is to translate sequences of content words in user queries rather than all the documents in a given collection. In the NACSIS collection, documents are technical abstracts published by Japanese associations, and thus query descriptions are usually short phrases consisting of one or more technical terms. To sum up, translation of technical terms is a crucial issue within our framework. Through preliminary experiments, we identified problems associated with technical term translation as give below:

- technical terms are often compound word, which can be progressively created simply by combining multiple existing morphemes ("base words"), and therefore it is not entirely satisfactory to exhaustively enumerate newly emerging terms in dictionaries,
- Japanese often represents loanwords (primarily for technical terms and proper nouns) based on its special phonogram called *katakana*, which creates new base words progressively,
- technical compound words sometimes include general base words, such as "*AI shougi*" (*shougi* is a Japanese chess-style game).

To counter the first problem, we use the compound word translation method, which selects appropriate translations based on the probability of occurrence of each combination of base words in the target language [7, 8]. For the second problem, we use a transliteration method, which identifies phonetic equivalents in the target language [7]. Finally, for the third problem we combine technical term and general word dictionaries to enhance our bilingual dictionary for base words.

2 CLIR System Overview

Figure 1 depicts the overall design of our CLIR system, where most components are the same as those for monolingual IR, excluding "translator".

First, "tokenizer" processes target language documents ("doc in T") to produce an inverted file ("surrogate"). Since our system is bidirectional, tokenization differs depending on the target language. In the case where documents are in English, tokenization involves eliminating stopwords and identifying root forms for inflected words, for which we used "WordNet" [17]. On the other hand, we segment Japanese documents into lexical units using the "ChaSen" morphological analyzer [16] and discard stopwords. In the current implementation, we use word-based uni-gram indexing for both English and Japanese documents. At the same time, we extract the "collocation" of content words from outputs of the tokenizer, which is used for the subsequent query translation.

Thereafter, the "translator" processes a source language query ("query in S") to output the translation ("query in T"), using the "dictionary".

Finally, the "IR engine" outputs top 1,000 documents according to the similarity between the translated query and each document, in descending order. We compare two different IR engines. The first engine is a naive implementation of the vector space model with TF·IDF term weighting [22], which we also used in the previous experiment [7]. The second engine is the SMART retrieval system [21], which is applied only to the retrieval of *English* documents. In practice, the SMART system includes the tokenization process. However, note that we use the "tokenizer" to extract the "collocation" disregarding which IR engine is selected.

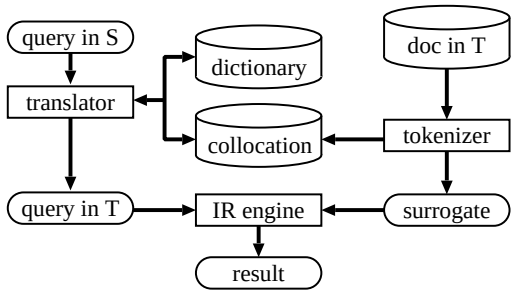

Figure 1: The overall design of our CLIR system

3 Query Translation

3.1 Overview

Given a query in the source language, tokenization is first performed as for target documents (see Figure 1). To put it more precisely, we use WordNet and ChaSen for English and Japanese queries, respectively. We then discard stopwords and extract only content words. Here, "content words" refer to both single and compound words. Let us take the following query as an example:

improvement of data mining methods.

For this query, we discard "of", to extract "improvement" and "data mining methods".

Thereafter, we translate each extracted content word individually. We currently do not consider relation (e.g. syntactic relation and collocational information) between content words. We translate each content word on a word-by-word basis, maintaining the word order in the source language. Note that a preliminary study showed that approximately 95% of compound technical terms defined in a bilingual dictionary maintain the same word order in both source and target languages.

In the case of Japanese-English translation, we consider all possible segmentations of the input content word, by consulting our dictionaries, because Japanese compound words lack lexical segmentation. Then, we select such segmentations that consist of the minimal number of base words. During the segmentation process, we derive all possible translations for base words. For this purpose, we use different types of dictionaries, i.e., "technical term", "general term" and "transliteration" dictionaries. However, a preliminary study showed that the number of translation candidates is great and thus computational cost can be prohibitive, when we consult multiple dictionaries simultaneously. Therefore, we consult those dictionaries *sequentially*. To put it more precisely, the second (third) dictionary is used only when base words unlisted in the first (second) dictionary are found. However, the transliteration dictionary is used only for *katakana* base words. On the other hand, in the case of English-Japanese translation, the transliteration dictionary is used for any unlisted base word (including the case where the input English word consists of a single base word).

After deriving possible translations for base words, we resolve translation ambiguity using a probabilistic model. The formula for the source compound word and one translation candidate are represented as below.

$$S = s_1, s_2, \ldots, s_n$$
$$T = t_1, t_2, \ldots, t_n$$

Here, s_i and t_i denote i-th base words in source and target languages, respectively. Our task, i.e., to select T which maximizes $P(T|S)$, is transformed into

Equation (1) through use of the Bayesian theorem.

$$\arg\max_T P(T|S) = \arg\max_T P(S|T) \cdot P(T) \quad (1)$$

$P(S|T)$ and $P(T)$ are approximated as in Equation (2).

$$\begin{aligned} P(S|T) &\approx \prod_{i=1}^{n} P(s_i|t_i) \\ P(T) &\approx \prod_{i=1}^{n-1} P(t_{i+1}|t_i) \end{aligned} \quad (2)$$

We estimate $P(t_{i+1}|t_i)$ based on the "collocation" in Figure 1. For the estimation of $P(s_i|t_i)$, we use the correspondence frequency for each combination of s_i and t_i in our dictionaries.

It should be noted that our query translation method uses a target document collection and bilingual dictionary *independent* of the collection. This feature provides a salient contrast to other methods which rely on bilingual document collections for the query translation [3, 14, 19].

In the following three sections, we will explain the way to produce our dictionaries, which can be summarized in Figure 2.

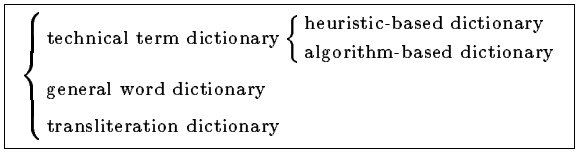

Figure 2: Different types of bilingual dictionaries used for the query translation

3.2 Technical Term Dictionary

For the production of our technical term dictionary for base words, we used the EDR technical terminology dictionary [13], which includes 120,000 English-Japanese translations related to the information processing field. Since most of the entries are compound words, we need to segment Japanese compound words and correspond English-Japanese translations on a word-by-word basis. However, the complexity of segmenting Japanese terms becomes multiply greater as the number of component base words increases. In consideration of these factors, we extracted 59,533 English words consisting of *two* base words, and their Japanese translations[2]. We then used two different segmentation methods.

The first method is the same as performed in our previous experiment [7], that is, we used simple heuristics to segment Japanese compound words into two parts. Those heuristics rely mainly on Japanese character types (i.e., *kanji*, *katakana*, *hiragana*, alphabets and other characters like numerals). As a result, we extracted approximately 24,000 Japanese base words and 8,000 English base words from the EDR dictionary. We shall call this "heuristic-based dictionary".

For the second method, we used the segmentation method for Japanese compound words we previously proposed [6]. Our method is based on the assumption that the set of technical terms in a given domain should consists of as small number of base words as possible [11]. To put it more precisely, we first identified heuristics that segment Japanese compound words with a great confident, as given below:

- `C+K+` ⇒ `C+/K+`,
- `C+A+` ⇒ `C+/A+`,
- `A+K+` ⇒ `A+/K+`,
- `K+A+` ⇒ `K+/A+`,
- `KKKK` ⇒ `KK/KK`.

Here, `C`, `K` and `A` denote *kanji*, *katakana* and alphabet characters, respectively. While symbol "`X`" denotes a single character, "`X+`" denotes one or more characters. Thereafter, using the resultant base words as a seed, we segment remaining compound words, so that the total number of base word types becomes minimum[3]. For this purpose, we implemented an efficient algorithm which requires $O(N)$ computational cost, given that N is the total number of compound words. The resultant base word set, the "algorithm-based dictionary", contains marginally smaller entries than that for the heuristic-based dictionary.

3.3 General Word Dictionary

We produced our general word dictionary from the EDR bilingual dictionary [12]. However, unlike the case of the technical terms, it is not necessarily feasible to segment general compound words into base words (e.g. "hot dog"). Therefore, we simply use single word entries (i.e., words that consist of a single base word).

[2]The number of base words can easily be identified based on English words, while Japanese compound words lack lexical segmentation.

[3]A preliminary study showed that the accuracy of our segment method is approximately 93%.

3.4 Transliteration Dictionary

Figure 3 shows example correspondences between English and (romanized) *katakana* words, where we insert hyphens between each *katakana* character for enhanced readability. The basis of our transliteration method is analogous to that for compound word translation described in Section 3.1. The formula for the source word and one transliteration candidate are represented as below.

$$S = s_1, s_2, \ldots, s_n$$
$$T = t_1, t_2, \ldots, t_n$$

However, unlike the case of compound word translation, s_i and t_i denote i-th "symbols" (which consist of one or more letters), respectively. Note that we consider only such T's that are indexed in the inverted file, because our transliteration method often outputs a number of incorrect words with great probabilities. Then, we compute $P(T|S)$ for each T using Equations (1) and (2) (see Section 3.1), and select k-best candidates with greater probabilities. The crucial content here is the way to produce the transliteration dictionary, i.e., a bilingual dictionary for *symbols*. For this purpose, we used approximately 3,000 *katakana* entries and their English translations listed in our technical term dictionary. To illustrate our dictionary production method, we consider Figure 3 again. Looking at this figure, one may notice that the first letter in each *katakana* character tends to be contained in its corresponding English word. However, there are a few exceptions. A typical case is that since Japanese has no distinction between "L" and "R" sounds, the two English sounds collapse into the same Japanese sound. In addition, a single English letter corresponds to multiple *katakana* characters, such as "x" to "*ki-su*" in "<text, *te-ki-su-to*>". To sum up, English and romanized *katakana* words are not exactly identical, but *similar* to each other.

We first manually define the similarity between the English letter e and the first romanized letter for each *katakana* character j, as shown in Table 1. In this table, "phonetically similar" letters refer to a certain pair of letters, for which we identified approximately twenty pairs, such as "L" and "R". We then consider the similarity for any possible combination of letters in English and romanized *katakana* words, which can be represented as a matrix, as shown in Figure 4. This figure shows the similarity between letters in "<text, *te-ki-su-to*>". We put a dummy letter "$", which has a positive similarity only to itself, at the end of both English and *katakana* words. One may notice that matching plausible symbols can be seen as finding the path which maximizes the total similarity from the first to last letters. The best path can easily be found by, for example, Dijkstra's algorithm [4]. From Figure 4, we can derive the following correspondences: "<te, *te*>", "<x, *ki-su*>" and "<t, *to*>". The resultant correspondences contain 944 Japanese and 790 English symbol types, from which we also estimated $P(s_i|t_i)$ and $P(t_{i+1}|t_i)$, for the transliteration.

English	*katakana*
system	*shi-su-te-mu*
mining	*ma-i-ni-n-gu*
data	*dee-ta*
network	*ne-tto-waa-ku*
text	*te-ki-su-to*
collocation	*ko-ro-ke-i-sho-n*

Figure 3: Examples of English-*katakana* correspondence

Table 1: The similarity between English and Japanese letters

condition	similarity
e and j are identical	3
e and j are phonetically similar	2
both e and j are vowels or consonants	1
otherwise	0

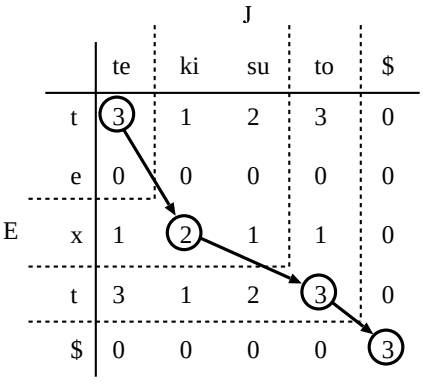

Figure 4: An example matrix for English-Japanese symbol matching (arrows denote the best path)

4 Experimentation

This section investigates the performance of our CLIR system based on the NACSIS workshop evaluation method: each system retrieves 1,000 top documents from the NACSIS collection, and 11-point average non-interpolated precisions were calculated (using the TREC evaluation software).

The NACSIS official collection consists of 39 Japanese queries and approximately 330,000 documents (in either a combination of English and Japanese or either of the languages individually), collected from technical papers published by 65 Japanese associations for various fields. Each document consists of the document ID, title, name(s) of author(s), name/date of conference, hosting organization, abstract and keywords, from which titles, abstracts and keywords were used for our evaluation. We used as target documents approximately 187,000 entries where abstracts are in both English and Japanese. Each query consists of the title of the topic, description, narrative and list of synonyms, from which we used only the *description*. Relevance assessment was performed based on one of the three ranks of relevance, i.e., "relevant (A)", "partially relevant (B)" and "irrelevant (C)".

We compared four J-E CLIR systems using different query translation methods, (1)–(4), and monolingual IR system (5) as a baseline for CLIR:

(1) only technical term dictionary is used,

(2) technical term and transliteration dictionaries are used,

(3) technical term and general word dictionaries are used,

(4) technical term, transliteration and general word dictionaries are used,

(5) J-J monolingual IR.

Note that (as explained in Section 3.1) in systems (2), (3) and (4), we use the subsequent dictionary only when base words unlisted in the preceding dictionary are found. We also compared the heuristic/algorithm-based technical term dictionaries. We then varied the number of translation candidates used for the retrieval ($k = 1, 3, 10$). Finally, we compared two retrieval engines, i.e., the naive vector space model and SMART systems. From Tables 2 to 5, we show average (non-interpolated) precisions for different systems. Note that in Tables 3 and 5 the average precisions for J-J IR are not available, because the SMART system is not implemented for the retrieval of Japanese documents. It should also be noted that in Tables 2 and 4 the average precisions for system (3) with $k = 10$ are not available because of a memory problem in the naive VSM engine.

We summarize what can be derived from these results as follows. First, the performance was improved as the number of dictionaries used increases. Although the general word and transliteration dictionaries individually improve on the performance obtained only with the technical term dictionary, when used together the improvement was even greater. Second, in most cases the performance obtained with $k = 1$ was marginally higher than that for other values of k. Third, the algorithm-based dictionary is more effective than the heuristic-based dictionary with respect to CLIR performance. To sum up, it is generally observable that system (4) with the algorithm-based dictionary and $k = 1$ outperformed other CLIR systems. In addition, in Table 2 system (4) with the algorithm-based dictionary and $k = 1$ outperformed the system (5), and in Table 4 both systems are quite comparable in performance. At the same time, note that comparisons between systems (4) and (5) are biased to some extent by the different properties inherent in English and Japanese IR.

Finally, comparing Tables 2 and 3 (or Tables 4 and 5), one can see that the use of the SMART system improved the performance obtained with the naive VSM search engine. It is expected that using a more sophisticated IR engine, our CLIR system will achieve a higher performance independent of the query translation method.

5 Conclusion

In this paper, we evaluated the performance of the latest version of our cross-language information retrieval (CLIR) system. We combined a query translation module, which uses different types of bilingual dictionaries, with different monolingual retrieval engines. Our experimental results showed that the combination of technical term, general word and transliteration dictionaries outperformed other dictionary combinations. We also showed that our CLIR system achieves a higher performance using a more sophisticated retrieval engine.

References

[1] Lisa Ballesteros and W. Bruce Croft. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21th Annual International*

Table 2: Average non-interpolated precision for A documents using the naive VSM

	heuristic-based dic			algorithm-based dic		
	$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$
(1)	.0368	.0415	.0459	.0513	.0553	.0485
(2)	.0543	.0560	.0610	.0689	.0701	.0629
(3)	.0630	.0639	.0586	.0738	.0730	—
(4)	.0764	.0766	.0817	**.0872**	.0868	.0774
(5)	.0675					

Table 3: Average non-interpolated precision for A documents using the SMART

	heuristic-based dic			algorithm-based dic		
	$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$
(1)	.0612	.0602	.0575	.0779	.0678	.0626
(2)	.0716	.0702	.0662	.0863	.0773	.0655
(3)	.0882	.0851	.0832	.1025	.0960	.0879
(4)	.0958	.0938	.0907	**.1103**	.1047	.0895
(5)	—					

Table 4: Average non-interpolated precision for A+B documents using the naive VSM

	heuristic-based dic			algorithm-based dic		
	$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$
(1)	.0420	.0452	.0492	.0590	.0604	.0534
(2)	.0567	.0569	.0614	.0737	.0725	.0651
(3)	.0725	.0713	.0504	.0834	.0805	—
(4)	.0830	.0812	.0867	**.0938**	.0914	.0838
(5)	.0948					

Table 5: Average non-interpolated precision for A+B documents using the SMART

	heuristic-based dic			algorithm-based dic		
	$k=1$	$k=3$	$k=10$	$k=1$	$k=3$	$k=10$
(1)	.0667	.0661	.0638	.0855	.0772	.0703
(2)	.0753	.0742	.0709	.0926	.0848	.0728
(3)	.1043	.1032	.0989	.1168	.1116	.1022
(4)	.1083	.1078	.1026	**.1210**	.1164	.1014
(5)	—					

ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 64–71, 1998.

[2] Jaime G. Carbonell, Yiming Yang, Robert E. Frederking, Ralf D. Brown, Yibing Geng, and Danny Lee. Translingual information retrieval: A comparative evaluation. In *Proceedings of the 15th International Joint Conference on Artficial Intelligence*, pp. 708–714, 1997.

[3] Mark W. Davis and William C. Ogden. QUILT: Implementing a large-scale cross-language text retrieval system. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 92–98, 1997.

[4] Edsgar W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol. 1, pp. 269–271, 1959.

[5] Susan T. Dumais, Thomas K. Landauer, and Michael L. Littman. Automatic cross-linguistic information retrieval using latent semantic indexing. In *ACM SIGIR Workshop on Cross-Linguistic Information Retrieval*, 1996.

[6] Atsushi Fujii and Tetsuya Ishikawa. Japanese word segmentation for producing English-Japanese base word dictionaries. *Information Processing Society of Japan, SIGNL*, Vol. 98, No. 99, pp. 67–72, 1998. (In Japanese).

[7] Atsushi Fujii and Tetsuya Ishikawa. Cross-language information retrieval for technical documents. In *Proceedings of the Joint ACL SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 29–37, 1999.

[8] Atsushi Fujii and Tetsuya Ishikawa. Cross-language information retrieval using compound word translation. In *Proceedings of the 18th International Conference on Computer Processing of Oriental Languages*, pp. 105–110, 1999.

[9] Denis A. Gachot, Elke Lange, and Jin Yang. The SYSTRAN NLP browser: An application of machine translation technology in multilingual information retrieval. In *ACM SIGIR Workshop on Cross-Linguistic Information Retrieval*, 1996.

[10] David A. Hull and Gregory Grefenstette. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 49–57, 1996.

[11] Masahiko Ishii. Economy in Japanese scientific terminology. Hans Czap and Christian Galinski (ed.), *Terminology and Knowledge Engineering*, INDEKS Verlag, pages 123–136, 1987.

[12] Japan Electronic Dictionary Research Institute. Bilingual dictionary, 1995. (In Japanese).

[13] Japan Electronic Dictionary Research Institute. Technical terminology dictionary (information processing), 1995. (In Japanese).

[14] Noriko Kando and Akiko Aizawa. Cross-lingual information retrieval using automatically generated multilingual keyword clusrters. In *Proceedings of the 3rd International Workshop on Information Retrieval with Asian Languages*, pp. 86–94, 1998.

[15] Noriko Kando, Teruo Koyama, Keizo Oyama, Kyo Kageura, Masaharu Yoshioka, Toshihiko Nozue, Atsushi Matsumura, and Kazuko Kuriyama. NTCIR: NACSIS test collection project. In *The 20th Annual BCS-IRSG Colloquium on Information Retrieval Research*, 1998.

[16] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Osamu Imaichi, and Tomoaki Imamura. Japanese morphological analysis system ChaSen manual. Technical Report NAIST-IS-TR97007, NAIST, 1997. (In Japanese).

[17] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, Katherine Miller, and Randee Tengi. Five papers on WordNet. Technical Report CLS-Rep-43, Cognitive Science Laboratory, Princeton University, 1993.

[18] Douglas W. Oard and Paul Hackett. Document translation for cross-language text retrieval at the University of Maryland. In *The 6th Text Retrieval Conference*, 1997.

[19] Akitoshi Okumura, Kai Ishikawa, and Kenji Satoh. Translingual information retrieval by a bilingual dictionary and comparable corpus. In *The 1st International Conference on Language Resources and Evaluation, Workshop on Translingual Information Management: Current Levels and Future Abilities*, 1998.

[20] Gerard Salton. Automatic processing of foreign language documents. *Journal of the American Society for Information Science*, Vol. 21, No. 3, pp. 187–194, 1970.

[21] Gerard Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice-Hall, 1971.

[22] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[23] Páraic Sheridan and Jean Paul Ballerini. Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 58–65, 1996.