Multilingual Information Processing for Digital Libraries

Akira Maeda

Department of Computer Science, Ritsumeikan University
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan
E-mail: amaeda@cs.ritsumei.ac.jp

Abstract

This paper presents some solutions to the problems in order to realize a digital library which can handle multilingual documents in a unified manner. Specifically, we focus on techniques such as: 1) display and input functions for multilingual text which does not depend on installed fonts and input methods on the client side, 2) an algorithm for the automatic identification of the languages and the character coding systems of documents, 3) a cross-language information retrieval technique, which is suitable for documents in diverse domains. By integrating these three techniques, we realized a system which supports access to documents written in languages other than the user's native language. This system provides some solutions to the problems in multilingual information processing that are specific to the Internet and digital libraries.

1 Introduction

With the increasing popularity of the Internet in various part of the world, the languages used for Web documents are expanded from English to various languages. Also, since libraries are inherently multilingual, multilingual document environment is crucial for digital libraries. However, there are many unsolved problems in order to realize a digital library which can handle such multilingual documents in a unified manner.

From the user's point of view, three most fundamental text processing functions for the general use of a digital library are *display*, *input*, and *retrieval* of the text. However, for languages such as Japanese, Chinese, and Korean, character fonts and input methods that are necessary for displaying and inputting texts, are not always installed on the client side.

From the system's point of view, one of the most troublesome problems for handling multilingual Web documents is that, many Web documents do not have meta information of the character coding system and the language used for the document itself, although character coding systems used for Web documents vary according to the language. It may result in troubles such as incorrect display on Web browsers, and inaccurate indexing on Web search engines. Also, other text processing applications such as categorization, summarization, and machine translation are dependent on identifying the language of the text to be processed.

Moreover, there might be some cases where the user wants to retrieve documents in unfamiliar languages, especially for cases where information written in languages other than the user's native language is rich. The needs for retrieving such information must not be small. Consequently, research on cross-language information retrieval (CLIR), which is a technique to retrieve documents written in one language using a query written in another language, is being paid much attention. However, it is difficult to achieve adequate retrieval effectiveness for a document collection in diverse languages and domains, such as the Internet and digital libraries.

The goal of this paper is to provide some solutions to these problems. Specifically, we focus on techniques such as:

1. display and input functions for multilingual text which does not depend on installed fonts and input methods on the client side,
2. an algorithm for the automatic identification of the languages and the character coding systems of documents,
3. a cross-language information retrieval technique, which is suitable for documents in diverse domains, based on the word co-occurrence information.

By integrating these three techniques, we realized a system which supports access to documents written in languages other than the user's native language. This system provides some solutions to the problems in multilingual information processing that are specific to the Internet and digital libraries[4].

2 MHTML: Display and Input Functions for Multilingual Documents

2.1 Introduction

In order to realize true international information sharing on the Internet, it is important to be able to read and retrieve documents from every part of the world, in the same manner. Recently, WWW browsers that support Unicode are becoming popular, and operating systems are becoming to support Unicode as the internal character code. Conse-

quently, it is much easier today to handle multilingual documents than before.

However, for languages such as Japanese, Chinese, Korean, and some minor languages, character fonts and input methods that are necessary for displaying and inputting texts, are not always installed on the client side. In ordinary PC environments, usually only fonts for the native language in addition to English are installed by default. Therefore, in order to display documents in other languages, the user has to install additional fonts for that language. This process is rather a hard task for casual users of PCs and the Internet. Moreover, even if a multilingual coding system such as Unicode and the browsers that support it become popular, it is not realistic to prepare fonts for all languages supported by Unicode in every client, especially for small clients with a limited storage, such as PDAs (Personal Digital Assistants) and mobile phones.

The simplest and the easiest solution to this problem is to realize a browser that does not require fonts on the client side. That is, display of a text will be possible regardless of the installed fonts on the client, by transmitting character glyphs instead of character codes. In this approach, it is inevitable that the number of bytes to be transmitted will increase, but on the other hand, it can avoid the problem of installation and storage of fonts. Incidentally, it can be used to display characters that are not included in existing character sets. For practical methods to implement such an approach utilizing an existing Web browser, we can think of following methods:

1. Convert the source document into a set of page images,
2. Replace each character or string in the source document into an inline image,
3. Add fonts for characters that appeared in the source document, and use that fonts to display the text (Figure 1).

The method 1 is a low-cost and practical method for digitizing existing paper documents, and used in many existing digital library systems. Hyperlinks can be implemented by using CGI image maps or servlets. The method 2 is already implemented as a conversion service of existing Web documents, i.e. *Shodouka*[1]. This method has an advantage that it can utilize the HTML layout engine of Web browsers. The method 3 is the one we propose. It converts the source HTML document into a format called MHTML in which fonts for appeared characters are added, and it is displayed in a Java applet which runs on a Web browser.

We have developed a browsing system which enables viewing multilingual HTML documents using this technique[5], and it is extended to implement the text input function. In this section, we introduce this technique in detail, and show some applications, such as an OPAC, a collection of Japanese folk tales[2], and an SGML-based text retrieval system.

[1]http://www.shodouka.com/

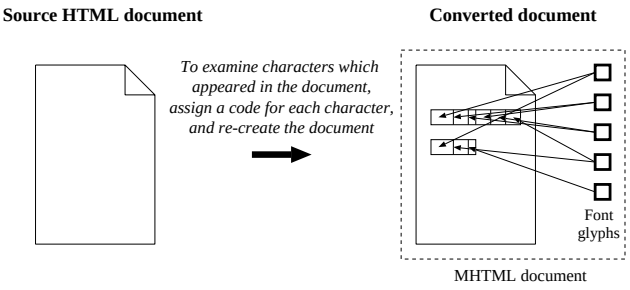

Figure 1: The method of adding fonts for characters appeared in the source document.

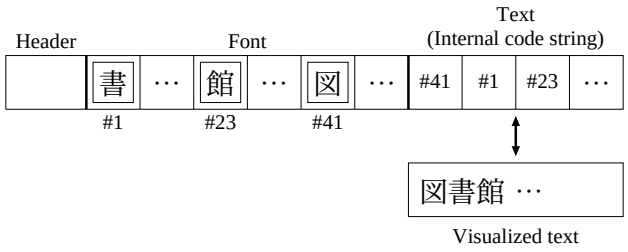

Figure 2: Outline of the MHTML document.

2.2 MHTML (Multilingual-HTML) Document Format

2.2.1 Overview of MHTML

Figure 2 shows the outline of the display mechanism of MHTML. An MHTML document contains a source HTML text and a minimum set of font glyphs. The character codes of the source text, except for HTML tags, are internalized to the document itself, so that the codes are effective only in the document.

Figure 3 shows the mechanism of displaying a document using MHTML. First, WWW browser on the client loads a Java applet called MHTML class from MHTML server, and the applet is invoked on the browser. Then, two parameters, the URL of the source HTML document (*URL*) and the coding system identifier which indicates the coding system used for the document (*coding system ID*), are sent to the MHTML server (1). The MHTML server fetches the source HTML document indicated by *URL* parameter (2), and converts it into MHTML using *coding system ID* parameter. Finally, the MHTML document is returned to the browser (3), and the document is displayed on the MHTML class applet.

2.2.2 Mixture of Multilingual Text and Handling of Gaiji

As described in the previous section, an internal coding system, which is independent from the encoding scheme and the language used in the source HTML document, is used for the character coding system of the text part. As the result, it is possible for MHTML to display a document written in multiple languages by creating an HTML

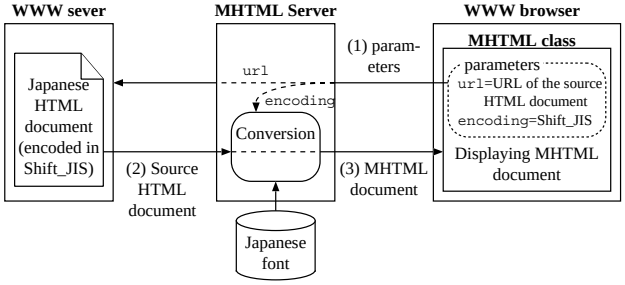

Figure 3: The mechanism of displaying a document using MHTML.

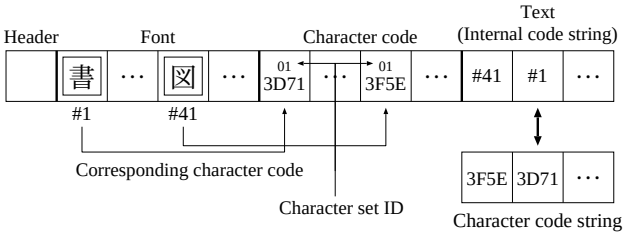

Figure 4: Outline of an extended MHTML document.

document using a multilingual encoding scheme such as ISO-2022-JP-2, and preparing the corresponding fonts on the MHTML server.

Furthermore, it is very easy for MHTML to display *gaiji*[2], which is particularly important for some applications such as OPACs for Asian languages. It only requires preparing a font that contains gaiji and registering it as a new character set in the MHTML server.

2.2.3 Extended MHTML Document Format

An extended version of the MHTML document format is defined in order to realize the text input function, which is required for HTML forms, etc. In the extended MHTML document format, an identifier of the character encoding scheme and the character code are added for each character in the document, in addition to the components of the basic MHTML document format shown in Figure 2. Figure 4 shows the outline of an extended MHTML document.

A source character code can be reproduced from an extended MHTML document by replacing every character in the internalized text with its corresponding source character code. The character encoding identifier is required to make the re-converted text conform to ISO-2022-JP-2. Conversion to Unicode is also possible.

We developed a Japanese text input system which does not require fonts on the client using this format. The information of the character code is added to the basic MHTML format, because it is needed for the server, for example a search engine, to interpret the submitted string.

[2]Gaiji refers to characters that are not included in existing character sets.

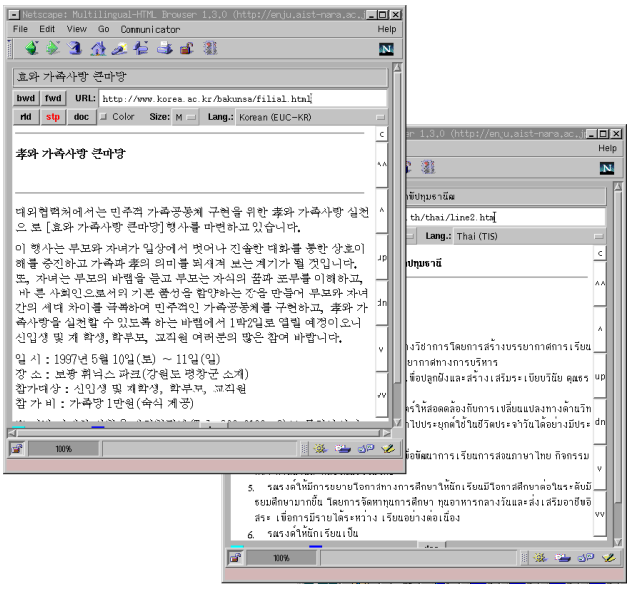

Figure 5: An example of viewing documents using the browsing service.

2.3 Example Applications Based on MHTML Technique

2.3.1 Browsing Service

Since 1996, we have been offering a browsing service which enables users browse Web documents without requiring fonts. This service has been opened to the public and freely accessible[3].

The MHTMLViewer class, which is an extension to MHTML class, is used for this service. In the MHTMLViewer class, URL input field, navigation buttons, and a pull-down menu to choose the document language, are added in addition to the core functionalities of the MHTML class. This service provides users with easy access to Web documents written in, for example Japanese, from anywhere in the world, regardless of whether the fonts are installed or not.

Figure 5 shows an example of viewing Korean and Thai documents using the browsing service.

2.3.2 Japanese Old Tales Collection

We applied MHTML to develop the user interface for a multilingual electronic text collection of Japanese old folk tales[4]. The tales were taken from Japanese old tales and rewritten by volunteer authors. Each tale is written in one or more languages, including English, French, German, Spanish, Portuguese, Italian, Bulgarian, Russian, Japanese, Chinese, and Korean.

Figure 6 shows an example of viewing a Japanese old tale in three languages in parallel. In this example, three MHTML class applets are embedded in a table.

[3]http://mhtml.ulis.ac.jp/
[4]http://www.DL.ulis.ac.jp/oldtales/

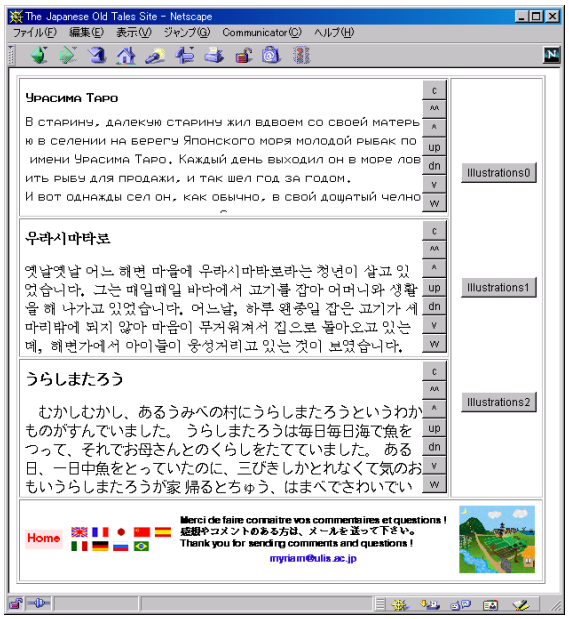

Figure 6: An example of displaying a Japanese old tale in the collection.

2.4 Summary

In this section, we proposed a system for providing multilingual HTML documents without requiring to install fonts on the client. This system might be especially useful for documents written in minor languages, for old literature which may include characters that are not defined in any standard character set, and for newspaper articles which may require gaiji's for some proper nouns.

Despite the name of MHTML, it is entirely independent from HTML. Therefore, it can also be used to encapsulate any text formats that are based on plain text, such as electronic mail and XML (eXtensible Markup Language). For example, it might be useful for a Japanese user to be able to read and write electronic mails in Japanese from a public Web terminal (e.g. in a public library) anywhere in the world.

3 Automatic Identification of Coding Systems and Languages of Documents

3.1 Introduction

With the growth of the Internet and WWW in recent years, documents written in various languages are being provided. In proportion to this growth, character coding systems used for Web documents also keep on increasing.

However, Web documents often lack information on the language and coding system they use. A mechanism to indicate those information to Web documents is already developed, but it is not very popular at this moment. In such a case that those information are not indicated, recipients of the document have to presume the coding system and language used. And if the presumption is wrong, it might result in troubles such as incorrect display on Web browsers. Besides, Web search engines create an index for collected documents, and lack of those information might cause inaccurate indexing. Even in search engines that support only one language, if the language were not identified for collected documents, it is most likely that documents in other languages will also be indexed, and it might cause a significant loss in precision of retrieved documents. Furthermore, in addition to search engines, other document processing applications such as categorization, summarization, and machine translation are also dependent on knowing the document language. Of course, coding system has to be identified before identifying the document language.

Although the use of multilingual coding systems such as ISO/IEC 10646-1 and Unicode will hopefully avoid such problems, it will take a long time for existing enormous amount of documents to be replaced by such coding systems. Therefore, at least in the present circumstances, languages and coding systems have to be identified prior to processing, for Web documents to be correctly processed.

In this section, we propose a method which automatically and efficiently identifies both languages and coding systems of documents by employing a simple statistical technique and heuristics together[6].

Table 1: Target languages/coding systems for identification.

Coding System	Language	Unit	Character range
ASCII	English	7	33–126
ISO-2022-JP	Japanese	7	33–126
ISO-2022-CN	Chinese	7	33–126
ISO-2022-KR	Korean	7	33–126
Shift_JIS	Japanese	8	33–252
EUC-JP	Japanese	8	33–126, 142–254
GB2312	Chinese (simplified)	8	33–126, 161–254
Big5	Chinese (traditional)	8	33–126, 161–254
EUC-KR	Korean	8	33–126, 142–254
ISO-8859-1	European languages	8	33–126, 161–254

3.2 Target Languages and Coding Systems

Our proposed method supports 12 languages and 10 coding systems as shown in Table 1. For ISO-8859-1 coding system, it supports 9 languages, namely, English, German, French, Italian, Spanish, Portuguese, Danish, Norwegian, and Swedish. Although ISO-8859-1 supports 14 languages, we only support 9 languages because of availability of training data for other languages. Besides, Chinese coding systems GB2312 and Big5 handle different coded character sets, we regard these as different languages.

In the table, "Unit" indicates the number of bits actually used for one octet, and "Character range" indicates the code range within which graphic characters are assigned.

Although the target coding systems except for ASCII and ISO-8859-1 are all variable length multi-byte coding systems, our algorithm treats all coding systems as one byte code, regardless of the segments of multi-byte characters, for simplicity and efficiency.

3.3 Training Data

3.3.1 7-Bit Coding Systems

Because ASCII and ISO-2022 variants are 7-bit coding systems, those can be distinguished from other 8-bit coding systems by checking if the document contains a byte in which MSB (Most Significant Bit) is 1. Moreover, documents written in ISO-2022 variants necessarily contain some escape sequences that is used to designate (and to invoke) a coded character set to be used after it. Therefore, subsets of ISO-2022 (-JP, -CN, and -KR) can be distinguished by checking those escape sequences.

Consequently, for all 7-bit coding systems mentioned, it is possible to infallibly identify those coding systems without using any training data.

3.3.2 8-Bit Coding Systems

8-bit coding systems are identified by analyzing the distributions of character codes for every one byte or every consecutive two bytes.

For the training data for analysis, we used the directory pages of country/language versions of Yahoo! directory service.

We extracted only the descriptions of sites from all directory pages of each country/languages version, in order to avoid fixed descriptions that are identical among all pages. From the extracted sentences, randomly selected 100 kilobytes were used for the training data. For the training data of Japanese Shift_JIS, sentences converted from EUC-JP using a coding system converter were used.

One byte code distributions of the training data for some 8-bit coding systems are shown in Figure 7. In these figures, the horizontal axis is the one byte code values (from 33 to 254), and the vertical axis is the occurrence frequencies for each code values.

Some characteristics in distributions of code occurrence frequencies for each coding system can be observed from these figures. For example, in Shift_JIS and EUC-JP, the first bytes of hiragana and katakana, that are frequently appeared in Japanese text, converge into a particular code range (i.e. 164-165 for EUC-JP), and are relatively high values. Similarly, in EUC-JP, code ranges for the first bytes of Greek and Cyrillic, and unused characters (166-174) are relatively low values compared to that of other two-byte codes (161-254).

However, no significant characteristics were observed among GB2312, Big5, and EUC-KR, and among languages of ISO-8859-1.

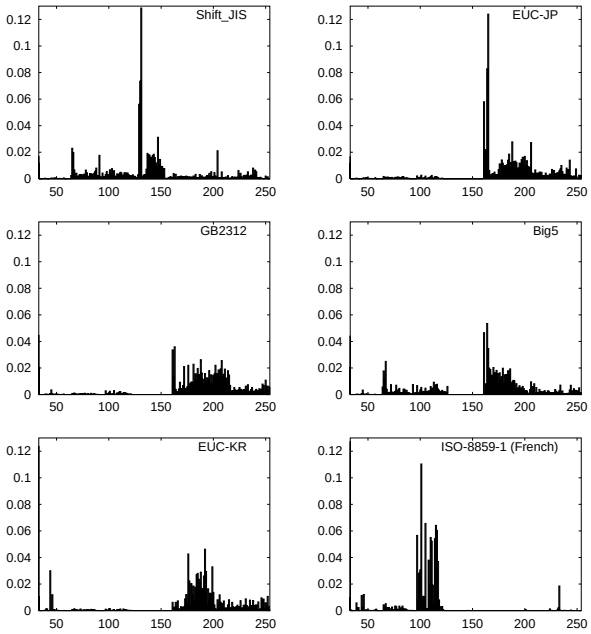

Figure 7: One byte code distribution of 8-bit coding systems.

3.4 Automatic Identification Algorithm

3.4.1 Parameter Method

This method is applied for Shift_JIS and EUC-JP, in which some significant characteristics were observed, and all 7-bit coding systems. In the method, firstly the frequencies of code values (0-255) for each one byte in the document are calculated. Then, identification algorithm, which is derived heuristically from the statistical characteristics of the frequency distributions of training data, is applied.

For the identification of Shift_JIS and EUC-JP, significant characteristics observed in the one byte code distributions are used. Specifically, we assume that the frequency of the code value of the first byte of hiragana or katakana is always the highest for documents written in Shift_JIS or EUC-JP. Therefore, we can set the threshold value to be the minimum frequency of the code that has the highest frequency in each document, among the training data for these coding systems. Similarly, we can also use relatively less appeared code value, such as the first byte of Greek and Cyrillic characters in EUC-JP. In this case, the maximum average frequency of the code range of these characters in each document in the training data is used. In addition, since Shift_JIS uses a code range that is not used in other coding systems (128-159), it can also be used for the identification.

ASCII is not identified here despite that it is a 7-bit coding system, because it cannot be distinguished from ISO-8859-1 documents that do not contain any accented alphabets.

3.4.2 Vector-Distance Method

On the other hand, if we regard the frequencies of code values as a vector, we can consider an alternative method for identification by comparing their distances. In the method, in the same way as the parameter method, firstly the frequencies of code values (0-255) for each one byte in the document are calculated. Then, that vector is compared with vectors of each language and coding systems that are calculated from the training data in advance, in terms of

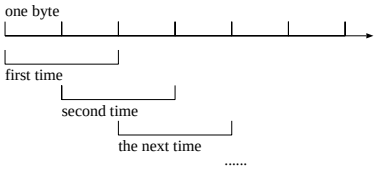

Figure 8: Statistics using consecutive two bytes.

their distance.

For 7-bit coding systems, ASCII and ISO-2022 are distinguished by checking the presence of a escape sequence, as described in section 3.3.1. For 8-bit coding systems, vector-distances between the target document and each languages and coding systems are calculated, and the language and coding system which have the minimum distance is selected as the result.

If we take the connection of consecutive characters into account, we can consider a method using consecutive two bytes as a unit instead of one byte unit. In this case, as shown in Figure 8, consecutive two bytes that slide simply one byte at a time from the beginning, regardless of the number of bytes used for one character. This method might catch the characteristics of connections of characters, that are dissimilar among languages.

The vector-distance D_c between the document to be identified and the training data of the coding system c, is calculated by the following formula based on *cosine distance*:

$$\cos D_c = \frac{\sum_i \sum_j freq_c(i,j)\, freq_d(i,j)}{\sqrt{\sum_i \sum_j freq_c(i,j)^2} \sqrt{\sum_i \sum_j freq_d(i,j)^2}} \quad (1)$$
$$(i, j = 32, 65 \ldots 90, 97 \ldots 122, 128 \ldots 255)$$

where $freq_c(i,j)$ is the occurrence frequency of the code value $i \times 256 + j$ for the training data of the coding system c, and $freq_d(i,j)$ is the occurrence frequency of the code value $i \times 256 + j$ for the document to be identified.

The vector-distances D_c between the document to be identified and each training data for the target coding systems are calculated, and the coding system that are the closest to the document is taken as the result of the identification. In order to avoid the negative effect of characters that are not useful for identification such as symbols, numerals, and control characters, the code ranges 0-64, 91-96, and 123-127 are not counted.

3.5 Evaluation

We conducted the experiment of the proposed methods in terms of accuracy. For the test data, we used the same document collection as described in section 3.3.2 (i.e. Yahoo! directory pages), but distinct set of documents to that of the training data. We used 539-1,000 documents that are longer than 300 bytes (1,255 bytes on average) for each coding system and language.

Table 2: Correct rate of identification using the parameter method.

Coding system (language)	Correct rate (%)
ISO-2022-JP (Japanese)	100.0
ISO-2022-CN (Chinese)	100.0
ISO-2022-KR (Korean)	100.0
Shift_JIS (Japanese)	100.0
EUC-JP (Japanese)	100.0
Avg.	100.0

Table 3: Correct rate of identification using the vector-distance method.

Coding system (language)	Correct rate (%)	
	1 byte	2 bytes
Shift_JIS (Japanese)	99.8	100.0
EUC-JP (Japanese)	99.2	100.0
GB2312 (Chinese)	100.0	100.0
Big5 (Chinese)	100.0	100.0
EUC-KR (Korean)	100.0	100.0
ISO-8859-1 (English)	90.9	98.9
ISO-8859-1 (German)	99.4	100.0
ISO-8859-1 (French)	95.1	99.9
ISO-8859-1 (Italian)	98.2	100.0
ISO-8859-1 (Spanish)	90.7	99.9
ISO-8859-1 (Portuguese)	92.8	100.0
ISO-8859-1 (Danish)	69.8	92.6
ISO-8859-1 (Norwegian)	70.1	91.5
ISO-8859-1 (Swedish)	94.9	99.7
Avg.	92.9	98.7

3.5.1 Applying Parameter Method

Correct rates of identification using the parameter method are shown in Table 2.

Identification using the parameter method achieved 100% for all target coding systems.

3.5.2 Applying Vector-Distance Method

Correct rates of identification using the vector-distance method are shown in Table 3.

3.5.3 Discussion of Identification Errors

From the table 3, one byte vector-distance method achieved 92.9% on average, but only achieved about 70% for Danish and Norwegian of ISO-8859-1. Mostly those two languages were mis-identified vice versa; about 25% of Danish were mis-identified as Norwegian, and about 24% of Norwegian were mis-identified as Danish. It is probably because the languages of Danish, Norwegian, and Swedish are very similar in terms of linguistics, and it might be the cause of the mis-identification between Danish and Norwegian. On the other hand, Swedish gained much better results than those two languages, probably due to the difference in the usage of accented letters.

For two bytes vector-distance method, correct rates are improved for almost all of the languages, and achieved

98.7% correct rate in average. Even for Danish and Norwegian, in which the performances were poor for one byte vector-distance method, the correct rates are improved over 20% for both languages.

3.6 Summary

In this section, we proposed a automatic identification method of coding systems and languages of multilingual documents which employs the statistics and its analysis.

In order to verify the effectiveness of the proposed method, we conducted experiments for 12 languages and 10 coding systems, and achieved over 98% correct rate in average. Especially for Asian languages that use multiple bytes to represent one character, we showed that it is possible to identify those languages and coding systems effectively and efficiently, simply using one byte unit and without discriminating the boundaries of characters. For European languages, we showed that it can achieve 98% average correct rate for 9 languages included in ISO-8859-1, by using the consecutive two bytes unit, which takes the connection of consecutive characters into account. Furthermore, we investigated the relation between text length and correct rate, and showed that our method can identify nearly perfect by using only first 300 bytes of text, excluding Danish and Norwegian.

Also, our method has an advantage that it can easily be extended to other languages and coding systems other than the target languages used in this paper, because it does not depend on the byte length of one character, which varies depending on the coding system.

One of the future work for this research is an extension to other languages and coding systems. For example, several different coding systems are used to represent documents in Arabic and Russian, so automatic identification technique for these coding systems is desired. Also, the language identification of documents in which multiple languages are used, which is expected to be increase in proportion to the increasing popularity of Unicode, has to be investigated in the future.

Another possible future work might be an exploration into the automatic creation of discrimination rules for the parameter method. The parameter method has an advantage that it is much more efficient than other methods, but has a disadvantage that the rule has to be manually created by observing the code distribution and finding a significant characteristic for a particular coding system. Therefore, a method to automatically create such rules might be a promising future work.

4 Query Term Disambiguation for Cross-Language Information Retrieval

4.1 Introduction

With the increasing popularity of the Internet in various part of the world, the languages used for Web documents are expanded from English to various languages. Some Web search engines such as AltaVista and Lycos can handle multiple languages in addition to English, and can specify target language of documents to be retrieved. Also, there exists many search engines which handle Web documents written in a particular language other than English. However, these search engines are essentially a collection of monolingual search engines from the user's perspective.

Nevertheless, there might be some cases where the user wants to retrieve documents in unfamiliar or illiterate languages. The needs for retrieving such information must not be small. For example, when Japanese is used for the query language, target collection will be only a very small portion of the whole Web documents, which is said to be several billions of pages. Also, there might be the case, depending on the user's demand, where information written in a language other than the user's native language is rich. For example, for the economic trend of a particular country, there should be an extremely rich information in the language used in that country.

Because of such needs, researches on Cross-Language Information Retrieval (CLIR), a technique to retrieve documents written in a certain language using a query written in another language, have been active in recent years.

Of course, an obvious solution is to translate all Web documents into a particular language in advance. However, considering the enormous amount of Web documents, this approach is unrealistic. Therefore, it is feasible to translate the retrieved documents. Recently, relatively inexpensive machine translation software are becoming to be available, and some of them can translate a Web document and display it on a Web browser on-the-fly. Consequently, even for the case that the user cannot read languages other than his/her native language at all, there are increasing situations that CLIR can be considered to be useful.

Therefore, the problem is how to formulate a query in the target language. In order to satisfy such needs on usual monolingual retrieval system, the user have to manually translate the query by using a dictionary (if the user is not familiar with the target language). This process not only imposes a burden on the user but also might choose incorrect translations for a query, especially for unfamiliar or illiterate languages. Accordingly, the approach in which the user formulates a query in hir/her native language and the system translates it, should be desirable. One major technical problem to be solved in CLIR is how to translate short queries which consists of a few keywords appropriately. Possible translation-candidates may be numerous in such a case and resolving such ambiguities becomes a hard task.

In this section, we propose a novel approach for CLIR system targeting Web documents, which uses natural language resources obtained from a Web search engine as a corpus, and resolves the ambiguities caused by the dictionary-based query translation approach, by using a co-occurrence information[7]. We have evaluated the effectiveness of this method by experiments. Using this method, we do not have to worry about obtaining expensive language resources, which is one of the drawbacks of existing

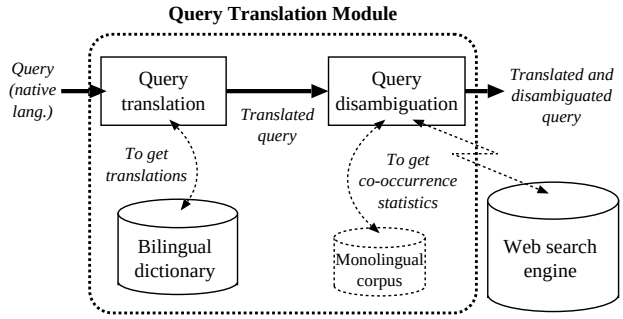

Figure 9: Flow of Query Translation.

CLIR approaches, and it is easier to extend to other languages, as well as it achieves a reasonable retrieval effectiveness.

4.2 Query Translation

Figure 9 shows the flow of query translation for the proposed query term disambiguation method.

A query in user's native language is first translated into the target language using a bilingual dictionary. The obtained translation-candidates are disambiguated using term co-occurrence statistics, and then passed to the search engine.

A query submitted by a user is first segmented into words using a morphological analyzer. Then, each word is translated into the target language using a machine-readable dictionary. In this phase, the longest matched term in the dictionary is used as the translation term. For example, an English query "digital library" can be segmented into "digital" and "library", but if the phrase "digital library" was found in the dictionary, the translation(s) of that phrase will be used instead. For the case of the longest match overlaps, (e.g. "distributed network environment" matches both "distributed network" and "network environment"), both phrases will be used.

Translation candidates obtained from the dictionary are then disambiguated using the method described in the next section, which is based on 2 or *n* words co-occurrence frequency information obtained from the target language corpus.

4.3 Query Term Disambiguation using a Web Search Engine

Since Web search engines gather an enormous volume of documents that cover extensive domains, they might be very useful as a natural language resource. For instance, a document count in a consequence of searching some terms combined by AND operator, can be regarded as a co-occurrence frequency of those terms in a Web document corpus. Ikeno et al.[3] investigate the possibility to apply it for selecting appropriate translated words for machine translation. We apply this method for query disambiguation in CLIR.

Table 4: Example of two words co-occurrence tendency values of English words.

w_1	w_2	$COT_{MI2}(w_1, w_2)$
database	multimedia	3.37
database	transaction	2.41
database	relational	2.01
database	chair	-2.96
database	soul	-3.43
database	iron	-4.62

4.3.1 Measure of Co-occurrence

Here we define a measure of co-occurrence (we call them *co-occurrence tendency*).

Generally, the window size of co-occurrence is a fixed number of words, but mainly for the limitation of utilizing search engines, we use one document as the window of co-occurrence.

Mutual Information

Mutual Information is one of the metrics that can be used for calculating the significance of word co-occurrence associations[1].

MI can be applied for words in documents and can be used for calculating the correlation between words.

Usually, co-occurrences are measured between two words mainly because of computational and storage cost, but when using a Web search engine, it is not necessary to calculate the frequencies for every word pairs in advance, so we can use co-occurrence frequencies among any *n* words.

n words co-occurrence tendency COT_{MIn} among words $w_1, w_2 \ldots w_n$ is defined, as an extension of COT_{MI2}, as follows:

$$COT_{MIn}(w_1, w_2 \ldots w_n) = \frac{1}{n-1} \log_2 \frac{\frac{f(w_1, w_2 \ldots w_n)}{N}}{\frac{f(w_1)}{N}\frac{f(w_2)}{N} \cdots \frac{f(w_n)}{N}} \quad (2)$$

where N is the total number of documents in the search engine, $f(w)$ is the number of retrieved documents for the word w, and $f(w_1, w_2, \ldots w_n)$ is the number of retrieved documents for the words $w_1, w_2, \ldots w_n$ combined using AND operator. Note that MI is essentially a measure between two events, so this is an *ad hoc* extension only for the purpose of calculating *n* words co-occurrence tendency.

Table 4 gives some examples of co-occurrence tendency values for English words obtained from randomly sampled 14 thousand English Web documents.

4.3.2 Selection of Translations

Using *n* words co-occurrence tendency, the terms actually used for the target language query are determined as follows:

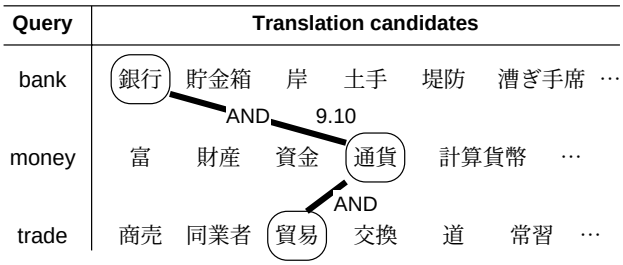

Figure 10: Example of a disambiguation using n words COT (COT_{MIn})

1. Obtain the number of retrieved documents for each term in the query from the Web search engine,
2. Obtain the numbers of retrieved documents for all combinations of translation-candidates whose occurrence frequency for each term exceed the threshold value T_{freq}, from the Web search engine (using AND operator),
3. The term sets whose COT exceed the threshold value T_{COT} are selected as the target language query.

Figure 10 shows an example of a disambiguation using COT_{MIn} for the same query as the previous example.

In this case, the term set “”, “”, and “” is given the highest average COT_{MIn}. In fact, these terms are the most appropriate translations for the source English terms. Actually, all term sets which exceed T_{COT} are combined with OR operator, and used as a Japanese query.

We eliminate the terms which rarely occur, in order to avoid the undesired phenomenon of MI described before. If we do not eliminate rarely occurred terms, unrelated term sets will be mistakenly given the highest average COT_{MIn} score.

This method may cost much time for querying the search engine, especially for queries which have many possible translation-candidate pairs. However, it can be greatly reduced by submitting multiple requests for a query in parallel.

We can consider using the proximity operator instead of AND operator. The proximity operator matches documents containing specified terms within a specific window of words, regarding or regardless of order. For example, AltaVista supports the proximity operator called “NEAR” which retrieves specified terms within 10 words regardless of order. In this case, to be exact, N is the total number of object windows, which cannot be calculated exactly using a search engine. However, since it does not affect the ranking of translation-candidates, and the absolute value is not important here, we used the total number of documents for N in experiments.

4.4 Evaluation

We have conducted experiments to evaluate the effectiveness of Japanese-English CLIR using the proposed query translation method.

4.4.1 Test Data

For the test data, we used NACSIS Test Collection 1 (NTCIR-1)[5] (Research Purpose Use). It contains about 330,000 technical documents, which are summaries of papers presented at conferences hosted by 65 Japanese academic societies, and we used E-Collection which contains about 190,000 English summaries. We used 39 Japanese search topics for evaluation.

We used only TITLE field, which is a very short description of the topic, for queries. It contains 1-7 words (2.7 words on average) and it resembles the queries often submitted by an end-user of Web search engines in terms of the length.

4.4.2 Language Resources used for Experiments

Bilingual Dictionary

For query translation, we merged three dictionaries, *Japanese-English Bilingual Dictionary* and *Technical Terms Dictionary (Information Processing)* of EDR Electronic Dictionary Version 1.5, and EDICT[6] which is a freeware Japanese-English dictionary. In order to avoid the effect of the quality of the dictionary, we added translations for 18 words which appeared in queries.

Monolingual Corpus

We used a Web search engine as the monolingual corpus, as described in section 4.3. We chose AltaVista as the Web search engine for the following reasons:

1. It can specify the target language for the retrieval,
2. It supports the proximity operator (NEAR) in addition to AND operator,
3. It has comparatively larger index as a Web search engine

Furthermore, we also experimented with NTCIR-1 E-Collection, which is identical to the target collection, as the monolingual corpus. In this case, AND operator was used for calculating COT.

4.4.3 Retrieval System

For the retrieval system, we used *Namazu* retrieval system[7] (version 2.0.1). It is a freeware full-text retrieval system based on a Boolean model, and supports basic functions such as a composition of Boolean operators, ranking of the results, and phrase retrieval.

4.4.4 Experiments

Experimental Results

Experimental results for n term co-occurrence tendency based on MI is shown in Table 5 and Figure 11.

[5] http://research.nii.ac.jp/ntcir/data/data-en.html
[6] http://www.csse.monash.edu.au/~jwb/edict.html
[7] http://www.namazu.org/

Table 5: Results of the experiment using NACSIS Test Collection 1.

	Precision						
	NODIS	**NOSTR**	**ONE**	**AND**	**NEAR**	**NTCIR**	**MAN**
0.00	0.4769	0.2836	0.3987	0.4241	0.4841	0.5051	0.5301
0.10	0.3143	0.2239	0.2980	0.2925	0.3415	0.3463	0.3480
0.20	0.2513	0.1704	0.2097	0.2427	0.2755	0.2650	0.2778
0.30	0.2050	0.1397	0.1443	0.1851	0.1964	0.2072	0.2166
0.40	0.1732	0.0905	0.0963	0.1591	0.1677	0.1812	0.1984
0.50	0.1629	0.0814	0.0871	0.1530	0.1613	0.1745	0.1813
0.60	0.0766	0.0582	0.0451	0.0843	0.0888	0.0840	0.0742
0.70	0.0514	0.0421	0.0428	0.0536	0.0565	0.0534	0.0526
0.80	0.0354	0.0217	0.0253	0.0368	0.0388	0.0366	0.0344
0.90	0.0031	0.0100	0.0031	0.0031	0.0033	0.0031	0.0091
1.00	0.0031	0.0056	0.0031	0.0031	0.0033	0.0031	0.0005
Avg. Prec.	0.1438	0.0896	0.1084	0.1371	0.1513	0.1544	0.1564

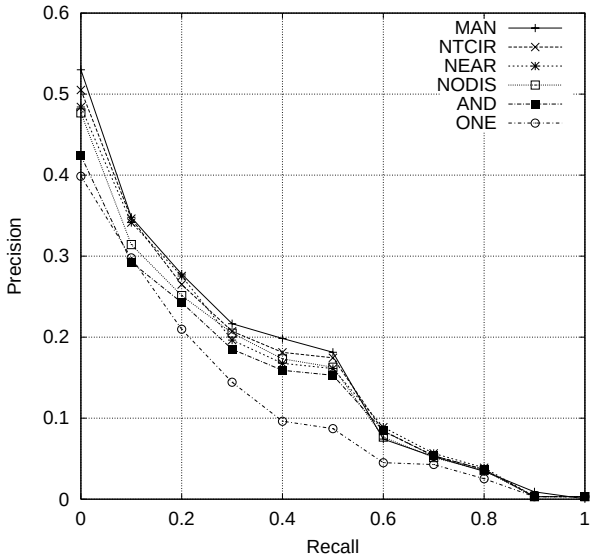

Figure 11: Recall-precision curves of the experiment.

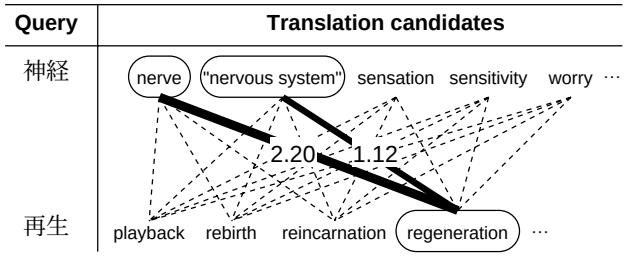

Figure 12: Example of disambiguation for a query used in experiments.

In the table, NODIS is the result of no disambiguation, which means using all possible translation-candidates obtained from the dictionary. In this case, each term in a translation-candidate were combined with AND operator, and all translation-candidate sets were combined with OR operator.

NOSTR is the result of using all possible translation-candidates without structuring (i.e. all translation-candidate terms were simply combined with OR operator).

ONE is the result of using only one translation-candidate which has the top COT_{MI2}, using NTCIR collection as a corpus.

AND and NEAR are the results of using the proposed disambiguation method described in section 4.3 which uses the Web search engine as a corpus. If there was no translation-candidates which co-occur in the search engine, all translation-candidates were used.

NTCIR is the result of using the proposed disambiguation method, but by using NTCIR collection as a corpus instead of a Web search engine.

MAN is the result of English queries manually translated from original Japanese queries.

In the table, 0.00-1.00 indicates the precisions at each recall level, and Avg. Prec. indicates the average precision.

Discussion of the Results

In terms of average precisions compared with NODIS, the proposed disambiguation method improved 1.0 point for NTCIR and 0.8 point for NEAR, but decreased 0.7 point for AND.

The result of NTCIR, in which the corpus is identical to the target collection, achieved 99% of MAN and the effect of disambiguation was significant. By using a corpus which is consistent with the target, appropriate translations were selected in most of the cases. However, for the results of using Web documents as a corpus, NEAR achieved 97% of the result of MAN, but AND did 88% of MAN and it was lower than NODIS. It is probably due to the scope of the co-occurrence. In AND, the scope is one document and there are more chances for errors, but in NEAR, the scope is 10 words and relatively a better co-occurrence information was acquired.

In this experiment, NTCIR achieved the highest performance, but our primary target is CLIR of digital library contents, and it is impractical to prepare a comprehensive corpus that covers all possible domains. Therefore, our

method using Web documents as a monolingual corpus will be effective for CLIR of digital library contents.

4.5 Summary

In this section, we proposed a method for query term disambiguation using a Web search engine, which is readily available. The results of experiments showed that our method is effective for very short queries which are often used by an end-user of Web search engines. We also showed that our method can achieve comparable effectiveness with the manual translation, using a corpus which is consistent with the target collection.

Our method can easily be extended to other language pairs by preparing only a dictionary. Besides, disambiguated queries are simple Boolean queries and can be simply fed into an existing Web search engine.

5 Conclusions

In this paper, we introduced practical solutions to several issues in realizing a digital library system which can handle multilingual documents in a unified manner. The issues we focused on are summarized as follows:

In Section 2, we introduced display and input functions of multilingual text which does not depend on installed fonts and input methods on the client side. The advantages of this technique are, it is easy to use by casual users of the Internet, it is inexpensive because no additional installation is required on the client side, and it is light-weight. This system might be especially useful for documents written in minor languages, for old literature which may include characters that are not defined in any standard character set, and for newspaper articles which may require gaiji's for some proper nouns.

In Section 3, we developed an algorithm for automatic identification of languages and character coding systems of Web documents based on combination of both statistics and heuristics. It is basic but important function in order to handle multilingual documents in practice. We conducted experiments for 12 languages and 10 coding systems, and achieved over 98% correct rate in average. For Asian languages that use multiple bytes to represent one character, we showed that it is possible to identify those languages and coding systems effectively and efficiently, simply using one byte unit and without discriminating the boundaries of characters. For European languages, we showed that it can achieve 98% average correct rate for 9 languages included in ISO-8859-1, by using the consecutive two bytes unit, which takes the connection of consecutive characters into account.

In Section 4, we investigated a cross-language information retrieval technique which is suitable for Web documents in diverse domains. The results of experiments showed that the proposed method is effective for very short queries which are often used by an end-user of Web search engines. We also showed that the proposed method can achieve comparable effectiveness with the manual translation, using a corpus which is consistent with the target collection. The main advantages of this technique are, since it is based on word co-occurrence information obtained from a Web search engine, we do not need to collect a large corpus of diverse domains, and it can easily be extended to other language pairs by preparing only a dictionary.

Acknowledgements

The auther would like to thank Prof. Koichi Tabata, Prof. Shigeo Sugimoto, and Prof. Tetsuo Sakaguchi at University of Library and Information Science for their guidance and sugggestions particularly on the development of MHTML. I would also like to thank Prof. Shunsuke Uemura at Nara Institute of Science and Technology and Prof. Masatoshi Yoshikawa at Nagoya University for their guidance and suggestions with this research, particularly on sections 3 and 4.

References

[1] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, Mar. 1990.

[2] M. Dartois, A. Maeda, T. Sakaguchi, T. Fujita, S. Sugimoto, and K. Tabata. A multilingual electronic text collection of folk tales for casual users using off-the-shelf browsers. *D-Lib Magazine*, Oct. 1997. http://www.dlib.org/.

[3] A. Ikeno, T. Murata, S. Shimohata, and H. Yamamoto. Machine translation using the Internet natural language resources. In *Proceedings of World TELECOM99+Interactive99 Forum*, Geneva, Switzerland, Oct. 1999.

[4] A. Maeda. *Studies on Multilingual Information Processing on the Internet*. PhD thesis, Nara Institute of Science and Technology, Sept. 2000.

[5] A. Maeda, M. Dartois, T. Fujita, T. Sakaguchi, S. Sugimoto, and K. Tabata. Viewing multilingual documents on your local web browser. *Commun. ACM*, 41(4):64–65, Apr. 1998.

[6] A. Maeda, Q. Guan, M. Yoshikawa, and S. Uemura. Automatic identification of coding systems and languages of web documents. *Transactions of IEICE D-II*, J84-D-II(1):150–158, Jan. 2001. (in Japanese).

[7] A. Maeda, F. Sadat, M. Yoshikawa, and S. Uemura. Query term disambiguation for Web cross-language information retrieval using a search engine. In *Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages (IRAL2000)*, pages 25–32, Hong Kong, China, Sept. 2000.