Identifying the Coding System and Language of On-line Documents on the Internet

Gen-itiro KIKUI
NTT Information and Communication Systems Laboratories
1-2356 Take, Yokosuka-shi
Kanagawa 238-03, JAPAN
kikui@nttnly.isl.ntt.jp

Abstract

This paper proposes a new algorithm that simultaneously identifies the coding system and language of a code string fetched from the Internet, especially World-Wide Web. The algorithm uses statistic language models to select the correctly decoded string as well as to determine the language. The proposed algorithm covers 9 languages and 11 coding systems used in Eastern Asia and Western Europe. Experimental results show that the level of accuracy of our algorithm is over 95% for 640 on-line documents.

1 Introduction

Recent advances in information infrastructure have made an enormous number of on-line documents accessible. A notable example is the explosive growth of World-Wide Web(WWW) involving more than 10 million documents.

Accessing and using such a huge number of online documents require intelligent document processing such as content-based search, categorization, information extraction, and machine translation. Most of these processes assume that documents are correctly decoded and the language is known. For documents on the WWW, however, these assumptions do not hold. This is because each language community uses its own coding system which is optimal for internal communication but is not appropriate for exchanging information with those outside at community.

A fundamental solution is to develop international standards for an internationalized coding system and language representation. In fact, there is active discussion on the international coding standards(Yergeau et al., 1995)(Nicol, 1995)(Unicode, 1994), and the language representation on the WWW(Unicode, 1994). However, it will require several years before most of the documents are encoded into a unique well-defined coding system.

A realistic approach, which also goes together with the above 'fundamental' approach, is to develop a more intelligent module that can estimate which coding system and language used for each on-line document on the current WWW.

Automatic identification of the coding system is achieved in communities where a limited number of coding systems are used. For example, (Lunde, 1993) presents an algorithm for selecting one of three coding systems for Japanese texts (UJIS, SJIS, and JIS) The algorithm, however, is not applicable to international domain where a lot of other coding systems are potential candidates.

Automatic language identification has been discussed in the field of document processing. Several statistic models have been tried including using the n-gram of characters (Cavnar, 1994), diacritics and special characters (Beesley, 1988), and using the word unigram with heuristics (Henrich, 1989). Among these methods, the result by (Cavnar, 1994) shows the best accuracy over 95%. (Giguet, 1995) achieved over 99% accuracy by using a rule-based (i.e., non-statistic) method.

These methods, however, cannot handle East-Asian languages, because they presuppose that input texts are easily segmented into words, which does not hold true in these languages. Another problem is that it presupposes that the input document is correctly decoded.

Sibun and Penelope (Penelope and Sibun, 1994) proposed a method of determining the language of a text image. The problem tackled by them is similar to ours, in the sense that the input is not a unique character string but a string that potentially corresponds to several different character strings. Their method, however, can not directly applied to our problem.

This paper presents an algorithm that identifies the coding system and the language of a given text. The algorithm is an application of an automatic language identification using statistic language models. It covers 11 coding systems and 13 languages used in East-Asian countries as well as Western-European countries.

The next section describes the problem. Section 3 introduces our algorithm. Section 4 and Section

5 describes an example and the experimental results respectively.

2 Problems in Decoding and Identifying Languages on the WWW

2.1 Brief Explanation of Character Coding Systems

In a communications network, characters are represented as numbers, or a sequence of numbers. A *character coding system* specifies the mapping between characters and numbers. A coding system consists of a *character set* and an *encoding scheme*.

A *character set* is a set of characters collected to represent texts in a certain language community. For example, JIS-X0208-1983 (referred to as "JIS character set" in this paper) is a character set for encoding texts (mainly) in Japanese. Each character in a character set has a unique identification number. It should noticed that the same number may appear in different character sets.

An *encoding schema* maps each element of a character set into a (sequence of) number(s) that is used in communication networks. The simplest encoding scheme directly uses the identification number of a character set for communication.

Some encoding schemes are designed to encode texts that contain characters from two or more character sets. For example,the encoding scheme for JIS coding system (ISO-2022-jp) uses *escape sequences* to indicate changes of character set in the code string in the following way:

1. "ESC $ B" shows the beginning of JIS character set.
2. "ESC (B" shows at the beginning of ASCII character set.

2.2 Ambiguity in Determining Character Coding System

For historical reasons, documents on the current WWW are encoded in various coding systems. For example, servers in Western-Europe normally use ISO-8859-1 (ISO-LATIN-1), whereas most UNIX servers in Japan encode text using Japanese EUC (Extended Unix Code). The problem is that different coding systems are applicable to the same code-string.

The fundamental solution is to have everyone use a unique coding system that can handle all the characters in the world. ISO-2022 is one such coding systems. This system assigns a unique identifier to every registered character set in the world and specifies escape sequences for switching one character set to another. Although most of the local coding systems in the world are 'compatible' with ISO-2022, many of them lack escape sequences which are not necessary for choosing the correct character set in the local domain but are necessary in the international domain.

Therefore, the sender should give or the receiver should infer with what coding system the received coded sequence is encoded.

One approach is to transfer the name of the coding system with the upper level protocols. For example, the Internet mail protocol can transfer the coding system with which the message is encoded. However, on the WWW, active discussion still continues on the WWW as to how to deliver the coding system.

Another approach is to uncover the coding system from the received byte code string. If the potential candidates for the code system are limited, the correct coding system may be inferred by using simple pattern matching. For example, the byte code contains the pattern "ESC $ B" then it must be encoded with ISO-2022 and include Japanese characters. However, in international domain, it is difficult or impossible to specify the coding system with simple pattern matching. For example, Japanese EUC and Korean EUC cannot be discriminated by this simple method, because most of their code values overlap[1].

In summary, a more sophisticated method is required to identify the coding system from the content of the code string.

2.3 Ambiguity in Determining Language

Most text processing systems have language-dependent components such as rules or dictionaries. Thus, it is crucial to know in what language the target document is written in order to choose the appropriate system or language specific rules and dictionaries.

If we restrict ourselves to HTML documents, then explicit *language tagging*, which represents the language of the text body, will be introduced in a future version of HTML specifications. This, however, does not handle multiple languages in one document. Moreover, there are a lot of non-HTML documents on the WWW.

If the character set used in the text is known, it might be good clue for identifying the language because some character sets strongly suggest which language(s) was used. For example, if a document consists of characters in the JIS character set, the document must be written in Japanese. However, this is not always the case. One reason for this is that the character set of a text is sometimes ambiguous due to the decoding problem described above. Another reason is that some character sets are designed to cover multiple languages (e.g., ISO-8859-1 for several Western-European languages). Sometimes a character set is used in a language that is not the primary candidate suggested by the character set. For example, a document containing only US-ASCII characters,

[1] In detail see (Lunde, 1993).

which suggests the document is in English, may be a German document in ASCII-format (e.g., ö is written as oe).

For this reason, it is necessary to identify the language from the content of the given character string.

3 Our Algorithm

Our basic idea is to use statistic language models to select the correctly decoded string as well as to determine the language. The idea comes from the observation that a human can distinguish whether or not a text is written in the language s/he can read. If the text is judged to be written not in the language the person can read, then it is written in another language or decoded with an incorrect coding system.

In our algorithm, the human intuition on his familiar language is realized by a statistic-based module which calculates how likely a text to be from a specific language.

3.1 The Scope of our Algorithm

The current version of our algorithm can handle the following 11 coding systems and 9 languages.

- coding systems

  7bit Coding ISO 646 USA (ASCII), JIS Code(ISO-2022-jp), KS C 5601-1992, GB 2312-80, ISO-2022-int

  8bit Coding ISO-8859-1, EUC-GB(Simplified Chinese), EUC-KS, EUC-JIS(UJIS), BIG5(Traditional Chinese), Shift JIS(MS Kanji code)

  Entity Reference with ASCII *entity references*(e.g., ö is represented as "Ö") for the ISO-8859 characters defined in HTML (or SGML) specifications.

- languages

  European English, French, German, Spanish, Italian, Portuguese

  East Asian Chinese, Korean, Japanese

3.2 Outline of the Algorithm

Our algorithm consists of the following two major steps.

Step 1 This step divides the given code string into East-Asian part (i.e., sub-strings consisting of East-Asian characters) and the rest (i.e., European) part.

Step 2 This step decodes each part and identifies its language(s).

The following two subsections describe the above two steps in detail.

```
procedure extract_ea_string(code-string)
## Loop 1
For each csys in ea_coding_systems
do
   if (ea_string(code-string, csys)) then
     push(get_ea_string( code-string, csys),
   ea_string_list)
   endif
end
## Loop 2
For each csys sort_by_length(ea_string_list )
do
   (score, lang) ←
   identify_language(ea_string)
   if (score > threshold) then
     return (lang, ea_string)
     endif
end
return nil
```

Figure 1: Extraction of Eastern-Asian characters.

3.3 Dividing Code-String

If the give code string contains escape code sequences defined in ISO-2022 or its variants, East-Asian character strings are easily extracted because East-Asian characters are explicitly marked by escape sequences in the string.

If the given code string does not contain such escape sequences, the Eastern-Asian part is identified by the procedure shown in Figure 1. The procedure consists of the following two loops.

- Loop 1

  If a coding system is determined, it is easy to extract Eastern-Asian characters. For every coding system that can handle Eastern-Asian characters, the first loop tries to extract Eastern-Asian characters by using the coding system.

  The function **ea_string** takes a code-string and (the name of) a coding system. It extracts Eastern-Asian character-strings, presupposing that the given code string has been encoded with the given coding system (*csys*). This function can be realized by simple pattern matching. For example, if we presuppose that the given code-string is encoded with EUC-JIS, then the adjacent two bytes that match [A1H-FEH]{2}, a two byte sequence whose values ranges from A1 to FE (in hexadecimal representation), correspond to a Japanese (or JIS) character.

  Table 1 shows examples of regular expression patterns in our system [2].

[2] More information on Eastern-Asian code values are available in (Lunde, 1993)

Table 1: Patterns for Extracting Eastern-Asian Characters (sample)

Char-set (Coding)	Pattern
GB (EUC-GB)	[A1-FE]{2}
KS (EUC-KS)	[A1-FE]{2}
JIS (EUC-JIS)	[A1-FE]{2}, 8F[A1-FE]{2}
BIG5 (Big5)	[A1-FE][40-7E,A1-FE]

If a non-empty string is returned by *ea_string*, it is decoded with the presupposed coding system and registered in *ea_string_list*.

- Loop 2

  The second loop tries to identify the language of each East-Asian character-string in *ea_string_list*.

  Each Eastern-Asian string is passed to the language identification routine in the descendent order of its length. The language identification routine, described in Section 3.4, takes a character string and returns the most likely language and the score of likelihood. If the score is larger than a predetermined threshold, the loop terminates and returns the language and the score.

  If the score of every Eastern-Asian string does not exceed the threshold, then the loop returns nil, which indicates that no Eastern-Asian characters are involved in the code string.

After the Eastern-Asian part is identified, the remainder is classified into the European part.

3.4 Identifying the Language

The language of a text is identified by the following three steps.

1. Selecting possible languages for the given coding system

   The coding system (or the character set(s)) of a text is loosely related to the language of the text. For example, a document encoded with US-ASCII is not written in Korean. We made heuristic rules to map a coding system to possible languages.

2. Calculating the 'likelihood' of the decoded string for each language.

   For each language, this step calculates how likely the decoded string is to be from that language by comparing the string with the statistic model of the language.

3. Selecting the language with the highest likelihood

   This step compares likelihood scores, then returns the language with the highest likelihood score.

The second step is the most important. Our system uses a unigram model for both Western-European languages and East-Asian languages, but the models for Western-European languages and the models for the East-Asian languages have different unigram units.

3.4.1 Likelihood Score for Western-European languages

In order to distinguish Western-European languages, we applied a method proposed by Cavnar (Cavnar, 1994). We assign a class name for each word. The class name of a word longer than n characters is the concatenation of "X-" and the last n characters of the word. If the word is not longer than n characters, the class name is the word itself. For example, if $n = 4$, then class names of "beautiful" and "the" are *X-iful* and *the* respectively.

Let $TEXT$ be the set of words in a text, then the likelihood of $TEXT$ with regard to language l is given as the following $P(TEXT, l)$

$$P(TEXT, l) = \prod_{w \in TEXT} P(C_w, l)$$

where $P(C_w, l)$ is the unigram probability of C_w in language l , and C_w is the class name of the word w.

$P(C_w, l)$ is estimated from text corpora in language l.

3.4.2 Likelihood Score for East-Asian languages

As compared with Western-European languages, East-Asian languages have the following properties:

1. A large number of characters

   East-Asian languages use over 3,000 ideographic or combined characters. A character is normally encoded with two (or more) bytes.

2. No Explicit Word Boundaries

   In East-Asian languages, there are no explicit word delimiters (corresponding to spaces in English) in a sentence. We cannot use word-based language models.

For East-Asian languages, we use a character unigram, instead of a word unigram, to model a language. Formally,

$$P(TEXT, l) = \prod_{char \in TEXT} P(char, l)$$

where $P(char, l)$ is the unigram probability of *char* in language l .

4 Example

Suppose the following code sequence is given to the algorithm.

Figure 2: Decoded strings

```
b8c0 b8ec bcb1 cacc a4ce cafd cba1 0a49
6465 6e74 6966 7969 6e67 2074 6865 204c
616e 6775 6167 650a
```

The string is first divided int Asian and European parts. Since there is no escape sequence, which begins with "1b", the procedure in Section 3.3 is applied.

The division procedure first tries to extract Eastern-Asian characters from the given string. The first 14 bytes, from b8 to a1, match with every pattern in Table 1. This means that all the four coding systems are potential candidates and that they extract the same Asian character part.

This part is decoded with each coding system. Result strings are shown in Figure 2.

Next, the statistic-based language identification is applied to each decoded string,

Table 2 shows the score (=probability) of each character as regards to the language that produced the highest likelihood (i.e., average score). For example, the second column shows the score of each character as regards to Chinese (zho) when the input code string is decoded with EUC-GB. This implies that Chinese(zho) is the most likely language if we presuppose the original string is encoded with EUC-GB.

Table 2: Likelihood scores of characters of Asian part

char.	SCORES(log prob.)[lang]			
(JIS)	GB[zho]	KS[kor]	JIS[jpn]	BIG5[zho]
1(言)	-11.05	-15.51	-7.25	-12.62
2(語)	-11.15	-15.51	-7.86	-10.86
3(識)	-8.47	-7.69	-8.68	-12.42
4(別)	-11.45	-15.51	-7.71	-15.51
5(の)	-15.51	-15.51	-3.40	-7.50
...	..	..	..	..
Ave.	-11.06	-14.39	-6.9	-12.64

zho=Chinese, kor=Korean, jpn=Japanese

The bottom row shows that the highest average score is obtained when the input is decoded with EUC-JIS and the language is Japanese(jpn). Since this score exceeds the threshold (-10), the Eastern-Asian part is confirmed to be Japanese string encoded with EUC-JIS.

The remaining part is decoded into "Identifying the Language" and sent to European language identifier. Table3 gives scores of tokens as regards to three languages.

Table 3: Likelihood scores of characters of European part

class of	SCORES(log prob.)			
token	eng	deu	ita	..
X-ying	-7.45	-10.80	-10.80	..
the	-3.11	-9.40	-7.21	..
X-uage	-7.20	-10.80	-10.80	..
Ave.	-5.9	-10.3	-9.60	..

eng=English, deu=German, ita=Italian

In this table, English(eng) is the most plausible language for European part with sufficient score.

The final result is easily obtained by combining results of the Asian and the European parts.

5 Evaluation and Discussion

5.1 Data

For evaluation purposes, we collected a set of documents on the WWW. Each document was manually assigned a correct language name. If a document contained more than one Western-European languages, a language that covered more than 80% of the document was chosen as the correct language. Documents without the correct language (i.e., documents without the unique main language) were discarded. The same process was applied to East-Asian parts of documents.

The remaining documents were divided into 700 training documents and 640 test documents.

5.2 Identifying Western-European languages

Table 4 shows the confusion matrix for the Western-European language results. The columns correspond to the outputs from the system, and the rows correspond to the correct answers. The value of n is set to 4, which gives the fewest number of errors for the training set.

The error rate is 4.8%. Error occurs when the document is not a normal text (e.g., computer programs, a list of proper names or network addresses, etc.).

The result shows that our method achieved the level of correctness equivalent to the previous methods that presuppose correctly decoded character strings (Cavnar, 1994).

5.3 Identifying East-Asian languages

Table 5 shows the confusion matrix for the East-Asian language results. The matrix shows that the system performs fairly well also for East-Asian languages.

The error rate is 4.6%. Most errors occur when the document includes only a few East-Asian characters. There are many documents which are written mostly in English but only proper names, es-

Table 4: Confusion matrix for Western-European languages

	deu	eng	esl	fra	ita	por	ELSE
deu	51						2
eng		178		1			6
esl		1	23				
fra				90			
ita					3		1
por						6	
ELSE	1	8	1				70

deu=German, eng=English, esl=Spanish, fra=French, ita=Italian, por= Portuguese

pecially people names and organization names, are written in Asian characters.

Table 5: Confusion matrix for East-Asian languages

	jpn	kor	zho	ELSE
jpn	123	0	0	4
kor	1	7	0	0
zho	2	0	47	0
ELSE	1	0	2	8

5.4 Dividing the Western-European part and the East-Asian part

As far as the above experiments are concerned, there is no confusion between Western-European part and the East-Asian part.

6 Conclusion

This paper proposed an algorithm for simultaneously identifying the coding system and the language of a given code-string. It handles three East-Asian languages as well as six Western-European languages with a high level of accuracy. The algorithm uses statistic language models to select the correctly decoded string as well as to determine the language. Since the algorithm uses statistic language models, it is robust and easily extendible to other languages.

The algorithm is implemented in a cross-lingual search engine for WWW pages which has a language index (i.e., WWW pages are indexed by their languages).

We intend to elaborate the algorithm so that it can identify languages in multi-lingual text, because many documents on the WWW are multilingual.

Acknowledgment

The author express his gratitude to Yoshihiko Hayashi and Seiji Suzaki for the comments on an early version of the paper. The comments by anonymous reviewers of Coling 96 helped to improve the paper.

References

Beesley, Kenneth R., "Language Identifier: a Computer Program for Automatic Natural Language Identification of On-line Text", Language at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association, October 1988, pp 47-54.

Cavnar, William B. and Trenkle, John M., "N-gram Based Text Categorization, Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval", April 1994, pp 161 -169.

Giguet, Emmanuel, "Multilingual Sentence Categorization according to Language", Proceedings of the EACL95 SIGDAT Workshop, October 1995.

Henrich, Peter, "Language Identification for the Automatic Grapheme-to-Phoneme Conversion of Foreign Words in a German Text-to-Speech System", Proceedings of Eurospeech 1989, September 1989, pp 220-223.

Lunde, Ken, "Understanding Japanese Information Processing", O'Reilly and Associates, Inc, 1993, Japanese translation "Nihongo-Joho-Shori" , April 1995

Nicol, Gavin T., "The Mutiliungual World Wide Web"
http://fuzine.mt.cs.cmu.edu/mlm/lycos-home.html

Sibun, Penelope, and Spitz, A.Lawrence, "Language Determination: Natural Language Processing from Scanned Document Images" Proceedings of ANLP 94, October 1994, pp 15-21.

Unicode Inc. "The Unicode Standard"
http://www.stonehand.com/unicode/standard.html

Yergeau, F., Nicol, G., Adams, G., and Duerst, M., "Internationalization of the Hypertext Markup Language", Internet Draft, draft-ietf-html-i18n-02.txt, November, 1995,.