Language Identification on the WWW

by

Lehe Cai

Thesis
submitted in partial fulfillment of the requirements for
the Degree of Master of Science (Computer Science)

Acadia University
May Convocation 2009

This thesis by Lehe Cai was defended successfully in an oral examination on Dec 18th, 2008.

The examining committee for the thesis was:

_____

Dr. Diane Holmberg, Chair

_____

Dr. Vlado Keselj, External Reader

_____

Dr. Andre Trudel, Internal Reader

_____

Dr. Darcy Benoit, Supervisor

_____

Dr. Danny Silver, Head/Director

This thesis is accepted in its present form by the Division of Research and Graduate Studies as satisfying the thesis requirements for the degree Master of Science (Computer Science).

I, Lehe Cai, grant permission to the Head Librarian at Acadia University to provide copies of thesis, on request, on a non-profit basis.

_____

Author

_____

Supervisor

_____

Date

Table of Contents

List of Tables

List of Listings

List of Figures

Abstract

This thesis discusses the problem of language identification on Web pages. Previous research has mostly focused on text language identification and a limited amount of research has concentrated on Web language identification. Web language identification is not a simple task that comes from text language identification. The noisy and diverse nature of Web pages introduces additional difficulties with Web language identification.

To solve this problem, we introduce a robust approach which consists of multiple methods such as detection of language and character encoding declaration and Web text language identification to better handle the difficulties of Web language identification. The methods are harmonized to maximize their strengths and complement each other. By verifying our approach on corpora of 1400 Web pages for seven languages, it achieved 99.6 percent accuracy rate.

Furthermore, we designed a system that employed our composite Web language identification approach and several other Web language and geographical distribution approaches for determining Web server location and language distributing status of portal Web pages on the Internet. The data collection came from the latest Web census (Benoit, Slauenwhite & Trudel, 2006 and 2007). This system allows us to provide information such as countries ranked by Web server number, the languages ranked by identified portal pages count and language distribution status in a country such as Canada.

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Dr. Darcy Benoit, for his support, patient guidance, creative suggestions and encouragement throughout the thesis research. Without his encouragement, this thesis could not have been completed. Working with him on this research has been the best experience I have had while studying at Acadia University. I also extend my thanks to my family and friends who have helped and supported me to complete this work.

1. Introduction

Language Identification on the WWW has become increasingly important since the amount of available information on the Internet has been rapidly increasing during the last decade. When processing a collection of Web pages, appropriate language annotations can be used to boot strap various Web applications, such as Web classification and automatic Web page translation.

The text language identification method is not a single approach for Web language identification according to the special characteristics of Web pages. Instead, multiple approaches should be employed to handle the noisy and diverse nature of Web pages; these approaches could complement one another in increasing the accuracy and coverage of Web language identification on the Internet.

1.1. Web Census Project

Our language identification was based on the collection of Web pages obtained by the Web census project (Benoit, Slauenwhite & Trudel, 2006 and 2007). This project was used to census the size of the World Wide Web by checking every possible IPv4 address on Port 80 (the default Web server port) for the presence of a Web server. Once a Web server was found, the home page of this Web server was downloaded to a local database for further research.

The IPv4 address contains four portions. Each portion is referred to as an octet and ranges between 0 and 255:

{0-255}.{0-255}.{0-255}.{0-255}

In total, there are approximately 4.3 billion (256^4) unique IP addresses. Among these 4.3 billion addresses, some address ranges are known to be invalid or not in use. Only 3.7 billion IP addresses can be associated with a computer. The Web census project checked each IP address among the 3.7 billion IP addresses to determine how many Web servers were hosting on these IP addresses on Port 80.

Starting in the fall of 2006, the world's first Web census (Benoit, Slauenwhite, & Trudel, 2006 and 2007) began by our Web census research group. The Web census data we used was from the data collection of the latest census that started in September 2007 and terminated in March 2008.

1.2. Web Language Identification

In the last two decades, various research efforts such as Grefenstette (1995), Dunning (1994), Cavnar (1994) and Kikui (1996) has focused on text language identification. Each approach has its merits and problems. The N-gram based approach, described in Cavnar (1994) is the most widely used text language identification approach because it is an agile, reliable, accurate and no-linguistic-knowledge-required approach. The experimental results also show that this approach performed well on short texts.

Compared with text language identification, only a small amount of research (such as Bruno and Mario, 2005) worked on Web language identification or some commercial research (such as Shanjian and Katsuhiko, 2001) was used to enhance character encoding auto-detection on Web browsers. Bruno and Mario (2005) provided a comprehensive approach for Web language identification but they manually generated some HTML documents as their test source collection, which means their experimental results are questionable.

Web language identification is not a simple task come from text language identification. Difficulties introduced in the Web domain are related to the noisy and diverse nature of Web pages; therefore, diverse knowledge and approaches should be involved to deal with Web language identification problems according to the special characteristics of Web pages.

This thesis presents a composite approach for Web language identification. It was composed of three methods: language declaration language identification, character encoding declaration language identification, and text language identification. They were used to maximize one another's strengths and complement the other detection methods.

We should clarify that three language identification methods are not original but it is an innovation by using them together for Web language identification. In addition, we improved each method and refined them together on a large realistic setting—10,393,461 home pages that were retrieved from the Internet with the HTTP response code of "200", which means the HTTP request has succeeded. More specifically, language declaration

and N-gram text language identification methods were used by Bruno and Mario (2005) for Web language identification. Furthermore, character encoding declaration language identification method originally was not used for language identification but encoding auto-detection on major Web browsers in order to display Web content correctly.

Web information extraction was responsible for extracting useful information from noisy Web pages to allow Web language identification. Without the help of this information extraction, there was impossible to perform Web language identification. Web language declaration language identification determined the language from the detected uniform language declaration on a Web page (for example: English can be specified as en, en-US, eng, English, etc). Web character encoding declaration language identification detected the character encoding that was able to uniquely identify a single language. An improved N-gram based text language identification method was used for identifying the textual text that was extracted from Web pages.

1.3. Objective

The main objective of this thesis is to develop a Web language identification system that is able to identify the languages from the portal Web pages on 24.2 million Web servers found by our latest Web census project. A portal Web page is the default Web page that responses to the http request when a web server is accessed by its IP address on port "80" (the default Web server port). Although a Web server could contain multiple virtual Web servers, only one web page is returned per IP address.

In addition to the Web language identification, we designed several approaches for determining Web language and geographical distribution. They include:

1. Language distribution on the portal Web pages: It showed us the language distributing status of portal Web pages on the Internet.
2. Geographical distribution of Web servers: It determined where the servers physically resided on the Internet.
3. Language distribution on the specific geographical location: The number of portal Web pages for each language was counted for the Web servers that resided in each country. It told us the Web language distributing status in different countries.

Based on all the distribution results, we also carried out multiple investigations for examining if the distribution results reflect the status of the international community and the language distribution in a specific country.

1.4. Scopes and Requirements

The scopes of this thesis research were based on collection of 24.2 Web servers and their default portal Web pages obtained by the Web census project that started in September 2007 and terminated in March 2008. We estimated geographical distribution on these 24.2 million Web servers. In addition, we identified languages of 10.4 million portal Web pages which is a complete subset of the 24.2 million default portal Web pages with “200” HTTP response codes.

The requirements for our research were included three parts:

1. Providing an accurate, efficient and extensible Web language identification approach to classify languages without linguistic knowledge.
2. Presenting Web server's geographical distribution and Web pages language distribution results
3. Analyzing all the distribution results

1.5. Organization of the Thesis

Some general background information and related applications are introduced in Chapter 2. In Chapter 3, we describe the design of Web Language Identification and Distribution System (WLIDS) from the architecture and detailed functional design perspective. Chapter 4 provides detailed information about the problems encountered during the implementation and our solutions. Chapter 5 presents our experimental results and our investigation based on the results. Finally, Chapter 6 highlights the key issues covered in this thesis with a summary of the main contributions and future work.

2. Background

The main objective of this chapter is to provide fundamental knowledge of various existing Web language identification solutions. The discussion of the evaluation and comparison results for these solutions are presented. The background includes four aspects: text language identification, Web information extraction, Web language identification and Web language identification and distributions.

2.1. Text Language Identification

Text language identification is used to determine in what natural languages a given text is written. Correct determination of languages is an important preprocessing step for a variety of natural language processing tasks. Traditional text language identification relies on manually identifying frequent words and letters which are known characteristics of particular languages. More recently, computational approaches such as Grefenstette (1995), Dunning (1994), Cavnar (1994) and Kikui (1996) have been applied to the problem and notably improved the text language identification efficiency and accuracy. Four automated text language identification approaches that are most widely used in practice are evaluated. They are common words and unique letter combinations approach, statistical approach, N-gram approach and character sets and statistical combinations approach.

2.1.1. Common Words and Unique Letter Combinations

The common words approach utilizes a library to contain the most frequent words of each language. Each word in the library is associated with a value based on its probability of occurrence. The language identification system receives a sequence of words from a document to be classified and compares each received word with all the words in the library. Whenever a received word is found in the library, the corresponding value is accumulated. This procedure is continued until the whole document is processed. Thus, the language discriminating values are compiled to indicate the language of the document.

Grefenstette (1995) used the ECI CD-ROM (Corpora from European Corpora Initiative) as corpora to perform language identification. The ECI CD-ROM is a large collection of multilingual newspaper articles from 27 European languages and is made available in digital form for scientific research. This common word test is based on ten languages and composed of two steps:

1. Constructing a common words frequency library
2. Estimating the language by using the common words method

In order to construct the common words library, the first million characters of text from the corpora in different languages are tokenized and all tokens less than six characters are extracted. For each language, the tokens are counted and the words that

appear more than three times are kept in the common words frequency library. Each word is associated with a value according to its probability.

For estimating the language, the second million characters for each tested language are extracted. The test corpora are broken down into sentences and each sentence is tokenized. Tokens appearing in the common words frequency library are assigned their probability values and tokens not in the library are assigned a minimum probability value.

Table 2-1 shows the average language identification accuracy ratios using the common words method for sentences with different lengths. The breakdown results demonstrate that this language identification method performs well for sentences with more than 11 words. The accuracy ratios for language identification increase when the sentences become longer.

Language	**Number of Words in Sentence**							
	1 or 2	3-5	6-10	11-15	16-20	21-30	31-50	More than 50
Danish	40.5	61.6	91.8	94.8	95.5	94.3	92.5	100.0
Dutch	47.1	84.2	98.5	99.2	99.5	99.6	99.9	100.0
English	52.6	87.7	97.3	99.8	99.9	100.0	99.9	100.0
French	30.8	81.8	96.0	97.2	99.8	100.0	100.0	100.0
German	23.1	71.6	89.6	98.2	99.8	100.0	100.0	100.0
Italian	16.7	65.0	96.9	99.8	100.0	100.0	99.9	100.0
Norwegian	87.5	97.4	99.2	99.8	99.9	100.0	100.0	100.0
Portuguese	51.1	88.9	98.2	99.7	99.9	99.9	100.0	100.0
Spanish	8.1	81.5	98.8	99.7	100.0	100.0	100.0	100.0

Table 2-1 Results for common words method (Grefenstette, 1995)

The “unique letter” is another common words approach. According to Tomas (2005), it is based on the idea that many languages have unique letters or rare combinations of characters. The way to count the score of unique letters is the same as the common words method; however, the unique letters are not only contained in the common words library but in the whole vocabulary of the correlative language.

Obvious advantages of using the common words and unique letter approaches are that they perform very quickly and are easy to implement. Despite these advantages, several shortcomings must be noted, and include:

1. Poor language identification accuracy for short texts.
2. Impossible to apply to languages that are incapable of tokenization.
3. Unique letters are not always as unique as they may seem.

For short lengths of text, the probability of the common words and unique letters appearing is quite low and results in poor language identification on these treated texts. In addition, the common word approach depends on the ability to define and recognize common words. If tokenization into words is difficult, such as with some Asian languages, this approach will not perform in the desired way. Moreover, since many words can also appear in different languages, the reliability of the unique letter approach is beneath our expectation.

2.1.2. Statistical Approach

The technical report of Dunning (1994) is one of the more famous articles on statistical language identification. The technique described in the article involves developing a set of character level language models from the training data. Then, based on these language models, Markov Models and Bayesian Decision Rules are used to calculate the probability that a particular test string might have been generated by each model. In short, the statistical language identification includes two steps: building language models and then language identification based on the built language models. In the following paragraphs, a detailed explanation and evaluation results will be provided.

Preparing a set of character level language models from training data is done by dividing the strings into segments and entering them into a transition matrix which contains the probabilities of the occurrence of all character sequences. While building the language models, the model parameters and the text segment size significantly impact the veracity of the language models. The model parameters are related to the best K order Markov model and a lower order Markov model in the second step. In Dunning (1994), a detailed discussion of the estimation of the model parameters is described. The basic idea is to use the ratios of counts come from the training data for estimating the values of the model parameters.

In the second step, Markov models and Bayesian Decision Rules are employed in Dunning (1994) to calculate the probability of input strings based on the language models

in order to deduce languages. A Markov model is a random process in which a probability is assigned for each character based only on the preceding characters of the sequence. More formally, in the Markov model, the probability of the strings from an alphabet *X* is defined by a sequence of random variables. The probability of a particular string *S* is

$$p(S) = p(s_1 \ldots s_n) = p(s_1) \prod_{i=2}^{n} p(s_i \mid s_{i-1}) \quad (1)$$

Where S_i indicates the *ith* character and S_n indicates the *nth* character in the string S.

The first order Markov model is the most commonly used Markov model. It is described by the initial state distribution $p(s_1)$ and the transition probabilities $p(s_i|s_{i-1})$. When the distribution of the next state depends on the last *k* state called *k*th-order Markov model, the equation of the *k*th-order Markov model can be simply relabeled from the first order Markov model. The transition probabilities for this relabeled model are:

$$p(s_{i+1} \ldots s_{i+k} \mid s_i \ldots s_{i+k-1}) = p(s_{i+k} \mid s_i \ldots s_{i+k-1}) \quad (2)$$

In other words, knowledge of the last k stages is equivalent to the knowledge of the entire past history.

If we assume that the dependence of characters in natural languages satisfies the *k*th-order Markov model, we can produce a fundamental model of natural language. Furthermore, a *k*-order of Markov model is relatively small because the size of the training data would grow exponentially with the increasing of the *k* order. When the training data is insufficient, a higher order model will perform poorly. A higher order

model captures the structure of a language much better but is not necessarily better at language identification.

Dunning (1994) employed Bayesian Decision Rules to compare new treated strings with past experience and pick the most likely probability of observing strings in order to minimize the error of language identification.

The practical results in Figure 2-1 indicated that for language identification, the more training data available, the better the results that can be achieved. The accuracy of language identification was also impacted by the size of observed strings. For 20 bytes of observed text with 50K of training, it achieved 92 percent accuracy. When 500 bytes were observed, the accuracy improved to about 99.9 percent.

In addition, the lower order models achieved better language identification results. For example, in Figure 2-1, the shorter test strings, models of order 1 or 2 achieved the best performance. For the longer test strings, except the model of order 0, other models' performances are similar. Overall, language identification is very accurate when the language model is in the lower orders (1 or 2).

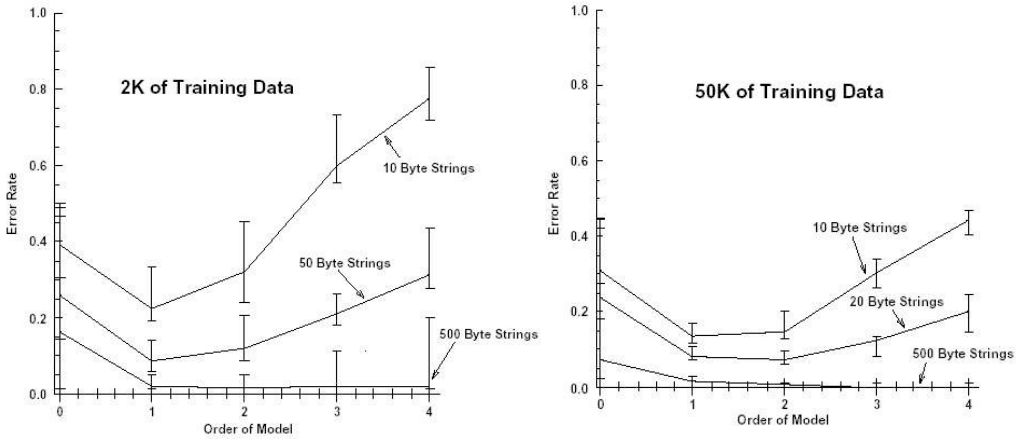


Figure 2-1 Statistical testing result in (Dunning, 1994)

The statistical approach requires more computational power than the common word approach. However, it does not require linguistic knowledge in advance and can be used for identifying the languages that are unable to be tokenized. For example, unlike Western languages, Chinese is written without spaces between words and each word could contain variable bytes, two to four bytes. Thus, it is hard to run any word- or token-based linguistic processing on Chinese.

2.1.3. N-Gram Approach

Character N-grams can be seen as substrings of words that are generated from larger text. The input strings are separated to an amount of substrings with a maximum size N. For each substring, the frequency of occurrence is counted. Following this manner, the category profiles that represent the various categories are computed on training data. The least frequent N-grams are discarded and the rest are written to a category profile. A profile of an examining string is computed, too. Finally, the system computes a distance

measure between the observed strings profile and each of the category profile. The category profile that has the smallest distance to the profile of the observed strings indicates the language.

Cavnar (1994) used an experimental text categorization system to identify the language of the input documents. Figure 2-2 illustrates the overall data flow of the system.

According to the definition in Cavnar (1994), an N-gram is an N-character sub-string of a longer string. A string is simultaneously divided into a set of overlapping N-grams with several different lengths. In order to match the beginning-of-word and ending-of-word, the blank characters are appended to the initial and final positions of the string. For example, the word “CENSUS” would be divided into the following N-grams:

bi-grams: _C, CE, EN, NS, SU, US, S_

tri-grams: _CE, CEN, ENS, NSU, SUS, US_, S_ _

quad-grams: _CEN, CENS, ENSU, NSUS, SUS_, US_ _, S_ _ _,

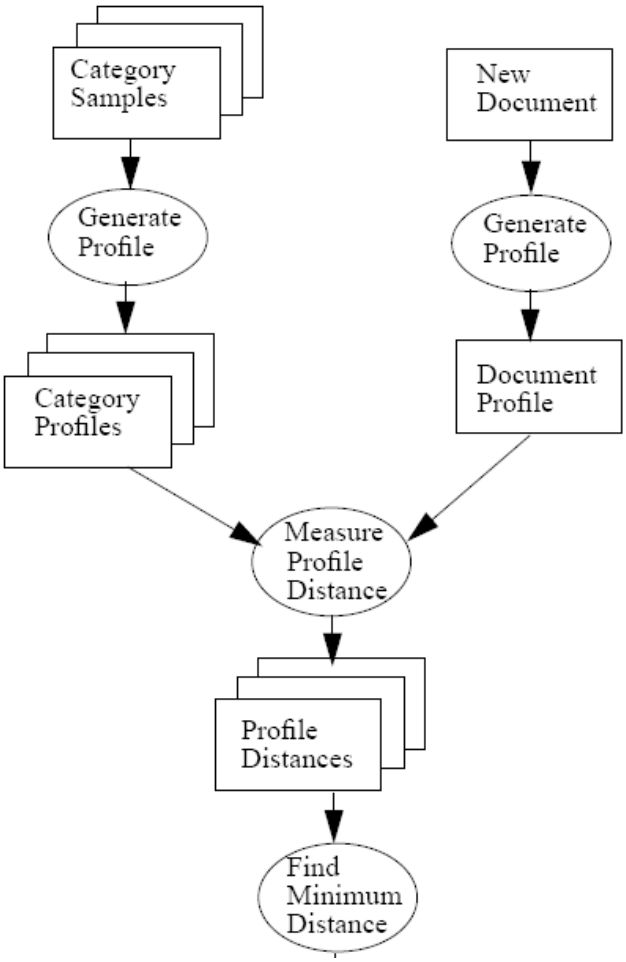


Figure 2-2 Dataflow for N-gram-base text categorization (Cavnar, 1994)

Including the blanks, the word "CENSUS" with a length of 6, has 7 bi-grams, tri-grams and quad-grams. Therefore, we know that if N is bigger than 1, a string with length *k*, padded with blanks, has *k+1* N-grams.

The "Generate Profile" process in Figure 2-2 is greatly simplified. It merely processes the observed text and counts occurrences of all N-grams. It is done by the following three steps:

1. The observing text is read in and all punctuation marks are deleted. The text is divided into separate tokens and padded with sufficient blanks before and after each word.
2. All possible N-grams are generated by scanning down each token. The N-grams are stored in a hash table. For each occurrence, the counter for the N-gram is increased.
3. The number of occurrences is sorted by reverse order, so the N-gram with the topmost occurrences ranks on the top.

The resulting file is an N-gram frequency profile for the input document. In category profiles and observed string profiles, some least frequent N-grams are discarded.

The “Measure Profile Distance” process in Figure 2-2 takes two N-gram profiles and calculates a rank-order statistic, which is called the “out-of–place” measure. It determines the ranking order distance of an N-gram between one profile and another profile. A simple example of this calculation can be found in Figure 2-3.

Finally, the “Find Minimum Distance” process in Figure 2-2 takes all the distance measures results and picks the smallest one.

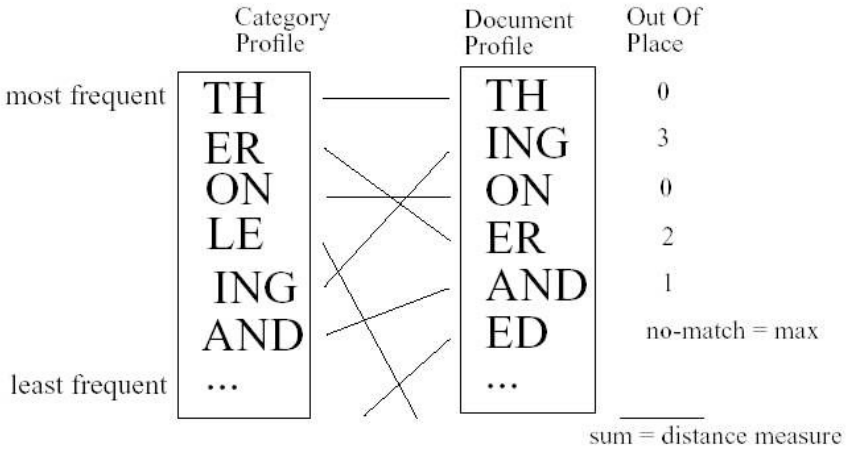


Figure 2-3 Calculating the out-of-place measure between two profiles (Cavnar, 1994)

The advantages of the N-gram-based approach are obvious. They include:

1. The approach is small and fast. The computational power required for generating the profiles and calculating the N-grams distances is very low.

2. The approach does not require advanced linguistic knowledge.

3. An N-gram-based approach is reliable since it tolerates a wide variety of textual errors. By using this approach, each string is divided into smaller parts. Any errors, such as misspelling or foreign words occurring in a string, have only a limited impact to a small part of a string. Thus, the rest of the strings remain intact.

4. This approach works well for very short input.

Cavnar (1994) test results showed that this method reaches a very high identification rate when the input text is short. The drawback is obvious in that it does not work on documents that are unable to tokenize because the N-gram-based approach relies on the correct tokenization. To overcome this problem, Abou-Assaleh, Cercone, Keselj & Sweidan (2004) and Vlado, Fuchuan, Nick & Calvi (2003) proposed approaches based on building a byte-level N-gram author profiles. These approaches use a byte as the basic token. Therefore, it does not require linguistic-dependent knowledge such as word and character separation.

2.1.4. Character Sets and Statistical Approach Combinations

Since tokenization is very difficult to apply to Asian languages, Kikui (1996) presents an approach to overcome the problem of tokenization. His algorithm consists of two major steps. In the first step, the algorithm tries to determine the character sets in order to set the East Asian text apart from European text. This can be identified when specific escape sequences occur since East-Asian characters are explicitly marked by escape sequences in the string.

The second step is identifying the language of the extracted text. It is constituted by three sub-steps. The first sub-step involves mapping the possible languages for the given coding system, following a heuristic rule. For example, if the document is encoded with US_ASCII, then it is not written in Chinese. In the second sub-step, the calculation of the probability of the decoded string for each language is conducted. For East-Asian

languages, a particular test string is compared to the statistical language models in order to determine which model is more similar to the test string. For the Western-European languages, the N-gram strategy is applied to the observing strings; therefore, the tokenization problem is solved. The third sub-step is responsible for selecting the highest probabilities and determining the language. The comparison is performed in a set of probability scores, and then the highest score is selected to identify a language. The above three sub-steps ensure that the language of the extracted text can be effectively identified.

Kikui (1996) employed 1340 articles found on the Internet. Among the articles, 700 articles were used for training and the rest were used for testing. The testing results showed the error rate for European articles was 4.8 percent and for the East-Asian articles, the error rate was 4.6 percent.

The character sets and statistical approach combinations are robust and easily extend to other languages. As well, this method can be extended to identify languages in multi-lingual text.

2.2. Web Information Extraction

Information extraction (IE) is a particularly useful sub-area of Natural Language Processing (NLP), and its goal is to locate specific information from a natural language document. IE is performed on free, structured and semi-structured text. According to Soderland (1999), a key element of the IE system is in applying a set of text extraction

rules or extraction patterns on the investigative documents to identify relevant information to be extracted.

Web documents have been the main target for the research on IE but they are different from the documents that are traditionally used in IE systems. A large portion of Web documents are semi-structured, although they may also have free and structured text. Moreover, the information on the Web is dynamic; it contains HTML tags, non-native words (e.g. English words in Japanese text), spaces and other format and control characters; and can be represented in different forms since it may be edited by any Internet resident. Hence, Web documents that are embedded with a large variety of noisy information have provided a special challenge for the IE field.

2.2.1. Applications

There are a number of different applications in which information extraction from the Web can be useful. Web Information extraction is often performed using wrappers. A robust wrapper must work with a standards-compliant HTML parser. This substantial component allows the wrapper to focus on the essential information extraction tasks such as a set of extraction rules and patterns. Several Web information extraction systems are used to extract information automatically from free, structured and semi-structured Web pages. We summarized the characteristics of these systems and listed them in Table 2-2. The table gives an overview of the type of texts the systems can handle.The IE systems listed in Table 2-2 use machine learning algorithms to generate extraction patterns for

Web documents. The first three rows are a group of open-source or noncommercial Web data extraction applications. They employ delimiter-based extraction patterns to extract information from the Web pages. The bottom two systems are commercial Web data extraction applications. Compared with the systems on the first three rows, they are more robust and flexible to handle a wider range of Web documents. Some popular search engines described in Steve & Lee (1999), such as AltaVista, EuroSeek, Excite, Google, HotBot, Infoseek, Lycos, Microsoft, Northern Light, Snap and Yahoo, performed a similar function for information extraction.

Name	**Structured**	**Semi-structured**	**Non-structured**	**Last Release**
Web-Harvest	Yes	Yes	Yes	2006
Deixto	Yes	Yes	No	2008
Odaies	Yes	Yes	No	2003
Unit Miner	Yes	Yes	Yes	2008
Kapow	Yes	Yes	Yes	2008

Table 2-2 The comparison of different information extraction systems

To summarize, Web information extraction employs different mature techniques, which are flexible and scalable, and are thereby able to adapt to the growth and the dynamics of the Web and to extract useful information from complex Web documents.

2.3. Web Language Identification

Since more and more textual data is making its way on-line, Web language identification has become increasingly important. Web language identification inherited the basic characteristics of text language identification but the complexity introduced by the Web domain requires a composite approach to identify languages within the noisy and

dynamic Web environment. To better understand Web language identification approaches, two main Web language identification applications, encoding auto-detection on Web browsers and N-gram based Web language detection, are introduced.

2.3.1. Encoding Auto-detection On Web Browsers

To deal with a variety of languages using different encodings on the Web today, most Web browsers, such as Microsoft Internet Explorer, Firefox and Mozilla, contain an automatic encoding detection option. In order to get the correct display result, browsers expend a lot of effort to automatically detect the encoding of Web pages. In general, two approaches are used to detect the encodings of Web pages.

The first approach detects the character encoding information. Some Web pages and HTTP headers contain explicit information that tells the Web browsers what encodings are used. For example, the World Wide Web Consortium (W3C) (2004) provides three common ways to declare encodings in XHTML and HTML, which include the encoding declaration in the HTTP header, the metadata element and XML.

The HTTP header is sent along with the Web page from a server even though the header is not part of the Web page. It may contain the character encoding information. Listing 2-1 shows an HTTP header with character encoding information.

```
HTTP/1.1 200 OK
Date: Wed, 05 Nov 2008 10:46:04 GMT
Server: Apache/1.3.28 (Unix) PHP/4.2.3
…
Content-Type: text/html; charset=utf-8
Content-Language: de, en
```

Listing 2-1 Character encoding declaration in HTTP header

A metadata element to explicitly declare the document's character encoding is used for HTML documents and XHTML documents served as text and HTML. An example of a metadata statement with character encoding declaration is shown in Listing 2-2.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

Listing 2-2 Character encoding declaration in metadata element

An XML declaration with an encoding attribute is used for XHTML served as XHTML and XML. An example of an XML declaration with character encoding declaration is shown in Listing 2-3.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Listing 2-3 Character encoding declaration in XML declaration

If a Web page or HTTP header does not include the above character encoding declaration information, the second approach is applied. It is responsible for determining the appropriate language encoding from the Web page text. The technologies used to determine the encoding of Web page text are varied on different Web browsers. Most of them are kept as a business secret, so they are not public. As one of the exceptions, Mozilla contributed a composite language and encoding detection approach to the public.

Shanjian and Katsuhiko (2001) present Mozilla's three methods to automatically detect the encoding of Web documents. The purpose of using three types of detection methods is to maximize their strengths and complement other detection methods. This composite approach using coding schema, character distribution and two-char sequence distribution is used in the Mozilla Web browser. Moreover, it can be easily adapted for other types of applications. Each method has its own strengths and weaknesses if it is used individually, but when the three methods are combined, the result is quite satisfying.

The coding scheme method is used to determine if an illegal byte or byte sequence exists in an examining string for certain encoding schema. Once it is encountered, we can confirm that the examining string does not belong to the certain encoding schema. Frank Tang (Netscape Communications Corp) implemented a parallel state machine that was used to detect character sets using coding scheme. The principal idea is that each coding schema has a corresponding state machine used to determine a byte sequence for this particular encoding. The examining string is input and fed to each available active state machine byte by byte. Based on its previous state and the byte it receives, the state of the state machine can be changed. There are three states which are: START, ME and ERROR.

1. START state: This represents the initial state or a state when a legal byte sequence for a character has been identified.

2. ME state: This indicates a unique byte sequence that belongs to the specific character set has been found. An immediate decision could be made that the correct encoding has been found.
3. ERROR state: This indicates that an illegal byte sequence has been identified by the state machine. When an ERROR state appears, an immediate decision could be made that this encoding is not a correct guess.

By this manner, one state machine will eventually provide a positive answer and the others will each provide a negative answer.

The character distribution method is similar to the common words text language identification approach and it is particularly useful for languages such as Chinese, Korean and Japanese.

The two-char sequence distribution method is similar to a statistical approach of text language identification where the first order Markov model is employed. A two-char sequence is defined as two characters appearing one by one in an input string, and the order is significant. Since the occurrence probabilities of the two–char sequence are varied in different languages, this method turns out to be extremely useful in detecting the character encoding.

As mentioned, each method has its weaknesses and strengths but if we employ the three together, they supplement one another and achieve a better encoding verification result. The comparison results of the three encoding identification methods are

represented in Table 2-3.

Method Name	Coverage	Strengths	Weakness
Coding scheme	Any kind of text	Multi-byte encodings Fast and efficient	Single–byte encodings
Character distribution	Typical text	Multi-byte encodings Fast and efficient	Single–byte encodings
Two-char sequence distribution	Typical text	Single–byte encodings Good results with small sample size	Multi-byte encodings

Table 2-3 The comparison of three encoding methods

Encoding auto-detection on Web browsers is a very quick and efficient approach but only as an encoding detection approach; it has a major limitation for language detection. For example, if some western European languages are encoded by the same coding schema, the encoding detection approach cannot tell exactly which language is used by detecting the encodings.

2.3.2. N-gram Based Web Language Detection

Bruno and Mario (2005) proposed an N-gram based approach that is complemented with a heuristics approach for automatic Web language identification. Besides the well-known N-gram approach, Bruno and Mario (2005) addressed some other issues when applying the language identification approach to Web documents. These include:

1. Extract the text, the markup information and metadata
2. Use available metadata information

3. Filter out automatically generated and frequently used strings

4. Weight N-grams based on HTML markup

5. Handle the insufficient data situation

6. Handle multilingualism cases

A robust parser is employed to remove tags and comments and retrieve the useful information from HTML documents before performing language identification. Having a robust parser is a very important aspect of dealing with Web data since it is capable of tolerating common errors associated with malformed HTML documents.

Some HTML pages contain metadata information to declare their languages; however, the default language value of most HTML editors is English. Therefore, the language declaration in meta-tag is taken into account for language identification but is not always reliable.

Another concern is that some strings, such as “This page uses frames,” that are automatically generated by the HTML editor, are repeated and have no meaning for the overall content and can be eliminated. Common words on Web pages, such as “Microsoft” or “Java,” are filtered out through this process. A small library keeps such sentences and common words which are filtered in a pre-processing stage.

Moreover, HTML provides a means to describe the structure of text-based information in a document. The text appearing in different parts of HTML have different

importance (Micha, Yungming & Weiyi, 1997). For example, the titles of Web pages are more important than other parts. More specifically, the titles are counted three times and metadata tags counted twice in N-grams.

In some cases, Web pages may contain little text. When the text within a Web page is less than 40 characters, the Web page is assigned with an “unknown language” label.

Finally, it is very common that a Web page contains multilingual text. This generates a problem if a single language is assigned to a given Web page. For example, some nouns, such as the names of places or people, are represented in foreign languages on Chinese Web pages. To better deal with this situation, the N-gram algorithm is re-applied to the largest continuous text block in a Web page. This text block is weighted three times more important than the normal text since the longest block will have a high probability of correctly describing the Web page and representing in its main language.

Experiments show that this approach achieves reasonable accuracy in identifying different languages on Web pages. Although the achieved results are good enough for the system to be used effectively, the accuracy is still lower than the pure text language identification results. This is mainly caused by the much noisier nature of Web pages that have been processed. Table 2-4 shows twelve different language identification results which are obtained by the N-gram based Web language detection system with the best setting.

The approach is a comprehensive approach for Web language identification but the main problem of this approach is its inability to identify the languages, such as Chinese and Japanese, which are hard to tokenize. Bruno and Mario (2005) mentioned that they could not find an appropriate collection of Web pages to use as the "golden standard." They organized a test collection and manually created HTML documents. Compared with real Internet Web documents, their test collection may achieve better language identification results.

Language	**Dan**	**Dut**	**Eng**	**Fin**	**Fre**	**Ger**	**Ita**	**Jap**	**Por**	**Rus**	**Spa**	**Swe**
Chinese	0	0	0			1		6	1	12		
Danish	480	48		6		3	5			12	2	22
Dutch	2	447			2							
English	5	3	449	31	1	6	12	4	10	37	24	17
Finnish				421								
French				4	495	1	4				2	
German				7		482	12		9	16	9	
Icelandic	1											
Italian				1			403					
Japanese								475				
Portuguese							20		475	6	15	
Russian										444		
Spanish						2	42				435	1
Swedish				16						2		425
Unknown	11	2	1	14	2	5	2	15	5	1	13	35
#Correct	480	447	449	421	495	482	403	475	475	444	435	425
%Correct	**96%**	**89%**	**100%**	**84%**	**99%**	**96%**	**80%**	**95%**	**95%**	**89%**	**87%**	**85%**

Table 2-4 Results for the N-gram based Web language identification

2.4. Statistics of Web Language Distributions

The Web is a universal information space and Web content contains different languages from all over the world. It is well known that in the past, the majority of Web content originated from a small group of English-speaking countries, chiefly the United States. With increasing Internet coverage, however, language diversity on Web content is becoming increasingly globalized. Two current projects examine the internationalization of Web content, and in particular, the language distribution.

2.4.1. Language Statistics by the Online Computer Library Center

The Web characterization project was conducted by the Online Computer Library Center (OCLC), (2002). To minimize time and computer resources, they eliminated the portions of the addresses that contained no reachable hosts from the whole IPv4 set. Then, 0.1 percent of the sampled IP addresses (4,294,967) were randomly drawn. The samples for Web characterization were obtained by attempting to connect to the Port 80 at randomly generated IPv4 addresses. For each sampled IP address, if an HTTP response code of "200" was returned, a Web site was captured and stored for further analysis. Table 2-6 shows the statistical number of unique Web sites. Since the number of the sampled IP addresses is quite small, Table 2.5 shows their calculated results. Only a small fraction of the websites was actually visited (Edward, Patrick & Brian, 1997).

Year	Unique Web Sites Number
1998	2,636,000
1999	4,662,000
2000	7,128.000
2001	8,443.000
2002	8,712,000

Table 2-5 Number of unique Web sites

The language statistics report, as a part of their research effort, is published on the OCLC website. Table 2-6 shows the distribution of languages across public Web sites for 1999 and 2002 clearly shows that English remains as a major language of Web content.

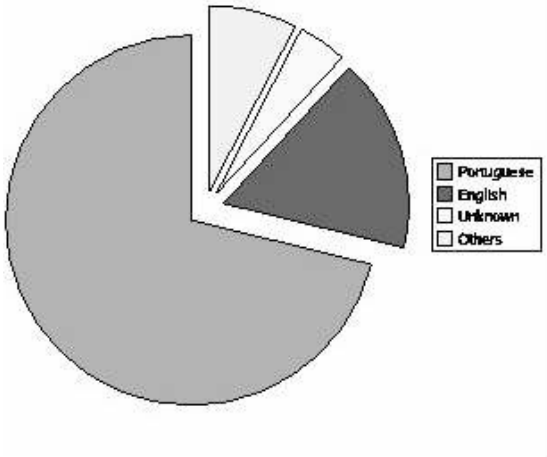
1999		**2000**	
Language	**Percent of Sites**	**Language**	**Percent of Sites**
English	72%	English	72%
German	7%	German	7%
French	3%	French	6%
Japanese	3%	Japanese	3%
Spanish	3%	Spanish	3%
Chinese	2%	Chinese	2%
Italian	2%	Italian	2%
Portuguese	2%	Portuguese	2%
Dutch	1%	Dutch	1%
Finnish	1%	Finnish	1%
Russian	1%	Russian	1%

Table 2-6 Language distribution results from OCLC

The scope of these randomly generated IP addresses for searching the Web sites is hardly representing the entire Web; thus, its language statistics results cannot completely depict the trend of the global Web language distribution. Moreover, this project has not been updated since 2003 and is currently inactive.

2.4.2. Language Distribution by Bruno and Mario

The language distribution results in Bruno and Mario (2005) particularly focus on the Portuguese Web, a large collection of about 3.5 million pages hosted under the ".PT" domain. Figure 2-4 shows that a significant portion of Portuguese Web pages is actually written in foreign languages, especially English.



Language	# Documents
Portuguese	(71%) 2334389
English	(17%) 552761
unknown	(4%) 142509
German	(3%) 108277
Spanish	(1%) 41813
French	(1%) 21411
Romanian	(0%) 12555
Polish	(0%) 11561
Swedish	(0%) 5320
Italian	(0%) 4454
others	(1%) 36841
Norwegian	(0%) 2951
	(100%) 3274842

Figure 2-4 Language distribution on the Portuguese Web

This is a very good example for language distribution in a particular scope since it has the wide coverage and comprehensive language identification algorithm to support; however, as mentioned, the N-gram based approach is not able to identify languages such as Chinese and Japanese, which are hard to tokenize. In other words, this approach is not appropriate to use on universal Web pages.

3. Web Language Identification and Distribution Design

In this chapter, we first give an overview of the design of Web Language Identification and Distribution System (WLIDS) from the architecture perspective. Then, we discuss the detailed functional design of WLIDS.

3.1. Overview

The main task of WLIDS was to perform Web language identification, geographical and language distribution based on 24.2 million found Web servers and their portal Web pages that were obtained by a Web Census (Benoit, Slauenwhite, & Trudel, 2006). The Web census attempted to connect to each possible IPv4 address and inquired for the presence of a Web server. Once a Web server was detected, a copy of the portal Web page for each found Web server (no images) was recorded to the Web census database. WLIDS played an important role for further Web Census investigation, such as Web categorizations and classifications, since the Web investigation must be customized according to the conventions of different languages and cultures. Moreover, with the proper modifications, the WLIDS can be extended to other Web language identification projects, no matter where the Web pages are stored.

WLIDS was a simple client and server model. The logic function of WLIDS was a client side application written in Java. The database server contained the source data, middle process data and final results of language identification and distribution. JDBC

was used to handle the communication among the server databases. This simple client and server model was able to be deployed on multiple machines with different combinations for improving the ability of parallel processing.

In WLIDS, the statistics of Web language distribution highly depended on Web language identification results. Accuracy, efficiency, extensibility and the ability to classify languages without linguistic knowledge were the main concerns in our Web language identification approaches. In order to achieve these goals, WLIDS combined multiple approaches from various domains such as text language identification, Web information extraction and Web language identification to efficiently generate a high accurate Web language identification results.

In the rest of this section, we briefly introduce the WLIDS from the high level design perspective, including its components, general task flow and database.

3.1.1. Components

WLIDS has four basic components which were responsible for geographical location distribution, information extraction, language identification, and language distribution. All of these components interacted with the Web Census database server, so this database was a common component of WLIDS.

The first component, geographical location distribution, determined the geographical location of all the Web servers by their IP addresses. The distribution results were used to observe the geographical dispersion of the Web servers. The second component, language

identification, employed multiple approaches to efficiently achieve accurate language identification results for the portal Web page of each Web server. The third component, language distribution, carried out two kinds of language distributions based on language identification results and geographical location distribution results. The last component, the relational database, was designed to contain data and support efficient queries from the other components.

3.1.2. General Task Flow

From the general workflow of WLIDS shown in Figure 3-1, we have a better vision of the structure and the relationship among the three basic components of WLIDS. The language identification component has two parts. The first part, information extraction, filtered out a large variety of noisy information embedded in Web pages and extracted the useful information for language identification at the next step. Moreover, when the language of each Web page in the Web Census was determined, the general language distribution results were achieved. All the results from the geographical distribution and language identification components made determining the language distribution in a specific geographical location possible.

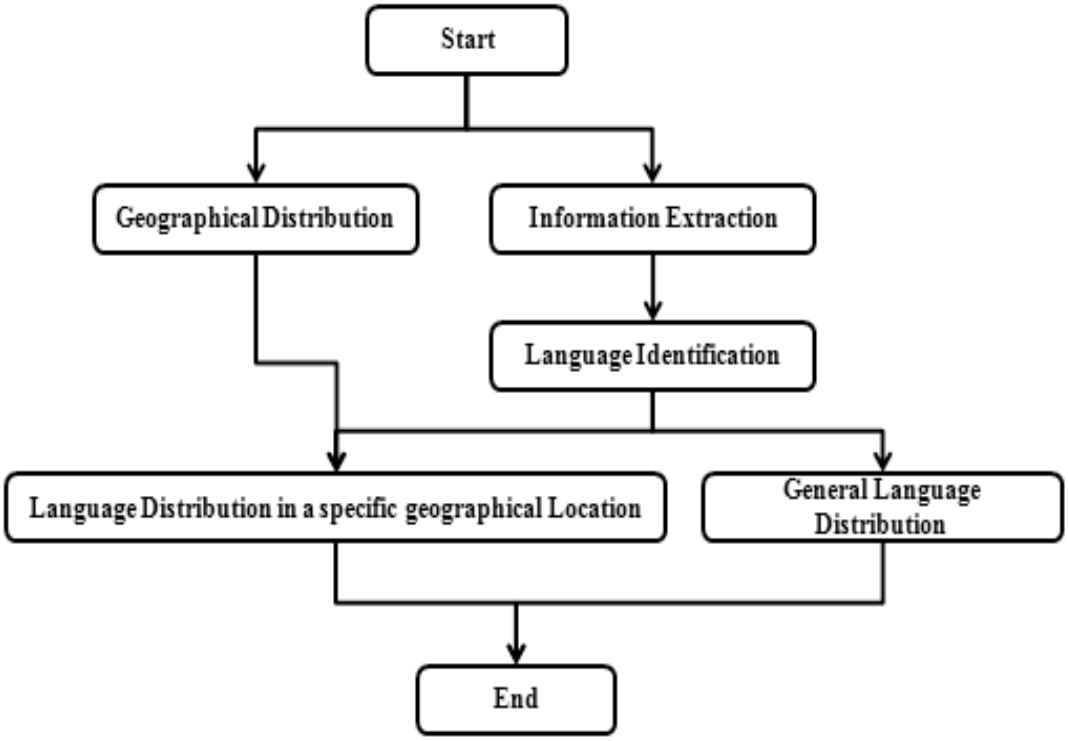


Figure 3-1 The general workflow of WLIDS

3.1.3. Database

All Web Census information was contained in a relational database using IBM's DB2 database management system. In order to conveniently access the Web Census information and store the data of WLIDS, the WLIDS data tables were created in the Web Census database but different database schemas were used to separate them from the original Web Census data tables. We discuss the design of the database in detail in Section 3.2.5.

3.2. Functional Design

This section provides detailed information on how we performed Web language identification and distribution. In the rest of this section, we present the functional design of geographical distribution, Web information extraction, Web language identification, statistics of language distributions and the relational database for WLIDS.

3.2.1. Web Geographical Distribution

The Web geographical distribution determined the physical location of all the Web Census servers on which the Web content was stored. It was inspired by some widely used address locator tools such as GEOBYTES (2006) and HOSTIP.INFO IP (2007). They determined the countries of Internet visitors or hosts based on the IP addresses that they were using. In general, the basic method used to determine the location of the IP address was that it matched an IP address within an IP address range in the geographical distribution database and output the geographical location of this matched IP address range. This simple approach suited for matching geographical locations from a single IP address; however, it was inefficient for matching the geographical locations for 24.2 million IP addresses of Web Census servers.

To achieve the geographical distribution of Web Census servers, we used the MaxMind GeoLite Country Database (2008) in which each entry consisted of a IP address range and a country code and ISO 3166-1 database which consisted of a set of two capital letters to represent countries. The MaxMind GeoLite Country Database (2008)

declared that it achieved 99.3 percent accuracy for country distribution based on the IP address range. We designed three functions to improve IP addresses geographical location matching and achieve statistics of Web geographical distribution results:

1. IP address to IP number mapping
2. IP number country distribution
3. Summary of country distribution

IP address to IP number mapping was responsible for converting all the IP addresses of 24.2 million Web servers into IP numbers. Based on this mapping, an IP address as a unique identity for a Web server was replaced by a corresponding IP Number on the rest of WLIDS for increasing the data query. Since the IP address is a character data type and the IP number is an integer data type, the database management system performs much faster to compare an integer number than a character string in the database table. For example, an IP address "61.65.0.245," is the IP Number "1027670261." Finding this IP address between "Beginning IP Address" and "Ending IP Address" in Table 3-1 is much slower than finding the IP Number between "Beginning IP Number" and "Ending IP Number." It is very useful, therefore, to replace IP addresses with IP numbers if we perform IP address lookups using the database.

Beginning IP Address	Ending IP Address	Beginning IP Number	Ending IP Number	Country Code
61.56.0.0	61.67.255.255	1027080192	1027866623	TW
61.14.132.16	61.14.132.31	1024361488	1024361503	AP
62.13.160.0	62.13.191.255	1041080320	1041088511	IT
62.23.36.32	62.23.36.55	1041703968	1041703991	FR
217.146.16.0	217.146.17.255	3650228224	3650228735	US

Table 3-1 A example of an IP address and IP number matching

An IP address was converted to an IP number using these formulas:

Assume

IP Address=W.X.Y.Z (1)

Where $0 \leq W, X, Y, Z \leq 255$

Then

IP Number=$(256^3)*W+(256^2)*X+(256^1)*Y+(256^0)*Z$ (2)

Similarly

IP Number=16777216*W+655536*X+256*Y+Z (3)

According these formulas, an IP address has only one IP number to represent it.

The IP number country distribution function found the IP number range for each given IP number in order to locate the country code. If an IP number range can be found, its corresponding country code was stored into a database table. Otherwise, an "unknown" label was added into this database table, too.

We summarized the country distribution by counting the number of servers for each country and stored the results in a database table. Therefore, we were able to sort the country distribution results by the number of servers.

3.2.2. Web Information Extraction

Web information extraction extracted useful information from the portal Web page (text only, no images) of each found Web server for language identification. Unlike the traditional Web information extraction, it did not require taking the content with a

specialized topic from the Web. Instead, it extracted Web language identification sensitive information from Web pages, such as textual data, language declaration and character declaration. Web information extraction included three steps:

1. Filtered out invalid portal Web pages
2. Extracted common text, metadata and HTML tag from Web pages by HTML or XML parsers
3. Filtered out noisy information

Portal Web pages without the HTTP response code of “200” were excluded from language identification. For instance, some Web portals with the HTTP response codes “405” (no permission to access the Web page) and “404” (the Web page cannot be found) can be excluded. Our research found that except the “200” HTTP response code that means the request was successful and information was returned, other HTTP response codes indicate that connection attempts encountered some problems. For achieving a better Web language identification result and to reduce unnecessary workload, only the portal Web pages with the “200” HTTP response code were used for language identification.

An error tolerant HTML parser was used to extract common text, metadata and HTML tag from portal Web pages in the second step. Instead of parsing well-formed XML and HTML documents, this compliant Simple API for XML (SAX) parser was able to parse malformed HTML documents. The event-based SAX parser performed faster than a Document Object Model (DOM) parser, especially for large HTML documents

where the DOM parser hit virtual memory or consumed all available memory (Steve, 2001).

We further filtered out noisy information, such as common words and unusable symbols from the common text. A small common words library (Bruno and Mario, 2005) was kept to filter out the common words from the common text. These common words, such as “home page,” “Microsoft,” “Linux” and “JAVA,” that occur in global Web pages in different languages should be removed. Another small symbol library was kept to eliminate those symbols that are useless for language recognition. These symbols included some HTML special characters, decimal numbers, space separators, math symbols, currency symbols and punctuation.

3.2.3. Web Language Identification

Web language identification employed multiple methods to determine the language of Web portals. Basically, these approaches included two parts. The first part utilized available metadata and HTML tag information, the knowledge of several corresponding ISO standards and W3C conventions for language identification. The second part utilized the improved N-gram based text language identification approach to determine the language of the Web common text. The Web common text is the visible text that can be manipulated by HTML code but does not include “Alt” text that is displayed when an image cannot be rendered. Moreover, a language identification priority list, which was summarized from accuracy comparison tests on all language identification approaches in

WLIDS, was used to pick a proper language identification result for a portal Web page.

The general workflow of language identification in Figure 3-4 shows that after Web information extraction, the workflow of language identification worked in two manners, serial and parallel. The serial manner sequentially performed three language identification methods. Once one of the methods obtained a result, it ceased the rest of language identification methods and recorded the result to database. The parallel manner executed in parallel for three language identification methods. In the end, one of the language identification results was picked as a final result according the language identification priority list. A switch was designed to select which manner should be employed.

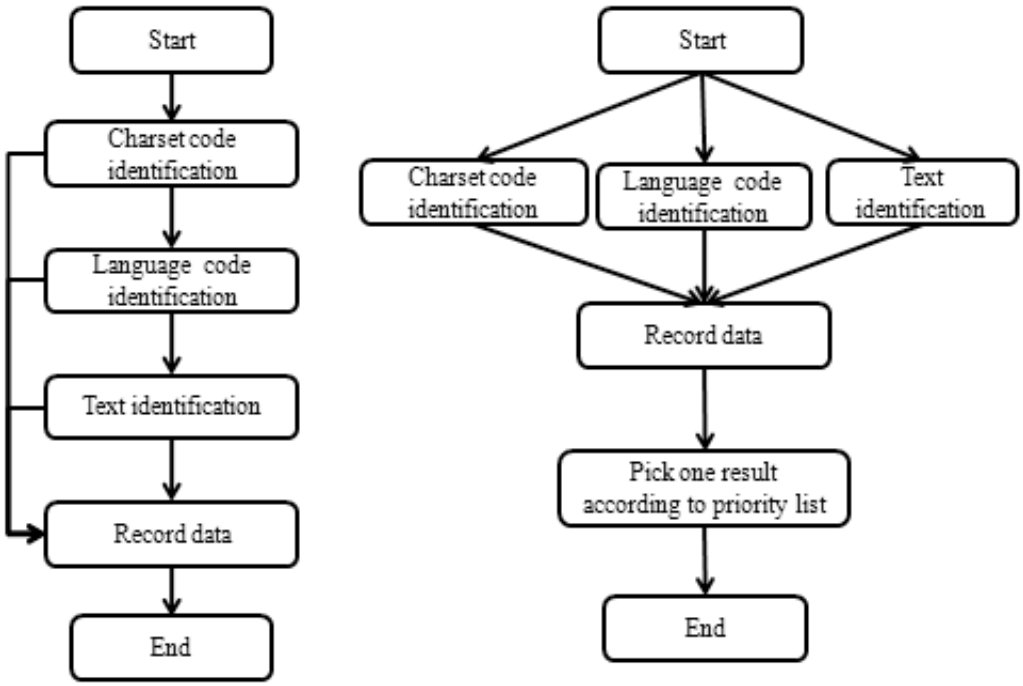


Figure 3-4 The general workflow of the language identification

3.2.3.1.Metadata and HTML Tags

Some metadata elements and HTML Tags of Web pages explicitly declare the document's character encoding and language code. This information is useful for determining the languages of Web pages.

In the background of Web language identification section 2.3.1, we listed a few character encoding declaration styles; however, from a character encoding code, we cannot always identify a unique language. For example, if character encoding code is “UTF-8,” a variable-length character encoding for Unicode, we are not able to tell which language is used. Thus, a set of character encoding codes that were capable of determining their corresponding languages were kept in a small library. If a character encoding code was detected in the metadata elements and if this character encoding code existed in this library, its corresponding language can be found.

From the W3C (2007b), we knew that the language code declaration is analogous with character encoding declaration. Similarly, we kept a library that contained a set of language codes and their corresponding languages. If a language code was detected in the metadata elements or HTML tags, its corresponding language can be found in this library. More specifically, we matched the language codes by two ISO 639 standards, ISO639-1 and ISO639-2, which were widely used on language declaration on Web pages. According to the Library of Congress (2006), ISO 639 uses codes for the representation

of names of languages. ISO 639-1 standard represents language code by two lower case letters and ISO 639-2 standard represents language code by three lower case letters.

3.2.3.2. Common Text

We employed an improved N-gram based text language identification approach to determine the language of the common text. This approach not only inherited advantages of the N-gram based approach but it also overcome its tokenization weaknesses and improves the language identification accuracy on the short text. As well, it brought new features to improve accuracy and enlarge the range of language identification. Compared to traditional N-gram based approach, six advanced features include:

1. Handling insufficient common text data situations: In the case that the extracted common text from a portal Web page was less than 40 bytes, we simply assigned such pages with an “unknown language” label. Once the page was assigned this label, it no longer participated in language identification, and its exclusion improved the accuracy of language identification.
2. Overcoming the weakness of tokenization in the N-gram approach: We were not trying to tokenize each single word in documents in different languages; however, we used one byte as a standard token to generate the N-gram language category profiles and observe document profiles.
3. Eliminating the dissatisfactory results to improve short text identification accuracy in the noisy environment: Based on the statistics test results, a border

value was set to determine if the best guess language can be returned as an identified language. More specifically, if the profile distance of the best guess language was larger than this border value, no language was identified.

4. Balancing speed and accuracy during the language identification: A few values were able to adjust for balancing the speed and accuracy of language identification. They were the N value for N-gram profiles, the topmost number of N-grams that should be kept in profiles and N-grams least appearance times that was used to discard the less frequent N-grams in profiles.
5. Identifying a language using multiple category profiles (language models): In some special languages, corpora were encoded by different encoding schemas before generating category profiles to enlarge the coverage of language identification. For example, a Chinese document encoded by “gb2312” encoding schema cannot be recognized by a “Chinese-UTF8” category profile but it can be identified by “Chinese- gb2312” category profile. Therefore, for identifying a language, we may use multiple category profiles.
6. Handling multilingual documents on the web: We assigned the full content of a given Web pages to one single language since foreign words are very common over Web pages. On the other hand, we were able to output the most likely languages on a Web document other than the best guess language for future research based on a typical value. This typical value was used to determine how much worse result must not to be mentioned as an alternative.

3.2.4. Statistics of Web Language Distributions

We observed the statistics of Web Language Distribution from two different scopes. One was from the global scope, in which the general language distribution examined all language identification results on the portal Web page of each found Web server with HTTP response code of “200”. Another scope was from an individual country, in which we examined language distribution for the specific geographical location.

3.2.4.1.General Language Distribution

This general language distribution was based on all of our language identification results that performed on portal Web page of each Web servers found by Web Census. Once all the Web Census pages were characterized by the language they used, the number of Web portals for each language was counted and stored in a database table as the general language distribution statistics results. Since the collection of Web Census portal Web pages has comprehensive coverage, these statistical results can depict the status of the language distribution of global Web servers’ portal Web pages. Based on the language distribution results, we can carry out more research on status and trends of Web internationalization. Ideally, Web content should express a broad range of languages and the Web language distribution results should reflect status and trends of the Internet community.

3.2.4.2.Language Distribution in a Specific Geographical Location

The language distribution in a specific geographical location determined the language distribution status of Web servers' portals in a specific country. Based on the Web geographical and general language distribution results, Web pages in a specific country can be distinguished from other Web Census pages and these Web pages can be characterized by the language they use. Then, the number of Web pages for each language was counted and stored in a database table as the language distribution statistics results for a specific country.

According to these statistics, we can carry out further research and investigation. For example, in a country such as Canada, where there are diverse multilingual populations and two official languages, does the language distribution of Web servers' portal Web pages reflect the diverse multilingual status? Moreover, in some non English speaking countries, does English become the top foreign language?

3.2.5. Relational Database

WLIDS used a server-side relational database to contain the source data, the middle process data and final results. In this section, we first introduce the database Entity-Relational model of WLIDS. We then discuss several design approaches that were used to boost the database performance and the communication between the client and database server.

The Entity–Relational model in Figure 3-5 helps to explain what information was stored in the WLIDS. The database tables on the left were used to store the data for Web language distribution and the tables on the right were used to store the data for geographical distribution. They were connected by the primary key of "Server IP and IP number mapping" which uniquely identified the Web server.

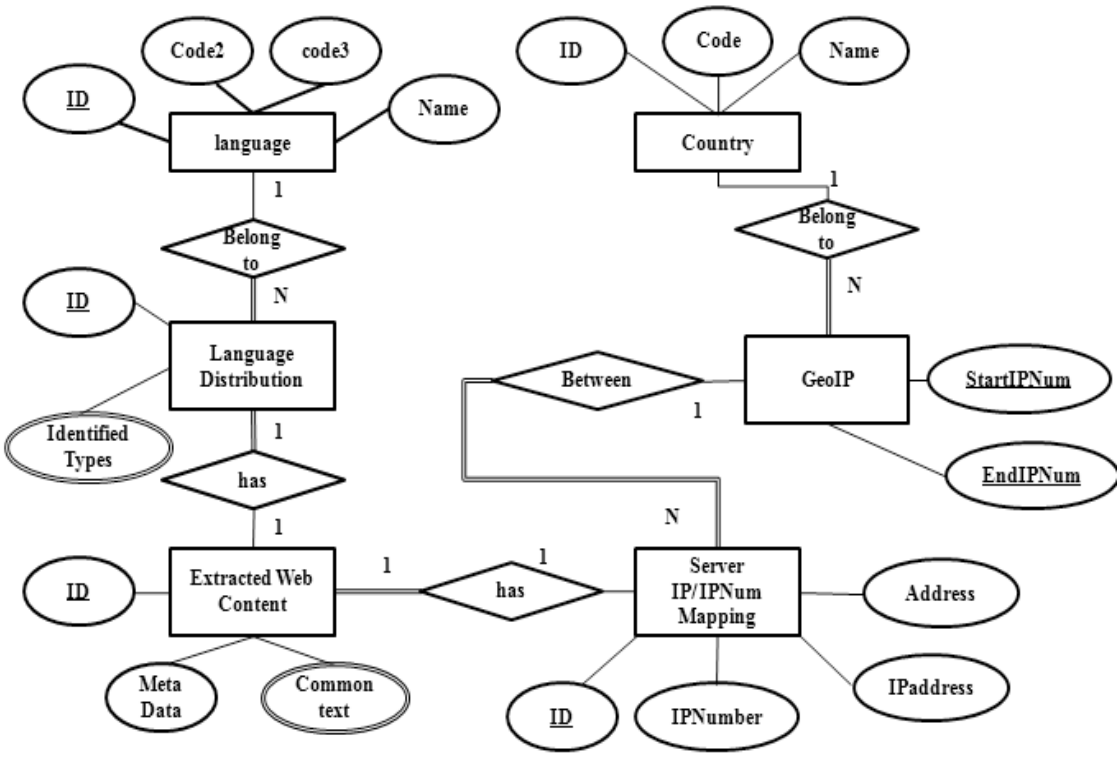


Figure 3-5 The Entity-Relational Model of WLIDS

For any given Web server in the Web Census, an **IP address** and an **Address** can be combined to give us a full IP address. In WLIDS, these full IP addresses were replaced by integer IP numbers in "Server IP and IP number mapping" in order to speed up the data query. Each IP number uniquely identified a Web server or its portal Web page in geographical and language distribution.

For Web language distribution, the HTML code of the portal Web page of each Web server was extracted. The extracted metadata and common text information were stored in the "Extracted Web content" table before performing language identification. After language identification, the results were stored in the "Language Distribution" table. An **identified type** attribute in this table indicated what kind of language recognition approaches were used to achieve the results. Moreover, the two-letter and three-letter language codes and their full names can be found in the language table.

For geographical distribution, the "GeoIP" table contained IP number ranges and their corresponding country IDs where the attributes **StartIPNum** and **EndIPNum** contained the IP number ranges. The corresponding country code and the country name of each country ID can be found in the Country table. Then, for a given IP number, if an IP number range and its corresponding country ID were found in the "GeoIP" table, the result was recorded to the "Country Distribution" table. Otherwise, the IP number was marked with an "unknown" country label and recorded to the "Country Distribution" table as well.

4. Implementation

This chapter presents implementation details of the Web Language Identification and Distribution System (WLIDS). It describes the development (testing) environment and the technologies used for WLIDS. The problems encountered during the implementation and our solutions are represented, as well.

4.1. Environment

Design of WLIDS followed the client and server model. The logic function of WLIDS was a client side application written in Java. The database server was designed to support the logic function of the client application. The description of the testing environment in Table 4-1 shows that WLIDS was able to be deployed in multiple machines.

List of Requirements	**Testing Environment**
Hardware:	
Client-side	PC with at least 256M memory (6 PCs were used)
Server-side	PC with 2G memory
Software:	
Operating System→Client-side	Linux, Windows XP or Vista
Operating System→Server-side	Linux, Windows XP or Vista
Additional software→Client-side	Java SE Development Kit 6 and Perl Interpreter (version 5.08 or higher)
Relational database→Server-side	IBM's DB2 database management system (version 8.0 or higher)

Table 4-1 The testing environment of WLIDS

4.2. Technologies

We implemented WLIDS using Java-based technologies, and chose Eclipse as the software development tool. In addition, a Perl script was used for text language identification because the scripting languages achieved a better efficiency for processing text content. Table 4-2 lists major technologies used for individual features.

Feature of WLIDS	Major Technologies
Common features	Java for implementing business logic
- Logging system	Apache commons-logging API (Apache, 2007)
- Database communication	JDBC for accessing the database
	Proxool, a Java connection pool for optimizing the database connections (Proxool, 2007)
Language identification and distribution	Java for implementing business logic
- Web information extraction	HTML Parser for extracting different portions of Web pages. (HTML Parser, 2006)
	Java Regular Expressions for filtering out noise information on the Web common text
- language identification	Optimized Textcat Language Guesser for generating the language models and identifying languages of the Web common text. (Textcat, 1997)
	Encoding detection approach for identifying languages of encoding declarations on Web pages.
	Language detection approach for identifying languages of language declarations on Web pages
- Summary of language distribution	Java for implementing business logic
Geographical Distribution	Java for implementing business logic
- IP to IP number mapping	An IP to IP number formula for converting an IP address to an IP Number
- Web servers country distribution	Java for implementing business logic
- Summary of country distribution	Java for implementing business logic
Server-side relational database	IBM's DB2 for building the database
	JDBC for accessing database

Table 4-2 Major technologies for WLIDS features

4.3. Web Language Identification

Our original design performed Web information extraction and language identification separately and asynchronously; therefore, we ran the Web information extraction program first before the language identification implementation was ready. The results of Web information extraction were recorded in the database for language identification and the future Web classification. By the time the Web language identification was completed, we had finished language identification on the extracted information from Web information extraction; however, the diversity of Web pages on the Internet caused a lot of issues during the implementation of Web information extraction. Some of the Web pages dramatically impacted the language identification accuracy, and they had not been discovered until we were testing the language identification code. The final version of Web information extraction and language identification was completed at the same time. Due to a limited time frame, it was no longer efficient to perform these two approaches separately.

Our new solution kept the intention of our original design but joined these two approaches together to improve the processing performance. After the Web information extraction, the results were still recorded to the database for further research but in the meantime, the results were sent to language identification. In this solution, we improved performance by eliminating duplicate database processing between these two approaches by accessing the database only once. Moreover, a control was added to separate the portal

Web page of each Web server to different ranges for supporting the parallel processing.

A data flow diagram in Figure 4-1 represents the “flow” of data through language identification based on our new solution.

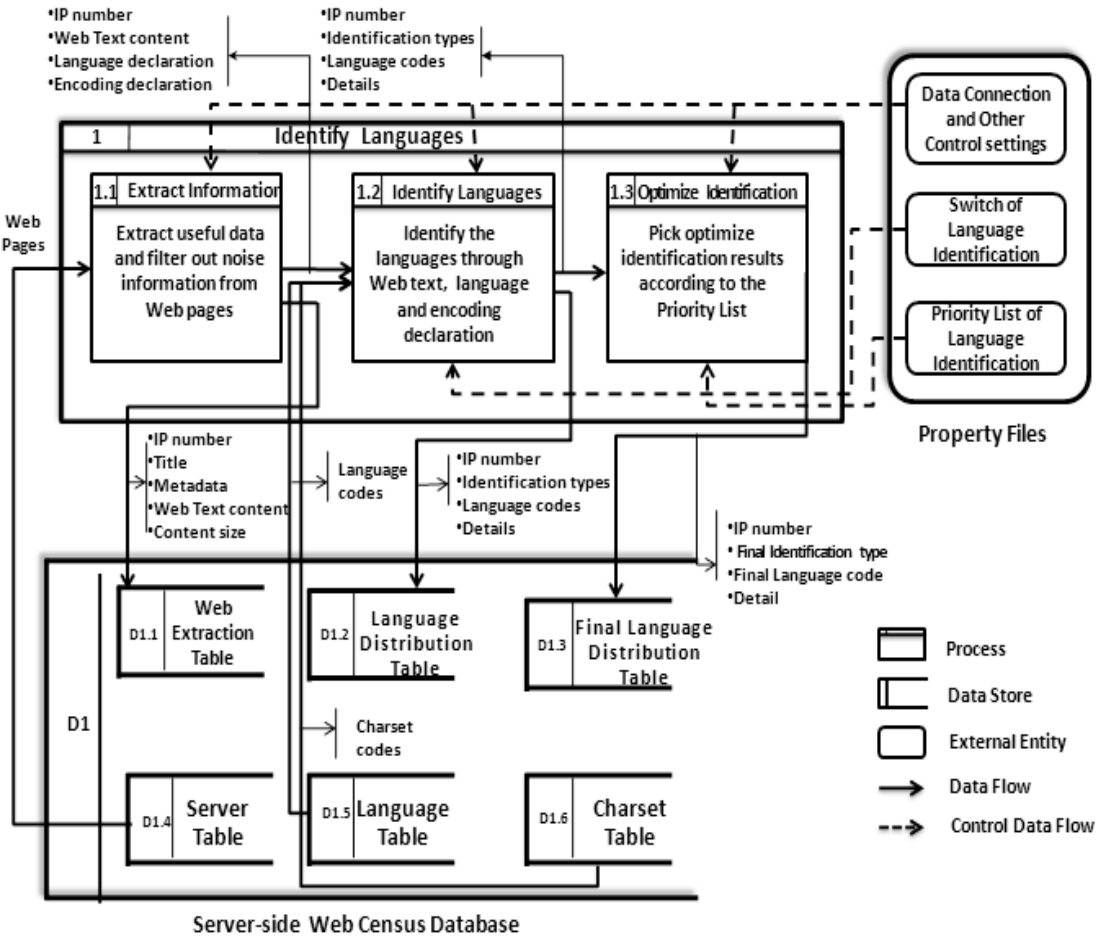


Figure 4-1 Data flow of language identification

The data flow Figure 4-1 indicates that for information extraction, process 1.1 was fed by Web pages from Server table. Once the useful data was extracted from Web pages and further noise filtration on extracted data had been performed, the extracted information was recoded to the “Web Extraction” table. In the meantime, the useful information including Web text, language declaration and encoding declaration was sent

to the process 1.2 for language identification. By default, the language identification process identified languages of Web pages through three methods: web text, languages and endcoding declaration. To speed up language identification, a control switch was deployed to simplify the language identification process. The details will be discussed in section 4.3.2. For each identified Web page, the language identification process outputed its IP number, language identification types and corresponding language codes. Similarly, the output data was recorded to the database table and flowed to the next step, process 1.3. This process picked a single language identification result for each identified Web page according to a priority list of language identifications. The output, the final language identification results, were recorded to the database as well.

4.3.1. Web Information Extraction

Web information extraction was responsible for extracting useful information from Web pages and further filtering out noisy information from the extracted information. The quality of the extracted information dramatically impacted the accuracy of Web language identification. Furthermore, the diversity of Web pages on the Internet made the implementation of accurate data extraction and noise information filtration extraordinarily difficult. In order to solve the diversified problems of Web information extraction, multiple technologies and methods were employed during the implementation. Figure 4-2 illustrates the overall workflow for the implementation of Web information extraction. The implementation was trained on 30,000 random Web pages.

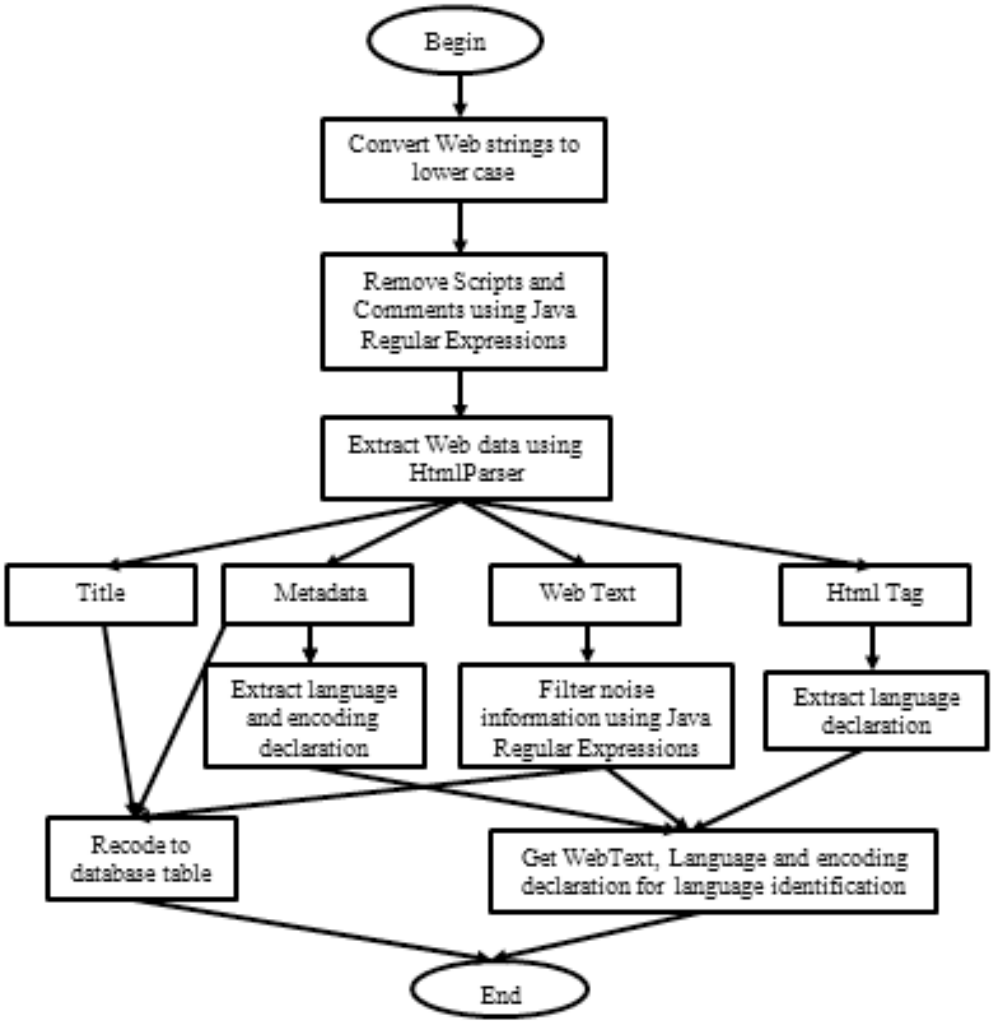


Figure 4-2 The workflow of Web information extraction implementation

When a new web page arrived for Web information extraction, to obtain better matching and locating information from it, we converted it to a lower case string. The scripts and comments had to be removed from the Web page string before HTML parsing because the HTMLParser (HTML Parser, 2006) used for Web data extraction may mistakenly categorize the script content and comments into Web text. If most of the Web text came from the script content and comments, the language identification results on Web text would be completely distorted. The HTMLParser considered the text between

two HTML tags as Web text. For example, the JavaScript content between the “<script> </script>” tags in Listing 4-1 and the comment between the “<P> </P>” tags in Listing 4-2 were falsely classified as Web text during the HTML parsing.

```
<script type="text/javascript">
      document.write("Hello World!")
</script>
```

Listing 4-1 The HTML script example

```
<P>
    <!-----A example:    a comment inside the HTML tags ------->
</P>
```

Listing 4-2 The HTML comment example

The process in Figure 4-2 labeled “Extract Web data using HTMLParser” was used to extract Web data, title, metadata, Web text and HTML tags. For metadata and the HTML tag, we further extracted the language and encoding declaration attributes following the guidance of W3C (2004) and W3C (2007). Three simple cases of language and encoding declaration are shown in Listing 4-3.

```
1.  Matadata--> Language Declaration
<meta http-equiv="Content-Language" content="en-US" />

2.  Matadata--> Encoding Declaration
<meta http-equiv="Content-Type" content="text/HTML; charset= gb2312"/>

3.  HTML tag--> Language Declaration
<HTML lang="en-CA" xml:lang="en-CA" xmlns=
"http://www.w3.org/1999/xHTML">
```

Listing 4-3 Three language and encoding declaration examples

For the Web text, we further filtered out varied noise information in order to improve the language identification accuracy according to the functional design in section 3.2.2. A

technology, Java regular expressions, was employed in our implementation to improve the filtration efficiency. In Table 4-3, a filtration comparison test between a normal Java string operation and Java regular expressions based on 30,000 randomly picked Web pages shows that Java regular expressions remarkably speeded up the filtration.

Technology	**Total Filtration Time for 30,000 Web pages (Sec)**	**Average Filtration Time(Sec)**
Normal Java string operation	9412	0.314
Java regular expressions	1216	0.041

Table 4-3 A comparison of Web common text filtration

Last, the title, metadata and "clean" Web common text were recorded to the database table for further Web classification research and the language and encoding declaration information and "clean" Web common text were ready for language identification.

4.3.2. Language Identification

The implementation of language identification was responsible for distinguishing the languages of language and encoding declaration and Web common text that were filtered out from Web information extraction. During language identification, we improved the function of the TextCat language guesser, an implementation of the text categorization algorithm presented in Cavnar (1994) for text identification. According to the characteristics of the Web pages, we added various improvements to increase language identification accuracy and speed. They include:

1. Added a priority list according to the accuracy investigation on three types of language identification based on 30,000 randomly picked Web pages.

2. Implemented a switch to select if we should perform three Web language identification approaches or pick a shortcut that was allowed to terminate the language identification process when one of the approaches obtained the result for Web language identification.
3. Distinguished the languages of the diverse language declaration based on W3C internationalization convention (W3C, 2007a) and data training results on 30,000 randomly picked Web pages. For instance, the English language declaration has multiple styles such as “en,” “en-US” and “English.”
4. Matched a character encoding to a single language according to the character encoding library that was constructed based on the W3C character set library (W3C, 2002) and Microsoft’s character set recognition library (Microsoft, 2008).
5. Implemented the code for generating multiple category profiles that were encoded by different encoding schemas for each language.

The language identification training results based on 30,000 randomly picked Web pages are shown in Table 4-4. Five hundred random cases were picked for each approach during correct rate investigation. For Web common text identification, the category files (language models) contained the top 400 ranked N-grams. The results show that the text identification had more coverage than the other identification approaches; however, the text identification coverage rate has only 75.97 percent coverage since some Web common text was too short. The Web common text was eliminated from the text language identification if it was less than 40 bytes after information extraction.

From data training results in Table 4-4, we were able to obtain the priority order according the correct rate of different language identification approaches. The encoding declaration had the best identification correct rate; language declaration ranked the second and text identification ranked last.

Item	Language Declaration	Encoding Declaration	Text identification	Text identification
Identified Number	3003	3768	22791	19803
Coverage Rate (%)	10.01%	12.56%	75.97%	66.01%
Correct rate (%) (Exclude unknown)	99.8%	100%	98.6%	98. 8%
Priority List	**2**	**1**	**3**	

Table 4-4 Language identification training results

4.4. Database Optimization

Four approaches were employed to improve the database performance and the communication between the client and database server, which are:

1. Added the proper indexes to database tables for increasing the database query speed. For the Web census giant database, a proper index is very important. Our experience shows a proper index can speed up the database query by over 20 times.
2. Analyzed data from different perspectives and summarizing it into useful information: For example, we added a summary table for geographical distribution where the numbers of Web servers were added for each country. This

was very useful for sorting the amount of Web servers by their country and reducing the repeated query time.

3. Fetched small amounts of data iteratively instead of fetching all the data at once: An ID attribute which was increased by one was added to the huge data table. Based on this feature, we were able to use a stored procedure to return a certain amount of data iteratively from a huge data table without keeping all the data in the cache (in-memory) thus, we avoided memory overflow when data was retrieved from the database.
4. Improved the performance of communication between client and database server: For example, a connection pool was added to arrange and recycle data connections. Moreover, several approaches, such as using batch operation, releasing the resource when finished and setting the proper direction for processing rows, were used to improve the performance of JDBC.

5. Results

We categorized the results into four fields: language identification experiment, Web geographical distribution, general Web language distribution and language distribution in a specific geographical location.

5.1. Language Identification Experiment

5.1.1. Accuracy

For our experimental scenario, three Web language identification approaches, language declaration, character encoding declaration and Web common text were employed. For Web common text identification, 83 language category files that cover 66 different languages were used and the category files (language models) contain the top 400 ranked N-grams. The textual information was extracted from portal pages of Web servers. For language declaration language identification, 1001 language codes (ISO 639-2 Code) were prepared for determining the language declaration codes extracted from the metadata and HTML tags of portal pages. For the character encoding declaration, 72 different character encoding codes (each of them recognizes a single language) were collected for classifying the character encoding codes extracted from the metadata of portal pages.

We verified the accuracy of our language identification results by randomly picking samples from seven languages which include some Asian and Middle Eastern languages

(Chinese, Japanese, Korean and Turkish) and Western languages (English, German and French). The total number of portal pages in the collections was 1400, with 200 random pages for each language. For example, we randomly picked 200 portal pages that had been identified in English by our language identification system and then we manually verified each of these pages. The results are shown in Table 5-1. It should be noted that the unknown portion was excluded from our experimental results.

System Identified Results / Verified Results	English	Chinese	Japanese	German	Turkish	French	Korean
English	200	0	0	0	0	1	0
Chinese	0	199	0	0	0	0	0
Japanese	0	1	200	0	1	0	0
German	0	0	0	199	0	0	0
Turkish	0	0	0	0	199	0	0
French	0	0	0	1	0	198	0
Korean	0	0	0	0	0	0	200
Danish	0	0	0	0	0	1	0
#Correct	200	199	200	199	199	198	200
%Correct	100.00%	99.50%	100.00%	99.50%	99.50%	99.00%	100.00%

Table 5-1 Results for the language identification

We looked into the pages that were misidentified. They have the same common characteristics: the pages were short and have no information regarding language or character encoding declaration. One Japanese portal page was identified as Chinese because it had more Chinese characters than syllables (Japanese writing incorporates both Chinese characters and symbols that have a sound value like an alphabet. This is called a syllabary), so the text language identification considered the primary language of this page as Chinese.

Comparing the training results based on each language identification approach in Table 4-4, the language identification results based on the combination of three types of language identification approaches were remarkable. Each approach has its strengths and weaknesses but with three types of language identification approaches combined together, it maximized strengths and complements other detection approaches.

The experimental results demonstrated good performance in a realistic setting. We achieved an average of 99.6 percent accuracy rate for seven identified languages on web pages over 40 bytes. Among seven languages, the worst language identification rate was 99 percent.

5.1.2. Coverage

For language identification coverage, the number of all identified portal pages increased when all three approaches were used together because each of the approach has its own strengths for different languages. The coverage rate comparison for overall approaches and each language identification approach based on all our language identification results are represented in Figure 5-1.

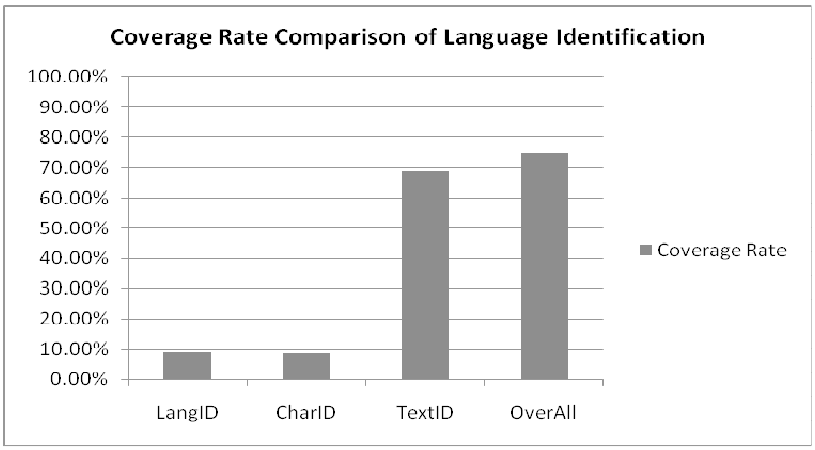


Figure 5-1 Coverage rate comparison of language identification

The coverage of character encoding declaration language identification on all the portal pages was around 10 percent because most of the character encoding declaration cannot uniquely determine a single language. For example, “ISO-8859-1” refers to the "Latin alphabet" which covers over 20 languages. No language can be identified when this character encoding declaration was retrieved from portal pages, yet character encoding declaration language identification performed very well on Chinese, Japanese, Turkish and Korean pages because a significant portion of these pages had a character encoding declaration and the encoding code can identify a single language. A comparison of character encoding declaration language identification coverage rate on portal Web pages among seven languages is shown in Table 5-2. The same data is presented in Figure 5-2.

Item	English	Chinese	Japanese	German	Turkish	French	Korean
Charset Identified Num	22448	296393	164207	0	176765	0	89020
Total Identified Num	6005084	320261	314021	196391	178435	132246	92821
Charset Identified Rate	0.37%	92.55%	52.29%	0.00%	99.06%	0.00%	95.91%

Table 5-2 Character encoding declaration coverage comparison

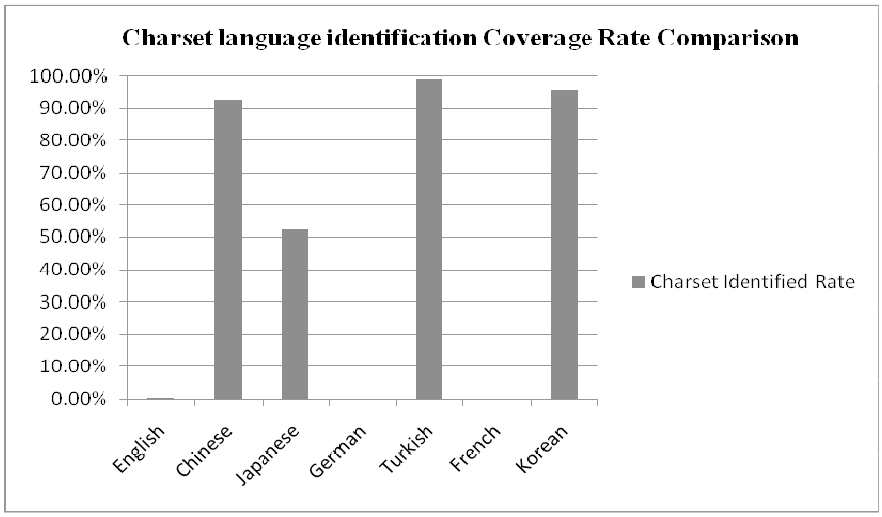


Figure 5-2 Character encoding declaration coverage rate comparison

Compared with English, German and French Web pages, the Chinese, Japanese, Turkish and Korean Web pages achieved outstanding character encoding declaration language identification coverage, especially the Turkish (99.06%), Korean (95.91%) and Chinese (92.55%) pages. The English, German and French Web pages were hardly identified by their character encoding declaration because their character encoding codes cannot indicate a single language. For example, a character encoding code, "ISO-8859-2", represents a character set that contains all the required characters for over 15 European languages which include German and English (Peterlin, 1996).

The coverage of language identification was increased when all three approaches were used together. For language recognition, we identified languages for 74.8 percent of portal Web pages (10,393,461 pages) that were retrieved from the Internet with the HTTP response code of “200”.

5.1.3. Unknown Languages

If the three types of Web language identification approaches failed to determine the language of a portal page, it was assigned an Unknown language label (25 percent out of the total identified portal pages). From our investigation, a large portion of "Unknown language" identifications were caused for five reasons:

1. The limitation of Web common text size: To ensure identification accuracy, the Web pages were eliminated from Web common text identification if their textual text was less than 40 bytes. Among all the "unknown languages" Web pages, 22 percent of Web pages were not eligible for text identification because of their textual text's size limitation.
2. The limitation of category files (language models) coverage: We were able to identify 66 languages using 83 language category files. Beyond the coverage of category files, the language of the text was not identifiable.
3. The limitation of language and character encoding declaration language identification: The coverage rate comparison of language identification in Figure 5-1 shows the coverage of language and character encoding declaration language identification was around 10 percent. The remaining portions had to depend on Web common text language identification.
4. Parts of the portal pages were improperly received: Some of the Damn Small Linux machines that were used for Web census processing had no International Language

support packages. This caused some language characters in portal pages to be improperly collected and replaced by question marks. These pages could not be used for text language identification.

5. The primary language was not strong under the noisy Web environment: If short Web common text contained a big portion of foreign words, the text language identification failed to output an assured result.

Languages of 25 percent of the portal pages cannot be determined because of the natural characters of Web pages and the limitation of language identification coverage. If given more time and resources, we may reduce the portion of unknown languages.

5.2. Web Geographical Distribution Results

5.2.1. Overall Results

A global geographic view of the results is shown in Figure 5-3. The US was ranked at first place with over 10,000,000 Web servers and China was ranked at second place with over 1,000,000 web servers. In Figure 5-3, six patterns were employed to depict global distribution of Web servers by their number.

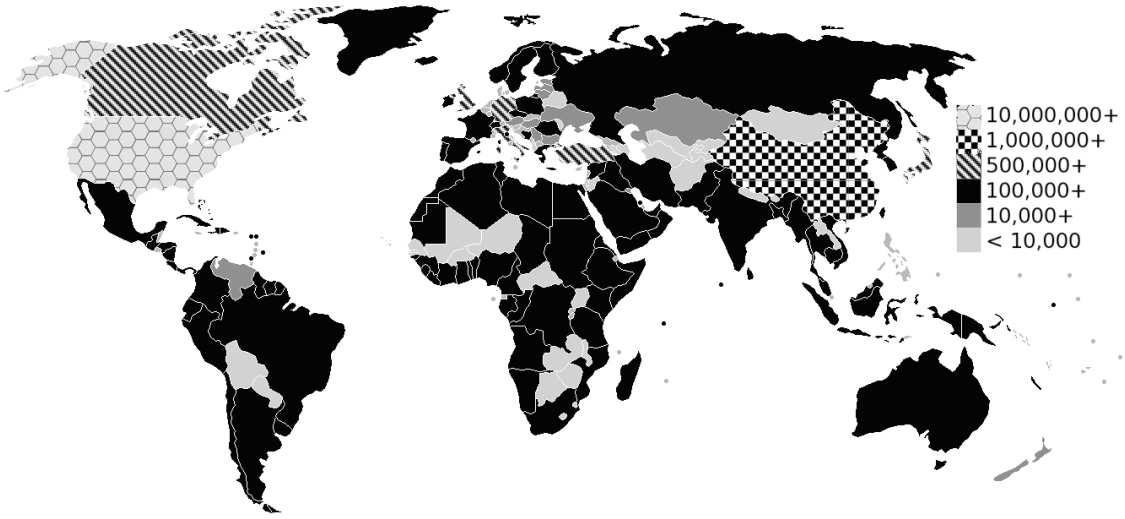


Figure 5-3 Global view of Web servers by country

According to the statistical survey report on the Internet development in China (July 2008 version) conducted by China Internet Network Information Center (CINIC, 2008), the US owns 56.9 percent the IPv4 addresses in the world, so it is not surprising that the US has 49 percent of the Web servers.

5.2.2. Top 20 Countries

According to Web geographical distribution results, the top 20 countries ranked by Web server number and their coverage rate out of 24,234,406 Web servers are listed in Table 5-3. As mentioned, the top position was the US and China ranked second. We should know that in MaxMind GeoLite Country database, Web servers hosted by AOL IP addresses were classified as being in the US. The CINIC (2008) also shows that by the end of June, 2008, the number of Internet users in China had reached 253 million and has leaped to first place, above the US. Beside the factor of IP address coverage previously

mentioned, a huge network population in the US and China should affect the number of web servers in these two countries.

Order	Rate out of Total (%)	Number of Servers	Country and Region Name
1	48.92	11854755	United States
2	4.21	1019734	China
3	3.98	963672	Japan
4	3.53	856157	Germany
5	3.17	769096	Canada
6	2.90	703805	United Kingdom
7	2.20	533737	Turkey
8	2.16	524116	Italy
9	1.89	458255	Brazil
10	1.61	389371	South Korea
11	1.52	369226	Australia
12	1.51	366120	France
13	1.49	361115	Netherlands
14	1.46	354194	Spain
15	1.39	337814	Taiwan
16	1.18	286849	Poland
17	1.13	274241	India
18	1.13	273257	Russian Federation
19	0.84	202654	Thailand
20	0.76	184071	Israel

Table 5-3 Top 20 countries by Web server count

One of the interesting entries in Table 5-3 is Turkey in seventh position. This country is usually not considered as an international Internet leader. Turkey is a country that stretches across the border between Asia and Europe and borders on eight countries. It is possible that some organizations in other countries, especially within Middle East countries, set their Web servers in Turkey.

5.3. General Web Language Distribution Results

5.3.1. Top 20 Languages

The top 20 languages ranked by the Web servers' portal pages count and their coverage rate out of 10,393,461 portal pages (considering only those portal pages with "200" HTTP response code) are listed in Table 5-4. In total, around 43 percent of Web servers' portal pages were involved in language identification. The same data is shown in Figure 5-4. We pulled the English and Unknown out of the Figure 5-4 to avoid details of other languages being overshadowed.

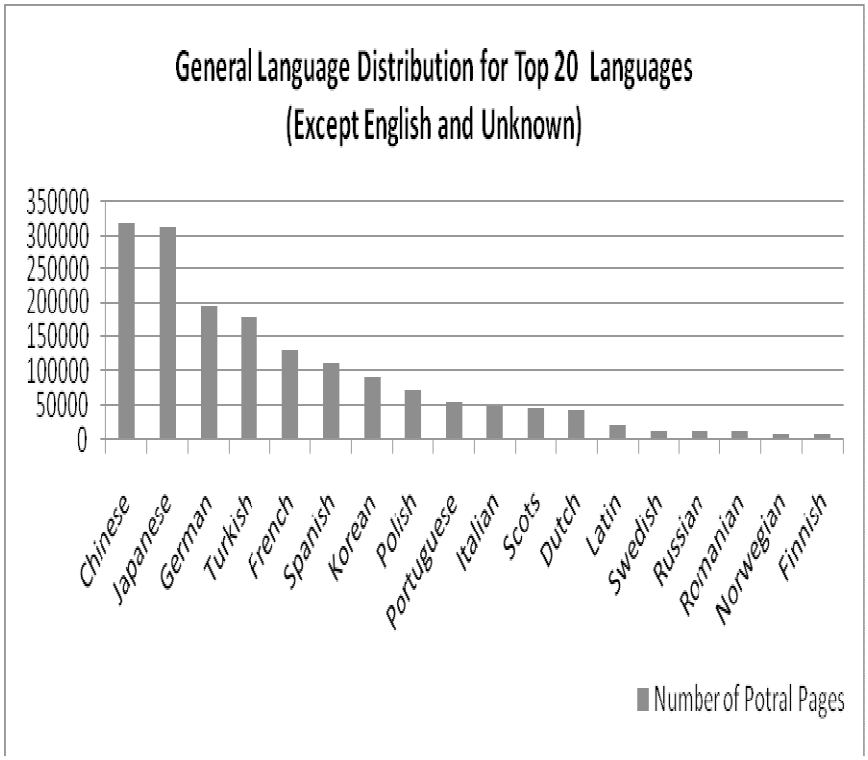


Figure 5-4 General language distribution for top 20 languages

Order	Rate out of Total (%)	Number of Web Pages	Language Name
1	57.78	6005084	English
2	25.18	2617272	Unknown
3	3.08	320261	Chinese
4	3.02	314021	Japanese
5	1.89	196391	German
6	1.72	178435	Turkish
7	1.27	132246	French
8	1.09	112975	Spanish
9	0.89	92821	Korean
10	0.70	72875	Polish
11	0.53	54941	Portuguese
12	0.48	50211	Italian
13	0.47	48616	Scots
14	0.43	44901	Dutch
15	0.23	23596	Latin
16	0.14	15004	Swedish
17	0.14	14394	Russian
18	0.14	14367	Romanian
19	0.08	8544	Norwegian
20	0.08	8316	Finnish

Table 5-4 Top 20 languages by Web servers' portal pages count

The top 20 general language distribution ideally matches the rank of the top 20 countries by Web server count in Table 5-3. English ranked in the first position with 57.78 percent because among the top 20 countries by Web server count , the United States (position 1), Canada (position 5), United Kingdom (position 6), Australia (position 11) and India (position 17), have English as one of their official languages. The languages of Chinese, Japanese and German were matching in their ranks of Web server number as well. It is reasonable that the rank of French (position 7) and Spanish (position 8) are

enhanced compared to the ranks of Web server numbers in France (position 12) and Spain (position 14) since these two languages are widely used in other countries. According to Central Intelligence Agency (2008), in the world there are 29 countries where French is an official language and 20 countries where Spanish is an official language.

5.4. Language Distribution in a specific Geographical Location

Two investigations are carried out for language distribution in specific geographical locations. They are language distributions in Canada and Chinese Web pages Geographical Distribution.

5.4.1. Language Distribution in Canada

Canada has diverse multilingual populations and a multitude of spoken languages. English and French are recognized by the Constitution of Canada as official languages. According to Statistics Canada (2006), the five most widely-spoken non-official languages are Chinese (the portal Web language of 2.6% of Canadians), Punjabi (0.8%), Spanish (0.7%), Italian (0.6%), and Arabic (0.5%). Our language distribution results in Canada based on 366,780 portal Web pages shows that except for the unknown portion, the top 2 languages on portal Web pages are English and French. Our top 10 languages by Web servers' portal pages count are shown in Table 5-5. The same data is plotted in Figure 5-5 (Note that English, Unknown and French are omitted to preserve detail). Besides the official languages, the languages such as Spanish, Chinese, Latin (an Italic

language) on the top 10 languages by Web servers' portal pages count are getting close to the distribution of widely-spoken non-official languages in Canada.

We were able to identify 66 languages on the portal Web pages in Canada. This result shows that the Web pages in Canada have a wide range of language distribution and matches the spoken languages status in Canada.

Order	Rate out of Total (%)	Number of Web Pages	Language
1	76.49%	280542	English
2	15.16%	55610	Unknown
3	5.69%	20870	French
4	0.56%	2038	Scots
5	0.38%	1406	Turkish
6	0.33%	1196	Spanish
7	0.26%	968	Chinese
8	0.20%	748	Latin
9	0.13%	466	German
10	0.08%	311	Breton

Table 5-5 Canada's top 10 languages by Web servers' portal pages count

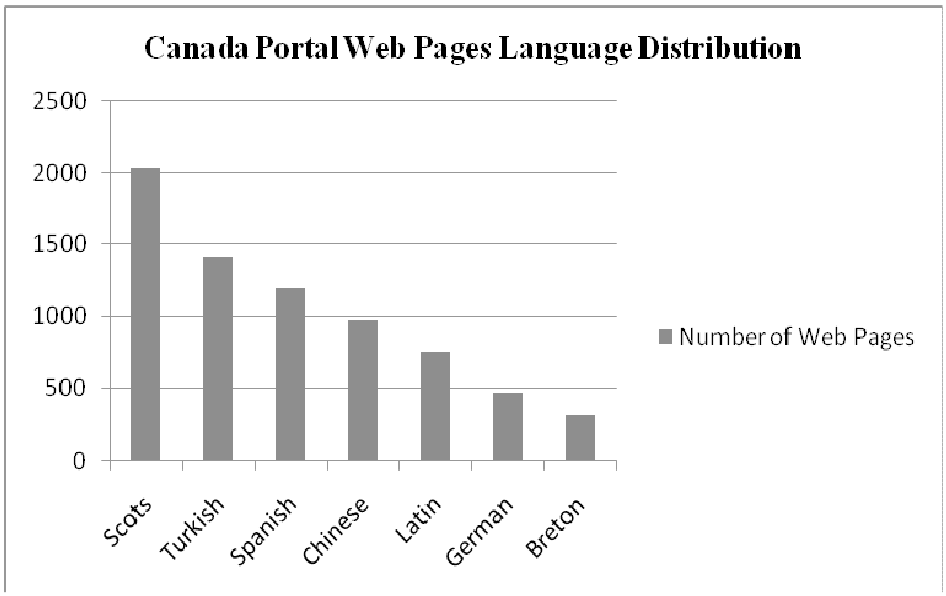


Figure 5-5 Canada Portal Web Pages Language Distribution

5.4.2. Chinese Geographical Distribution

320,261 Chinese Web pages were identified by our language identification approaches. Chinese Web pages geographical distribution result shows that the Chinese Web pages were mostly located in mainland China, Taiwan, United States, Hong Kong and the Russian Federation. The top 10 countries by Chinese portal pages count are listed in Table 5-6.

Country and Region	Rate out of All (%)	Number of Web pages
China	64.02%	205046
Taiwan	22.65%	72546
United States	3.71%	11872
Hong Kong	3.27%	10486
Russian Federation	3.05%	9778
Korea	0.36%	1158
Israel	0.34%	1098
Canada	0.30%	968
Italy	0.21%	681
Japan	0.18%	561

Table 5-6 Top 10 countries by Chinese portal pages count

There are currently two systems for Chinese characters, the traditional system and the simplified Chinese character system. The traditional system is still used in Hong Kong, Taiwan, Macau and Chinese speaking communities outside mainland China. The simplified Chinese character system, developed by the People's Republic of China in 1954 to promote mass literacy, simplifies most complex traditional glyphs to fewer strokes. The geographical distribution of Chinese portal Web pages reflects the

distribution of the Chinese population. It also proved that our language identification approaches are able to identify two systems of Chinese characters. Following cultural and commercial exchange and emigration, some countries such as the United States and the Russian Federation have a certain number of Chinese Web pages, as well.

6. Conclusions and Future Work

This thesis presents a Web language identification system based on three types of Web language identification approaches. We discuss characteristics and coverage of each approach and combine them to maximize their strengths and complement other detection approaches. According to the special characteristics of the Web pages, the Web information extraction method was augmented to better handle the Web language identification within a noisy Web document environment. This system was trained and deployed on the portal Web pages of 24.2 million Web servers and provided a satisfying result for global portal Web pages language and geographical distribution. The results also help to further Web categorization since Web investigation must be customized corresponding with the conventions of different languages and cultures.

6.1. Summary of Contributions

6.1.1. A Composite Approach to Web Language Identification

A composite approach constituted by a Web information extraction method and three types of Web language identification methods: language declaration, character encoding declaration, and text language identification was used to improve the performance of Web language identification. The three types of languages identification methods were not original but using them together for language identification is an innovation. Our sampling result based on seven languages in Table 5-1 shows the average correct rate is

very high. It achieved a 99.64 percent (Note unknown is excluded) accuracy rate. Each method contributed its merits for language identification. They include:

1. A precise Web information extraction method dealt with a large variety and diversity of Web pages and extracted usable information for three different language identification methods.
2. Language declaration and character declaration language identification complemented the text language identification. They had less coverage but if Web pages were identified by these two methods, the correct rate was quite high (Language declaration: 99.8 percent and character declaration: 100 percent). The character declaration method achieved outstanding coverage on some languages such as Turkish and Chinese.
3. An improved N-gram text language identification overcame tokenization problems in the regular N-gram approach (Asian languages identification became available) and increased accuracy by adding multiple languages models that were encoded by different encoding schema for a language.

6.1.2. Web Language and Geographical Distribution

It is not a trivial task to determine Web language and geographical distribution on the Internet. We contributed Web language and geographical distribution results by using the composite language identification approach, geographical distribution approach and other comprehensive supports to glue these two parts together. More specifically, the

distribution results include:

1. Web servers' geographical distribution on 24.2 million found Web servers
2. Language identification on the portal pages of 24.2 million found Web servers
3. The combination of Language and geographical distribution

6.2. Future Work

Although this system has already demonstrated good performance in a realistic setting, there is considerable room for further improvements. We propose some ideas and solutions to improve the accuracy and coverage of Web language identification and enhance the investigation on the results of language and geographical distribution.

We would like to explore principal approaches to better handle language identification on Web pages with multiple languages; to raise the performance of other Web classifications by using Web language identification system and its categorization results; and to support Web page translation by employing Web language identification technology (Currently Google Web page translations required selecting the source language of the Web page before translation).

Since we recorded different portions (Metadata, title, textual text, text size) of Web pages using the Web information extraction method, additional statistical research such as metadata usage and Web text size could be studied.

6.2.1. Accuracy and Coverage

In order to improve the accuracy and coverage of Web language identification, we

propose ideas from three scopes. They are language identification on text, language declaration and character encoding declaration.

For text language identification, several approaches could be used. They include:

1. Enlarge the coverage of language models to support more languages and encoding schemas. Since the code for creating new language models is available, more language models could be added to our system.
2. Train and adjust the best edge distance (output unknown if average distance bigger than edge distance) for each N-gram language model. Currently, we use a single edge distance for all the language models.
3. Use linkage information and the text from hypertext anchors such as alternative (Alt) text could also provide improvements on the overall results.
4. A better tuning of the N-gram language models used to classify very small text samples (i.e.: smaller than 40 bytes).

For character encoding declaration language identification, the encoding code that cannot uniquely identify a single language could be used to narrow down the text language identification range. It will also speed up the process of text language identification in the long run. For example, the character encoding, “KOI8-R,” covers the two languages Russian and Bulgarian. Once this character encoding declaration is detected on Web pages, the language models in only these two languages should be loaded for text language detection.

For language encoding declaration language identification, since there is no uniform

way of specifying the language declaration codes on Web pages (for example: English can be specified as en, en-US, eng, English, etc.), more investigation should be carried out on cases in which language encoding declaration codes have been extracted from Web pages but cannot be identified to any languages.

6.2.2. Investigation on Language and Geographical Distribution

More investigation on our language and geographical distribution should be carried based on our results. For instance, investigation could include:

1. The Web servers' size and growth
2. The trend of Web language geographical distribution on the Internet
3. The language coverage of portal Web pages in some specific countries, etc.
4. The fastest growing languages.

Appendix A Glossary

Term	Abbreviation	Definition
Document Object Model	DOM	Document Object Model is a programming interface specification being developed by the World Wide Web Consortium (W3C). It lets a programmer create and modify HTML pages and XML documents as full-fledged program objects.
Hypertext Markup Language	HTML	A mark-up language designed for the creation of Web pages with hypertext and other information to be displayed in a Web browser.
Hypertext Transfer Protocol	HTTP	Hypertext Transfer Protocol is the set of rules for transferring files on the World Wide Web.
International Organization for Standardization	ISO	An organization established to promote the development of standards to facilitate the international exchange of goods and services, and to develop mutual cooperation in areas of intellectual, scientific, technological, and economic activity.
Information Extraction	IE	Information extraction is a particularly useful sub-area of natural language processing (NLP), and its goal is to locate specific information from natural language documents.
	JDBC	A standard used by Java programs for connecting to various databases using the same interface.
Message Understanding Conferences	MUC	The Message Understanding Conferences were initiated and financed by Defense Advanced Research Projects Agency (DARPA) to encourage the development of new and better methods of information extraction.
Natural Language Processing	NLP	Natural language processing is a sub-field of artificial intelligence and computational linguistics. It studies the problems of automated generation and understanding of natural human languages.
Online Computer Library Center	OCLC	Founded in 1967, Online Computer Library Center is a nonprofit, membership-based, computer library service and research organization dedicated to public purposes for

		furthering access to the world's information and reducing the rising rate of library costs.
Simple API for XML	SAX	Simple API for XML is an event-based interface for processing XML documents.
Extensible HyperText Markup Language	XHTML	Extensible Hypertext Mark-up Language is a hybrid of XML and HTML. Web pages designed in XHTML should look the same across all platforms.
Extensible Markup Language	XML	Extensible Markup Language is a standard for creating markup languages which describe the structure of data.
Web Language Identification and Distribution System	WLIDS	A Web language identification and distribution system used for Web Census System that is conducted by the Jodrey School of Computer Science, Acadia University.
World Wide Web	WWW	World Wide Web or simply Web, a subset of the Internet which uses a combination of text, graphics, audio and video (multimedia) to provide information on most every subject imaginable.

Bibliography

Abou-Assaleh, T., Cercone, N., Keselj, V., Sweidan, R. (2004). N-gram-based Detection of New Malicious Code. *Proceedings of the 28th Annual International. Volume 2, 41- 42.*

Apache. (2007). Apache Commons Logging. Retrieved March 20, 2008, from http://commons.apache.org/logging/.

Benoit, D., Slauenwhite, D. & Trudel, A. (2006). A Web Census is Possible. *Proceedings of the 2006 International Symposium on Applications and the Internet (SAINT2006), Phoenix, Arizona, 23-27.*

Benoit, D., Slauenwhite, D., & Trudel, A. (2007). On the path to a World Wide Web census: A large scale survey. *Proceedings of the 2nd International Conference on Internet Technologies and Applications (ITA 07), (Sept 2007), Wrexham, UK, 3, 378-389.*

Bruno, M. & Mario, J. S. (2005). Language identification in Web pages. *Proceedings of the 2005 ACM Symposium on Applied Computing, Santa Fe, New Mexico, 764 – 768.*

Cavnar, W. B. & Trenkle, J. M. (1994). N-Gram-Based Test Categorization. *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, 161-175.*

Statistics Canada. (2006). Profile of Language, Immigration, Citizenship, Mobility and Migration for Canada, Provinces, Territories and Federal Electoral Districts (2003 Representation Order). Ottawa, 2007, 6-10.

Chinchor, N. (1992). MUC-4 Evaluation Metrics. *Proceedings of the Fourth Message Understanding Conference. McLean, Virginia, USA, 22-29.*

China Internet Network Information Center (CINIC). (2008). Statistical Survey Report on the Internet Development in China. Retrieved August 17, 2008, from http://www.cnnic.cn/download/2008/CNNIC22threport-en.pdf.

Central Intelligence Agency (CIA). (2008). The 2008 World Factbook. Retrieved August 17, 2008, from https://www.cia.gov/library/publications/the-world-factbook/.

David, W. (1992). Relating Relational Learning Algorithms. Retrieved Feb 29, 2008, from http://citeseer.ist.psu.edu/aha92relating.HTML.

Deixto. (2008). Web Content Extraction Make Easy. Retrieved May 29, 2008, from http://deixto.csd.auth.gr/index.php.

Dunning, T. (1994). Statistical Identification of Language. *New Mexico, United States: Computing Research Laboratory, New Mexico State University, Technical report, 94-273.*

Douglas, E. A. & David, I. (1999). Introduction to Information Extraction Technology. *IJCAI-99 Tutorial, Stockholm, Sweden.*

Edward, O., Patrick, M., & Brian, L. (1997). A Methodology for Sampling the World Wide Web. Retrieved March 8, 2008, http://www.oclc.org/research/publications/arr/1997/oneill/o%27neillar980213.htm.

GEOBYTES. (2006). IP Address Locator Tool. Retrieved March 12, 2008, http://www.geobytes.com/IpLocator.htm.

Grefenstette, G. (1995). Comparing Two Language Identification Schemes. *Proceedings of 3rd International Conference on Computational Linguistics, Volume 2, Copenhagen, Denmark, .652-657.*

HOSTIP.INFO. (2007). Domain to IP or Host name lookup. Retrieved March 1, 2008, http://www.hostip.info/.

HTML Parser. (2006). HTML Parser. Retrieved Feb 29, 2008, from http://HTMLparser.sourceforge.net/license.HTML.

Kapow. (2008). Harvesting Web Intelligence. Retrieved May 19, 2008, http://www.kapowtech.com/solutions/solutions_Webintelligencecollection.aspx.

Kikui, G. (1996). Identifying, the coding system and language, of on-line documents on the Internet. *Proceedings of 16th International Conference on Statistical Analysis of Textual Data, Rome, Italy, 2, 652-657.*

Line, E. (1999). Information Extraction from World Wide Web A Survey. Retrieved Feb 19, 2008, from http://citeseer.ist.psu.edu/eikvil99information.HTML.

Library of Congress. (2006). ISO639-2. Retrieved Feb 21, 2008, http://www.loc.gov/standards/iso639-2/.

MaxMind GeoLite Country Database. (2008). MaxMind GeoLite Country Database. Retrieved March 8, 2008, http://www.maxmind.com/app/country.

Micha, C., Yungming, S. & Weiyi, M. (1997). Using the Structure of HTML Documents to Improve Retrieval. *Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems, Monterey, California, 22-22.*

Microsoft. (2008). Character Set Recognition. Retrieved Feb 29, 2008, from http://msdn.microsoft.com/en-us/library/aa752010.aspx.

Ning, Z., Hong, C., Yu, W., ShiJun, C., MingFeng, X. Odaies: Ontology-driven Adaptive Web Information Extraction System. *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), 13 (16), 454 – 460.*

Online Computer Library Center. (2002). Web Characterization Project. Retrieved March 7, 2008, from http://wcp.oclc.org.

Peterlin, P (1996). ISO 8859-2 Character Set. Retrieved August 40, 2008, from http://nl.ijs.si/gnusl/cee/charset.html.

Proxool. (2007). Proxool 0.9.0RC3. Retrieved March 7, 2008, from http://proxool.sourceforge.net/.

ShanJian, L. & Katsuhiko, M. (2001). A composite approach to language/encoding detection. Technical report, Netscape Communications Corp, Retrieved March 2, 2008, from http://www.mozilla.org/projects/intl/UniversalCharsetDetection.html.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. Machine Learning*, 34, 233-277.*

Steve, F. (2001). XML Parsers: DOM and SAX Put to the Test. Retrieved March 4, 2008, from http://www.devx.com/xml/Article/16922/0/page/1.

Steve, L. & Lee, L, G (1999). Accessibility of information on the Web. *Nature, Volume 2. 400, pp. 107- 109.*

Textcat. (1997). Textcat Language Guesser. Retrieved May 29, 2008, from http://odur.let.rug.nl/~vannoord/TextCat/.

Tomas, O. (2005). N-Gram Based Statistics Aimed at Language Identification. *Proceedings of 1st Informatics and Information Technologies Student Research Conference, Bratislava, Slovakia, 1-7.*

Unit Mine. (2008). Unit Miner Web data extraction. Retrieved May 3, 2008, from http://www.qualityunit.com/unitminer/.

Vlado, K., Fuchuan, P., Nick, C., Calvin, T. (2003). N-gram-based Author Profiles For Authorship Attribution. *Proceedings Of The Conference Pacific Association For Computational Linguistics, Pacling'03, Dalhousie University, Halifax, Nova Scotia, Canada, 255-256.*

World Wide Web Consortium. (2002). Internationalization. Retrieved March 3, 2008, from http://www.w3.org/International/O-charset-lang.HTML.

World Wide Web Consortium. (2004). Authoring Techniques for XHTML & HTML Internationalization: Characters and Encodings 1.0. Retrieved March 3, 2008, from http://www.w3.org/International/geo/HTML-tech/tech-character.HTML.

World Wide Web Consortium. (2007a). Internationalization Best Practices: Specifying Language in XHTML & HTML Content, from http://www.w3.org/TR/i18n-HTML-tech-lang/#ri20030510.102829377.

World Wide Web Consortium. (2007b). Tutorial: Declaring Language in XHTML and HTML. Retrieved March 3, 2008, from http://www.w3.org/International/tutorials/language-decl/#Slide0140.

Web-Harvest. (2006). Web-Harvest, Web extraction tool released. Retrieved March 3, 2008, from http://www.theserverside.com/news/thread.tss?thread_id=42021.