Multilingual Search engine: Implementation using UNL

Jalindar Baban Karande
Department of computer engineering,
Vishwakarma Institute of Technology,
Pune.
jalindar_karande@yahoo.co.in

Abstract. In this paper, I am proposing an implementation of multilingual search engine using a Universal Networking Language (UNL). As a local market for web is increasing there are increasing numbers of pages in native language of customers. So it becomes very important to search a information which is scattered in different languages and present to the users of different languages. In my approach UNL system, which is used for cross language access of information, is used by spider in search engine to find out pages in different native languages and converted into UNL before creating a indexing. Once index are created retrieval is same as monolingual retrieval.

1 Introduction

Before 90’s there is only one language used for representing content on web i.e. English. But now a day web pages in native languages are going to increase and nearly half of web contents are now in other than English language. In recent study (Fredric, 2004) English language is at third rank according to number of people speaking where as Chinese at top and Hindi at fifth rank. So we can’t neglect information present in web content other than English languages. Table 1 shows a distribution of web content in top ten languages used for web.

This Internet multilinguality creates the most significant challenges regarding to the possibility to find out materials in distinct languages. Multilinguality becomes a serious issue for search engines.

Following are Requirements of Internet search engine concerning to multilingual stuff:

- Possibility to search materials in different languages;
- Possibility to retrieve documents in defined language only;
- Possibility for the user to choose interface in desirable language;
- Ability to work correctly with multi coding languages;
- Possibility to translate query;
• Possibility to translate search results;
• Possibility to translate documents itself.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Language</th>
<th>Percent of web</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>English</td>
<td>57.37</td>
</tr>
<tr>
<td>2</td>
<td>German</td>
<td>8.72</td>
</tr>
<tr>
<td>3</td>
<td>French</td>
<td>2.88</td>
</tr>
<tr>
<td>4</td>
<td>Russian</td>
<td>2.49</td>
</tr>
<tr>
<td>5</td>
<td>Japanese</td>
<td>2.49</td>
</tr>
<tr>
<td>6</td>
<td>Chinese</td>
<td>2.50</td>
</tr>
<tr>
<td>7</td>
<td>Spanish</td>
<td>2.46</td>
</tr>
<tr>
<td>8</td>
<td>Italian</td>
<td>2.21</td>
</tr>
<tr>
<td>9</td>
<td>Korean</td>
<td>1.92</td>
</tr>
<tr>
<td>10</td>
<td>Dutch</td>
<td>1.90</td>
</tr>
</tbody>
</table>

Table 1 Percentage distribution of web content. (Fredric, 2004)

2 Traditional approaches to multilingual retrieval

The most traditional approach multilingual retrieval is a use of controlled vocabulary for indexing and retrieval. In this approach, a computer program is used to select a few keywords taken from a closed list of authorized terms for each document. Semantic relations can be used to help choose the right descriptors, and solve the sense problems of synonyms and homographs. The list of authorized terms and semantic relations between them are contained in a vocabulary.

To implement multilingual querying using this approach, it is necessary to give the corresponding translation of each term in vocabulary for each new language recognized. This work is facilitated by the fact that each keyword is chosen in order to express a precise unambiguous concept.

A problem remains, however, since concepts expressed by one single term in one language sometime are expressed by distinct terms in another. For example, the common language term *mouton* in French is distinguished into two different concepts in English, *mutton* and *sheep*. One solution to this problem, given that these distinctions are known
between the languages implemented is to create pseudo-words such as mouton (alimentation) ---mutton and mouton (animal) ---sheep. These domain semantic tags (such as animal and alimentation) as well as the choice of transfer terms depend on the final use of the multilingual vocabulary, and it is therefore sometimes easier to build a multilingual vocabulary from scratch rather than to adapt a monolingual one.

This controlled vocabulary approach gives acceptable results but prohibits precise queries that cannot be expressed with these authorized keywords. It is however a common approach in well-delimited fields for which multilingual vocabulary already exist (legal domain, energy, etc.) as well as in multinational organizations or countries like India with several official languages, which contain lexicographical unit familiar with problems of terminological translation.

3 Introduction of the UNL System

The UNL system basically consists of UNL Language Servers, UNL Editors and UNL Viewers. The Universal Networking Language consists of the UNL Relations, the UNL Attributes, the Universal Words and the UNL Knowledge Base (Uchida & Zhu, 2005).

Any one having access can convert pages from his/her native language to the UNL by using Enconverter and once pages is in UNL they can be converted into any language by using Deconverter. So this UNL act as an Intermediate language for conversion between native languages. E.g. Chinese can be converted into UNL and now user from any language like Hindi or English can read that page by converting UNL into their native language. Figure 1 shows a basic block diagram of UNL system. UNL system consists of following parts:

1. UNL Viewer
2. UNL editor
3. Language Servers
We will see working of UNL with example of how one page developed in Hindi is displayed in English. Whenever author of web page develops web page in Hindi he/she and sends this page to the Hindi language server through UNL editor. Each language server is associated with only one language and having facility to convert this language into UNL using Enconverter facility and convert UNL into his own language using Deconverter facility. In this example Hindi language server convert Hindi content into UNL and sends back to UNL editor. Now this UNL contents are published instead of Hindi contents.

Now when English language user wants to view this page, he/she has to use UNL viewer. UNL viewer reads this UNL content from web server and sends this content to English language server. English language server converts this UNL into English language and sends to UNL viewer, which is responsible for displaying content to the User.

4 Proposed Implementation of multilingual Search engine Using UNL

Figure 2 shows a basic block diagram of search engine. To implement multilingual search engine using UNL we requires to convert content form any language to UNL before building list of keywords or index terms. Every index term is now in UNL, Each ongoing operation is also remains as it is.
Figure 2 Basic block diagram of search engine. (From http://static.howstuffworks.com/gif/search-engine-chart.gif)

Figure 3 Proposed Modification is Basic block diagram of search engine.

In Above figure 3 Spider sends pages in native language to the converter, which is responsible for converting this native language pages into UNL. Converter first finds out a language of web pages and
then sends this page to that language server. Language server translates this native language page into UNL. Now this UNL page is available for further operations in search engine same as normal page.

At the time of searching information again query is given to converter, who converts this query into UNL. All searching operations are performed on UNL, because query as well as index terms are present in UNL form. Finally Result is to be converted into native language of Query.

**Conclusion**

Implementation of Multilingual search engine is quite simple, because of use of UNL. User from any language can write a query in his/her native language and able to retrieve result from other languages also. Contents from other languages are available to user in his/her own native language again through UNL. So this technique really finishes the information exchange problem between users of different languages.

**References**


