Domain Specific Information Retrieval in Multilingual Environment

S.M. Chaware¹, Srikantha Rao²

¹Department of Information Technology, D.J. Sanghvi College of Engineering, Mumbai-56, India
Email: smchaware@gmail.com

²Department of Computer, SVKM’s Mukesh Patel School of Technology Management and Engineering, NMIMS University, Mumbai-56, India
Email: dr_s_rao@yahoo.com

Abstract—In today’s world of globalization, local language storage and retrieval is essential for the developing nations like India. As our country is diversified by languages and only 10% of population is aware of English language, this diversity of languages is becoming barrier to understand and acquainted in digital world. It has been found that when services are provided in local languages, it has been strongly accepted and used.

We proposed a new methodology called ‘Information Retrieval in Multilingual Environment’ that provides the functionality of processing and retrieval of Indian languages by entering through a keyword as input. We proposed a way of processing multilingual information where backend will be English database and front-end uses local languages like Hindi, Marathi or Gujarathi. Our system provides an interface to enter a keyword in local language, the keyword is parsed, query is formed and the result is displayed in desired local language. The result is tested by taking the Shopping Mall as a domain, where there is necessary of multilingual database processing.

Index Terms—multilingual data processing, multilingual environment, phonetic matching, shopping mall

I. INTRODUCTION

The rapidly accelerating trend of globalization of businesses and the success of e-Governance solutions require data to be stored and manipulated in many different natural languages. The primary data repository for such applications, need to be efficient with respect to multilingual data. Efficient storage and query processing of data spanning over multiple natural languages are of crucial importance in today’s globalize world.

India is a multilingual country with millions of people speaking variety of languages. Indian languages use a syllable as basic linguistic unit. The syllable writing in Indic Scripts is based on the phonetics [5]. The multilingual information can be exchanged in verbal and text mode. Thus the basic model for syllable made up of consonant phoneme, vowel phoneme [5].

There are possible ways of providing a multilingual environment for data processing. Multilingual environment is a platform which supports co-existence of multiple languages in a single database system where data interaction is not restricted by the language. It is a mechanism to enhance the user interaction irrespective of the language domain.

The goal is to provide a seamless interface to the user crossing all the language barriers. It has been found that a user is likely to stay twice as long at a site and four-times more likely to buy a product or consume a service, if the information is presented in their native language. Today English on the web is down to 35% from 90% in 1995. The fraction of Internet users that are non-native English speakers has grown from about half in mid-90’s, to about two-third and it is predicted that the majority of information available in the Internet will be multilingual by 2012.

In this paper, we proposed a better approach of storing and processing multilingual data. We proposed an efficient and easy way to convert local language keyword to English. For the inputted keyword a query will be formed, fired to the database to retrieve the data and the data is displayed in local language in a proper format. The result has been tested using three local language interface namely Hindi, Marathi and Gujarathi. We had tested using an application like ‘Shopping Mall’, where there is need of local language query processing.

II. RELATED WORKS

An attempt has been made to generate cross-lingual query with relational database using relational operators like MLineJoin and MLSemJoin, which used the phonetic matching approach by using text-to-phonetic (TTP) methodology [1][3][4]. In this approach, every string is converted into its corresponding phonetic representation by using Text-to-Phonetic algorithm. An edit distance is calculated between two strings from TTP representation for matching. If distances are between the tolerances values, they are phonetically same. But this approach is difficult for Indian languages as TTPs are not available. Some work also carried out by using MLLike operator for cross-lingual query [3]. This approach is
limited to domain and for few Indian languages. In both approaches the data is stored in multiple languages and then processed [2]. Indian Institute of Technology, Bombay has developed WordNets for Hindi and Marathi languages, and used for name and semantic matching. They had used uniform natural language (UNL) for language conversion [7].

III. SAMPLE MULTILINGUAL APPLICATION

Consider a hypothetical Shopping Mall application as shown in fig. 1. In a Shopping Mall, there may be several shops with different product brands, like Readymade Garments, Jewelries, and Music Stores etc. Some of the shops might be providing various services to the users like Internet service. The visitor should know the details of the Shopping Mall along with shops within a mall in their local language when he/she enters the Mall. The visitor will get all the details when he/she enters a Mall. In fig. 1, attributes for shopping mall shops has been shown. Apart from this, there are also databases regarding the products, services, malls, brand-details etc, which is not shown but used in our application.

IV. DESIGN & IMPLEMENTATION DETAILS

A. Design Module

The figure 2 shows the steps involved in implementation. It consists of five modules: local language interface module, query formation module, the database module, translation module and the display module. The local interface module is the user interfaces which will take input string in local language, interpret and pass to the query formation module. The query will be formed after converting string to English by using character-to-character mapping method. The database module will process the query to find the exact match from the database. Once the keyword has found, its corresponding fields are converting to selected local language by translation module. Finally, it will be displayed in proper format.

<table>
<thead>
<tr>
<th>No.</th>
<th>Flr.</th>
<th>Type</th>
<th>Disc _ ount</th>
<th>Shop _ id</th>
<th>Shop _ name</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>1</td>
<td>Sports</td>
<td>Y</td>
<td>1</td>
<td>Zenith</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>Furniture</td>
<td>N</td>
<td>2</td>
<td>Wood World</td>
</tr>
</tbody>
</table>
and corresponding character will be displayed, as shown in fig. 4. This task has been done by translation module and display module.

Fig. 4: English to Hindi conversion

The sample screen-shots of the result for the application are as shown in figure 5 and figure 6. Figure 5 is taking local language keyword as input in Hindi and figure 6 is the outcome for the same query when mentioned local languages selected. We can select any local language outcome, depending on user’s need. The result shows that the efficiency does not depend on number of phonetic primitives, but depends on minimum number of keys to be mapped to enter a local string [5].

<table>
<thead>
<tr>
<th>Equivalent English Characters</th>
<th>R</th>
<th>a</th>
<th>gh</th>
<th>oo</th>
<th>l</th>
<th>i</th>
<th>l</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalent ASCII Codes</td>
<td>2352</td>
<td>2309</td>
<td>2328</td>
<td>2370</td>
<td>2354</td>
<td>2311</td>
<td>2354</td>
<td>2310</td>
</tr>
<tr>
<td>Equivalent Hindi Characters</td>
<td>र्</td>
<td>आ</td>
<td>घ</td>
<td>ओ</td>
<td>ि</td>
<td>ल</td>
<td>ि</td>
<td>ल</td>
</tr>
</tbody>
</table>

CONCLUSIONS

The result shows that the proposed method is an efficient for retrieval of multilingual data, when search is given in local languages other than using WordNet, which may need more processing power and memory than our approach. Also we are considering all possible writing styles to match exact keyword. This way proved to be simple and effective for all possible writing styles of inputted string. For better performance, we can restrict to use this algorithm to only text-based processing applications. The future scope for the research can be extended for multilingual semantic matching using different approach.

REFERENCES

[7] www.iith.ac.in