http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 751-755, 2007
ISSN 1812-5638

Multilingual Querying and Information Processing

K. Ganesan and G. Siva
School of Computing Sciences, Vellore Institute of Technology University,
Vellore-632014, India

Abstract: Local language computing is essential for developing nations like India, as these countries are diversified by languages and this diversity of languages is the barrier for the penetration of digital revolution. We propose a new methodology called Multilingual Querying and Information Processing that provides multilingual processing functionality for some of the real world deployments like relational database systems. We propose a way of processing multilingual information wherein the back end uses English language and the front end uses local language like Tamil. The processing of the front end has the capability to display multilingual information. Our implementation is flexible enough to deploy the query processing and information retrieval systems. In this study, we present our approach to implement the above methodology.

Key words: Unicode, query-crawler, word-crawler, phonemic matching

INTRODUCTION

In today's digital world, the various technology applications, such as search engines, e-commerce portals, digital libraries, etc., must work across multiple natural languages, flawlessly. A critical requirement to achieve this goal is that, the way of displaying multilingual content and creation and maintenance of multilingual data in the database. Storing local language content in that natural language itself is a tedious task. Our proposal, Multilingual Querying and Information Processing uses the back end database in the English language and processes the English language statements into local language statements. Further, our proposed methodology is amenable for easy implementation of query processing and information retrieval systems.

Present study attempts to query and process multilingual information, say English and a local language. Here the classical Tamil language is used. For storing multilingual information and processing them, this research uses open source concepts and tools like PHP and MySQL. The significance of multilingual content is witnessed in everyday life as a survey says, currently two-thirds of Internet users are non-native English speakers (www.nua.ie) and it is predicted that the majority of web data will be multilingual by 2010 (Unicode.org).

The tools and applications available in the World Wide Web are in one of the two extremes. Either fully depends on the Global English language or the pure local language. But the content to be effectively consumed, the essence of both has to be combined. This research tries to bring-in this concept and by means of a case study, this research demonstrates that the multilingual content can be delivered in reality by means of two proposed algorithms, query-crawler and word-crawler, with minimal effort.

t09	கம்ப்யூட்டரே ஒரு கதை சொல்லு	சுஜாதா	திருமகள் நிலையம்
e27	E-FINANCE: Log into the future	V C Joshi	Sage India Pub

Fig. 1: Hypothetical bookstore database catalog

A sample multilingual application: Consider a hypothetical e-commerce application, a bookstore database that sells books across the world, with a sample catalog in multiple languages as shown in Fig. 1. This research takes this kind of bookstore database as a case study and applies the proposed methodology on this. The catalog shown may be considered as a logical view assembled from data contained in multilingual information, but searchable in a unified manner for multilingual users.

DESIGN

Supposing that a user wants to search for a book by combining criterions such as author, category, pricing and publishing information, we propose a simple methodology

Corresponding Author: Dr. K. Ganesan, School of Computing Sciences, Vellore Institute of Technology University, Vellore-632014, India

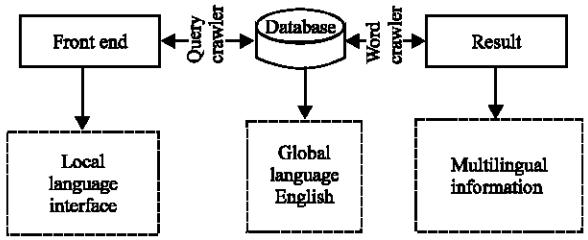

Fig. 2: Execution environment of multilingual querying and information processing

such as shown in Fig. 2, which takes input as multilingual query string or primarily local language query but returns the result in the user-specified set of languages namely English and/or Tamil, with returned tuples phonemically/semantically close to the query string. The specification of all the list of languages would have brought all records containing result of requested query.

The input, i.e., the query string contains both keywords and informal-words, which are used in daily life. To convert this blend into database understandable format, each word is scanned to test whether the word under study is a keyword or not, by looking it into the common-word-table. If it is a commonly used word, it is removed from the search sentence. The query-crawler algorithm, follows a mechanism wherein the keywords form a group in such a way that a query will be formulated and can be applied to the master database.

The design of the methodology is such that database information will be stored in character-based foreign language, namely English. The local language used here, Tamil, is glyphs-based. To map the two, word-crawler algorithm is used, as there is no one-to-one mapping. By the concept of Backtracking, the letters of local language are being formed from the foreign language words.

The letter formed is searched in the Phonetics-based-Unicode (multilingual.com) database, which is an architecture neutral flat file. The combination of backtracking and searching makes the word-crawler algorithm to display the local language content.

IMPLEMENTATION STRATEGY

The case study of this implementation is web oriented and hypertext markup language is used as core part. The underlying mechanism, Unicode, which is in hex format, is not applicable to hypertext markup language. For this we have a mapping table, called Unicode-mapping-table that stores the four digit decimal equivalent of all Unicode characters. For searching multilingual information, there exists two methodologies, one is based on Phonemes and another is based on semantic matching.

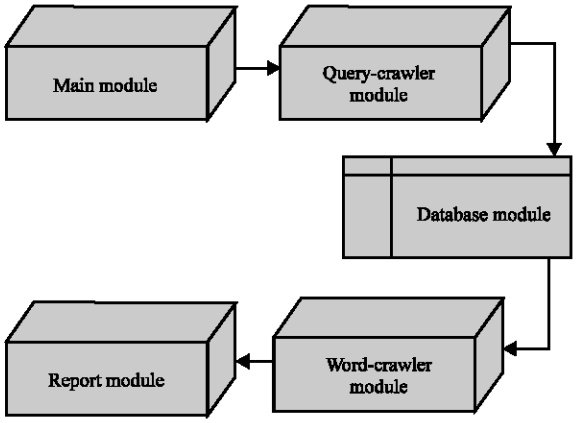

Fig. 3: Modules involved in the implementation strategy

For the first kind namely multilingual phonemic matching, which is matching text strings based on their phonetic equivalence (Apama *et al*., 2001) the Unicode-mapping-table is used. Example of this kind is name of authors and publisher names, which does not involve any translating functionality.

For the later kind namely multilingual semantic matching, which is matching text strings based on their generalised meanings, irrespective of languages, the local language names are mapped to the global English language words and then the searching is done by means of proposed query-crawler algorithm. For example, the English language word we will be translated into naam or namadhu in Tamil language. Sample snap shot is shown in Fig. 6. The various modules involved in the implementation strategy are shown in Fig. 3.

The main module is the initial user interface module and the search is being requested by means of available kinds i.e., with respect to author, title, category, publisher, price, advanced search, search engine or translator engine. The user interface module has provision to key-in multilingual content in a single text box. The input can be given from three ways. (1) The drop-down box having commonly used words, selected categories and author names of local language content, (2) the built-in local language keypad and (3) the ordinary keyboard. Sample snap shot is shown in Fig. 6.

Multilingual semantic matching: The multilingual query contains both languages. The query-crawler module, shown in Fig. 4, applied in this environment scans through each word in the query string and transforms the multilingual statement into a database understandable query by checking each word, whether a keyword or not by looking onto the common-word-table in the database.

```
QueryCrawler(S)
Input: String S contains multilingual information
Output: Formulated Query
1. S ← S_l + S_r
2. L_l ← Language of S_l; L_r ← Language of S_r
3. foreach word w_i in S
4.     IF w_i is keyword w_k Then
5.         insert w_k into the query string S_q
6.     Else
7.         return Common Word
8. Formulate query string S_q
```

Fig. 4: Query-crawler algorithm

```
WordCrawler(W)
Input: Foreign language, English word W
Output: Local language words W_l
1. Loop
2. Combine maximum number of characters
   and formulate subword W_s
3. Search W_s in the Unicode-Mapping-table T
4. If W_s is found in 'T' Then
5. Replace equivalent Unicode digit sequence
6. Else
7.     Backtrack and reduce number of characters
       to be combined
8. Until all the letters in W exhausts;
```

Fig. 5: Word-crawler algorithm

The multilingual query string can be converted into local language words using other standard linguistic resources also, such as WordNet (Apama *et al*., 2001). The database module is responsible for the following sequence of operations:

- The formulated query from the query-crawler module is given to the database.
- The required fields to be displayed from the result set are chosen.
- A result table is created based on the query and the required fields.

Multilingual phonemic matching: The strategy for matching multilingual data based on Phonemes is shown in Fig. 5. To achieve the Phonetic mapping, backtracking and searching concepts are used. First, from the foreign language word, maximum number of possible characters is grouped to form a sub word and they are given a backtracking criterion. The sub word is matched with the unicode-mapping-table and if found, the equivalent unicode character sequence is replaced in the output. Else it is omitted as a common word. The proposed algorithm scales well. Even for thousands of words, the algorithm proved its capability.

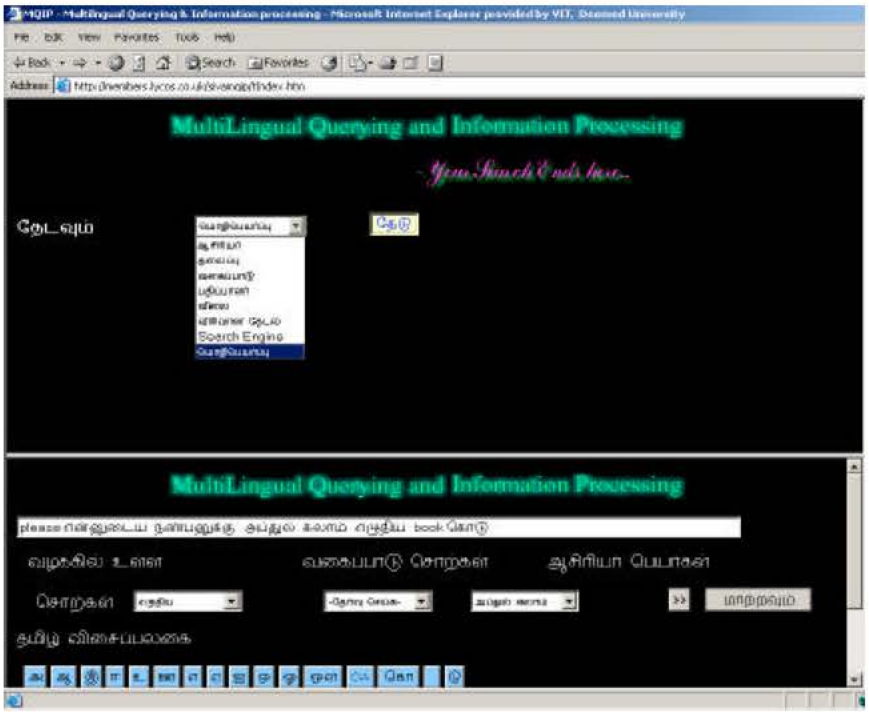

Fig. 6: The typical implementation of multilingual querying and information processing

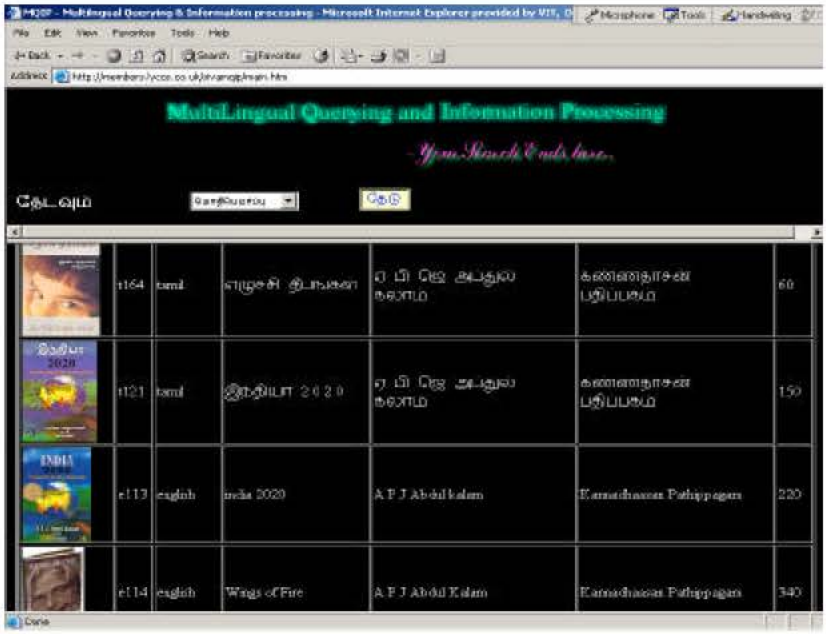

Fig. 7: Outcome of multilingual query

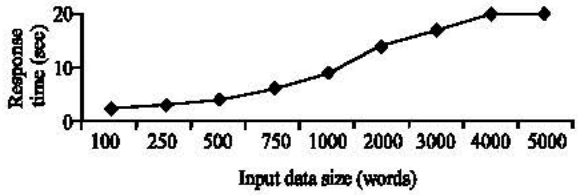

Fig. 8: Time complexity measure

We can further enhance the performance of the matching based on the classic soundex algorithm (Kumaran *et al*., 2003) or by adopting q-gram techniques (Haritsa and Kumaran, 2003). The matching algorithm and its performance are discussed by Kumaran *et al*. (2006). The other approaches for phonemic matching strategy are using International Phonetic Alphabet (IPA). The IPA phoneme strings are also stored in the unicode encoding format, as specified by the unicode consortium, using basic latin and IPA supplement code characters.

In the report module, each row of the result-table is checked to see, in which language the output is to be displayed. If it is foreign language, it directly publishes the row information. On the other hand, if it is local language, the word crawler module is called to display the information in the local language. The final output information can be copied into a storage place like WordPad etc. or they can be pasted in the text field of the Email to transfer it over the internet. A sample output of report module is shown in Fig. 7.

Performance issues: As a natural language processor, this paper's implementation crosses morphological, syntactic and semantic levels. How ontology is applied in the implementation is that the keywords in the database are treated as individuals. Along with the keywords, when the grammatical symbols combine together, it forms the class.

At the time complexity side, since in our implementation the back-end-database is in English and storing, querying etc. like database operations solely in English this implementation will be on par with the monolingual-English-database operations' performance. The implementation's database consists of more than 300 entries. The implementation demonstrated that the amount of time required to query multilingual information in this 300+ entries grows very minimal as the size of input data grows (Fig. 8).

CONCLUSIONS

In this study, we proposed architecture for processing multilingual data transparently across languages, on the traditional information processing platforms, such as relational database management systems. A survey on the functionality and performance of the current systems indicate that they fall short of the requirements on several counts (Davis, 2001) motivating further research on multilingual systems.

The proposed two algorithms are specific and have the capability to query and process multilingual information in a stipulated period of time as discussed in the time complexity measure above. From the functionality perspective, the proposed algorithms work for specific domains. Query-crawler was designed to work with Semantic domain and word-crawler to work with Phonemic domain. Thus we proposed a multilingual querying that displays result in a very minimal amount of time.

From the efficiency perspective, since the database and its related operations only involves English, our unicode based multilingual querying will be more like ASCII querying only. The rest of the mapping is done by means of the above-discussed modules effectively.

This study intends to address the problems related to accessibility and publication of content in a local Indian language, initially to support Tamil and later to support the requirements of many Indian languages. The same concept can be transported to other Indian languages also by using this paper's work as a template. Through this concept, we also intend to make computing and internet be habitually accessible to everyone. There will be a day in near future, where our grandmother uses electronic spreadsheets to reckon her home budget.

REFERENCES

Apama, G., G.L. Jayavardhana Rama and Ramakrishnan, 2001. Machine reading of Tamil Books: An aid for the blind. Proc. Biovision, IISc, Bangalore, 3: 34-41.

Davis, M., 2001. Unicode Consortium Technical Report, 2001.

Haritsa, J.R. and A. Kumaran, 2003. On the costs of multilingualism in database systems. In: Proc. of the 29th VLDB Conf., Berlin, Germany, pp: 93-97.

Kumaran, A., 2003. MIRA: Multilingual information processing on relational architecture. Proceeding of Computer Science and Automation, Indian Institute of Science, Bangalore, pp: 113-120.

Kumaran, A., P.K. Chowdary and J.R. Haritsa, 2006. On pushing multilingual query operators into relational engines. Proceedings of the 22nd International Conference on Data Engineering, pp: 98-98.