Full Text Search in Multi-lingual Documents
- A Case Study describing Evolution of the Technology
At Spectrum Business Support Ltd.

This paper was presented at the ICADL conference December 2001 by Spectrum Business Support Ltd. (SBSL) This paper may not be reproduced or published without the explicit permission of SBSL. Contact *info@sbsworld.com* for futher details

A. Executive Summary

Full text search & retrieval is the bedrock technology that supports almost all major technology initiatives in the areas of Information Retrieval and Knowledge Management. At Spectrum Business Support Ltd. we have been working in these areas for over 10 years. **WordMiner™**, our text & image database management system has been used for this entire period to develop applications involving large text and multi-media databases.

Today, with the spread of the Internet we have the death of distance but an increasing appreciation of the importance of geography and geography-based diversity in languages and databases. In this context, the need for technologies to ably address multi-lingual databases is increasing with leaps and bounds.

Here, we would define a Multi-lingual document as a document containing more than one language. We are not referring to different documents each in a different language, which may be called multiple language files, but necessarily, documents developed with content in more than one language. More often than not, this would mean English and the mother-tongue/native language of the author/reader/both. This could mean more than one script in the same document. Also, possibly transliterations from other scripts into Diacritical Roman. **WordMiner™** supports such needs ably and fully. This paper attempts to describe the evolution of this technology.

WordMiner™ began life as a full-text search & retrieval engine for English-language databases, primarily addressing law information. It has slowly but surely grown in sophistication. **WordMiner™** now capably addresses multi-media information access issues. All standard access methodologies viz., usage of Boolean search parameters, proximity parameters, (thesaurus support) combination text and index search, progressive search, and so on are fully supported.

Four years back, a major South Indian publishing organisation, needed access to such capability in respect of Malayalam. This was achieved and brought to the same level of sophistication, as was possible with English, or more appropriately, the Roman script. The technology was then successfully extended to the Devanagari and Diacritical Roman scripts. Today, **WordMiner™** fully supports all single-byte scripts, including therefore all Indian languages within its ambit.

Meanwhile, a need arose for creating the ability to address multiple languages or scripts within the same document. This need was felt by a leading Oriental Research institution in Western India. This problem was resolved by the end of October 2000, with

demonstrable ability to address documents with content in Sanskrit, Roman and Diacritical Roman.
This paper discusses the issues that crone up and were addressed along this path. To close, we will enumerate further research issues, including issues relating to Double-byte character-sets such as those used by the Japanese, Korean and Chinese languages, order of representation differences such as those used by Arabic, transliteration, translation, intelligent classification of content in different languages, and so on.

B. Single Language Search

The basic steps for providing single language search and retrieval are summed up below:

Firstly, let us discuss the case of doing a simple single word search in a single language. The process would be simply described as below:

- Submit all documents that will be searched into a database to the database management engine.
- Divide each document into "words" - character strings separated by commonly and consistently understandable separators such as "space" marks
- Build a list of unique words in a dictionary. Dictionaries are basic building blocks of search.
- Repeat for all the documents with the dictionary itself growing for new unique words.
- Store an inverted index of words and documents in which they are to be found.
- Additionally store offsets from start of the document, if certain proximity searches are to be permitted.
- When a search request is received, match the requested string of characters with the "dictionary" and report the location of the string if it exists in the database.

In essence, this is all that is required for single word searches.[i] These can, however, result in a very large number of documents being found containing the word the user has searched for.

To provide more focus, the engine provides the ability to query multiple words in combinations defining their inter se relationships with each other. Boolean Algebra provides expressions of set theory that would allow users to obtain results as they desire. Boolean Search is a mathematical (set theory based) way of specifying how the two (or more) words in a search instruction are related to each other. The logic requires some careful handling,[ii] but no additional information is required to be pre-stored.

For example, if a user wants to look for a vacation package in Hawaii or Paris, he can search Hawaii OR Paris AND vacation. This will give all the documents containing vacations in Hawaii or Paris or both! Here 'OR' and 'AND' are used as Boolean operators. In case no Boolean operator is specified, **WordMiner™** takes the 'AND' operator as default.

For further fine-tuning, we turn to Proximity searches. Proximate occurrence of words typically tends to provide more relevance to search results, thereby reducing the junk data that the user needlessly must scan to arrive at the valuable nugget of desired information. To enable proximity searches, the process could be described thus:

- When multiple words are to be searched, in proximity with one another, an additional item of information is also required to be saved i.e. all the offsets or locations in the document where the word exists.
- This information can be saved in several different ways. As a byte offset from the start of the document or as word-count offset from the start of the document or in any other manner. The trade-off is primarily between providing early search results and minimising the size of the dictionary or other overhead, by whatever name called.
- Armed with this additional information, search can be extended to a specified proximity i.e. the two words occur in the same document, in the same page, in the same paragraph, in the same sentence or in variations of these criteria.
- If the order of the two words is relevant, as is in case of a phrase, the logic is easy to extend to enable phrase search.

Today, with **WordMiner™**, single language search with Proximity Search and Boolean Search features enabled can be considered to be a proven art with many successful implementations. (See Appendix A and B for screen shots of Grand Jurix in Operation) It requires careful handling of large data. It also requires tweaking of algorithms to ensure fast finding of the required data and displaying of the data in the manner that the user expects it.

Our experience with users suggests that they have come to expect this minimum level of proficiency from the software product. They are, also, due to their focus on their professional needs, focused a lot more on ease of use and speed of use. These are parameters that need close and careful attention and a lot of hard and painstaking work to ensure user happiness.

Thus, by the time we had arrived at the conclusion that we had a reasonably accomplished software product; we were driven to conclude that this is just the necessary part of the equation. Sufficiency could only be achieved through attention to User Interface design and focus on value additions to content, quite apart from making the search engine more capable and sophisticated.

For this paper, however, we will now turn to the desire to make the engine more capable of addressing different languages - or more appropriately different scripts. To be exact, we began to look to extending our ability to search & retrieve characters going beyond say the first 64 characters from the ASCII standard.

C. Different Language Search

There were two steps we needed to take when requiring search in different languages.

Firstly, acquaint **WordMiner™** with the key differences between various languages. Essentially, **WordMiner™** needed to be taught the script and character set used by each new language. Each language uses a character set specifying the characters used in representing the language on computer. **WordMiner™** was taught - on being told before hand - to understand the script and character sets in a different language (to begin with Panchari for Malayalam, later Devanagari for Sanskrit) and to create dictionaries in each such identified character set.

Thereafter, the most important requirement for a search engine to move from a single well-understood language to any other, is to provide the ability to switch between various languages when the language of the document changes. The Search Engine must exhibit the ability to figure out the language within the document (within a set of documents which are themselves in two or more languages) and then search accordingly. Here, therefore, we built technology into **WordMiner™** that allowed the software to distinguish between scripts and character sets, without being told - before hand - which language was being addressed.

Incidentally in the first implementation of Multi-lingual search engine, there was another issue that needed to be addressed at the same time was to do with distributed databases to be accessed from geographically distant servers and providing access to user groups on Internets as well as off of the Internet. This was successfully done.

Then, having acquired confidence with one more script apart from Roman, we turned our attention to Devanagari. Why Devanagari - since it allowed us to address three languages while addressing one script, viz., Sanskrit, Hindi and Marathi.

After having achieved success with Devanagari, we approached an Institution in Western India with whose archival problems we were somewhat acquainted. On seeing the demonstration of single language searches in different languages, they came up with the problem of addressing documents that contained matter in more than one script. Primarily their concern was with three scripts - Devanagari, Roman and Diacritical Roman. The last script was often used to represent Devanagari characters where Devanagari could not directly be used.

D. Multi-lingual Search

Documents containing multiple languages contain multiple fonts. This type of search, therefore, requires capabilities very different from Multiple Language Searches.

To begin with, the document under consideration is written in more than one language (two or more languages have been used within the same document) and all the features of the Search are to be made available across all the languages of the document.

The single words to be searched could come from any of the languages found in the document. The proximity search could be between two (or more words) could be from two different languages. Similarly Boolean Search could also involve words from multiple languages simultaneously. Successful implementations of multi-lingual searches are rare[iii].

Basic Search:

While searching, the search engine identifies the language of the word to be searched and activates the corresponding dictionary to get the result. Thus, if words in multiple languages are required to be searched simultaneously, multiple dictionaries are activated to get the desired result.

For example, word 'country' will be searched in English dictionary. Word भारात will be searched in Hindi Dictionary and word દેશ will be searched in Gujarati dictionary.

Boolean Search:

WordMiner™ now provides Boolean search in multiple languages. While searching **WordMiner™** will search the words independently by activating respective dictionaries and then give the result after performing the Boolean operations desired.

For Example, you can search for छुट्टी AND Paris, **WordMiner™** will search छुट्टी in Hindi dictionary and Paris in English dictionary to give all documents containing both the words.

Proximity Search - specifically Phrase Search:

A phrase is a set of words coming in continuation in a specific order. A phrase is required to be specified in double quotation marks (" "). **WordMiner™** identifies the language in which phrase is entered and enables the dictionary to search it. **WordMiner™** uses a sophisticated phrase search mechanism to search phrases. E.g. if you search for "George Washington" all documents containing the words "George" and "Washington" in the specific order will be given.

Currently, **WordMiner™** requires a phrase to be only in one language i.e. the entire phrase searched should be in the same language. However, **WordMiner™** can search for phrases of different languages can be given for search to **WordMiner™**.

For Example, भारात का संविधानcan be searched as a phrase, and will give all the documents containing भारात का संविधान in the same specific order.

Progressive Search:

All searches result in documents that contain the words searched for. It may not be necessary that all the documents resulted by a search are relevant to the user. To reach the desired document, it is necessary to search more words or phrases in results. This is called Progressive Searching.

For **WordMiner™**, the language of the previously searched word/phrase is immaterial. If the first search was for English words, and within the resulting document list,

documents containing a Hindi word are to be sought, the progressive search mechanism will search for the Hindi word(s) in the documents previously found by the search.

For example, if we search for "languages for computers", the user may locate a large number of documents containing the phrase. Further, she could search for संस्कृत in the results found by the search from "languages for computers". In the first search, **WordMiner™** will use the Roman dictionary and in the second search it will use the Devanagari dictionary to search संस्कृत.
(Appendices C, D, E and F contain screen shots of "Mahabharat on CD ROM" where the above aspects are shown at work.)

E. Further Research Issues

Based on our interactions with users and looking to the emerging needs, we believe we will need to conduct further research in the following areas to develop truly appropriate technology:

1. Addressing the need for search & retrieval of strings in languages represented in Double-byte character-sets such as Japanese, Chinese and Korean.
2. In most languages, text is written from left to right and from top to bottom. A few languages do, however, do things differently. For example, Arabic is written from left to right. For such languages, character strings will need to be searched in the reverse order.
3. In order to provide fully multi-lingual search & retrieval, transliteration will need to become easy and ubiquitous, allowing for changing script without changing language.
4. Apart from merely searching for strings, software will need to be developed for intelligent classification of content in different languages, so as to allow for more focused searches that reduce result list length when coupled with full text searches.
5. Finally, translation automation - domain by domain - will need to be developed to a level of significance, using Artificial Intelligence techniques to arrive at meanings and thereafter meaning-based searches.

[i] See the following for an understanding of history of Text retrieval :
Knuth, D. 1973. The Art of Computer Programmig : Sorting and searching, vol 3, Readings, Mass.: Addison Wesley
Belkin N.J. ‘ and W. B. Croft. 1987. “ Retrieval Techniques,” in Annual Review of Information Science and Technology .ed. M. Williams. New York : Elsevier Science Publishers, 109-145
Faloutsos ,C. 1985, “ Access Methods for Text”, Computing Surveys,17(1), 49-74
Ricardo A. Baeza-Yates, 1992,” String Search Algorithms” in Information Retrieval, ed. William B. Frakes Ricardo Baeza-Yates, Prentice Hall New Jersy 219-239

[ii] Steven Wartik, 1992 “Boolean Operations” Information Retrieval ed William B. Frakes, Ricardo Baeza-Yates, Prentice Hall

[iii] See appendix G for an example of multiple language search posing as multilingual search