11

Tamil Search Engine

J. Deepa Devi, Ranjani Parthasarathi, T.V. Geetha

Resource Centre for Indian Language Technology Solutions – Tamil,
School of Computer Science and Engineering, Anna University
Chennai – 600 025, India.
E Mail: ***deepajayaveer@rediffmail.com, tvgeedir@cs.annauniv.edu, rp@annauniv.edu***

Abstract

The web creates new challenges for information retrieval. The amount of information on the web, as well as the number of new users is growing rapidly. Search engine technology has to scale dramatically to keep up with the growth of the web. In this paper, we present the details of constructing and maintaining a Tamil Search Engine. We discuss the issues such as the crawler, the database storage architecture the searcher and the functional modules of the search engine.

1 . Introduction

With the phenomenal growth of the web, more and more information are available online through web, from personal data to scientific reports to up-to-the-minute satellite images. Human maintained lists cover popular topics effectively but are subjective, expensive to build and maintain, slow to improve, and cannot cover all esoteric topics. Hence, searching for a particular information in this is not only time consuming, but also a daunting task. Thus comes the need of a search engine.

Search Engine is a software package that collects web pages on the internet through a robot (spider, crawler) program and stores the information on the pages with appropriate indexing to facilitate quick retrieval of desired information. The crawler continuously looks for updated information on the web and stores it in a database. An efficient indexing mechanism is normally used to quickly retrieve the informa tion. When a user queries the search engine for a particular topic, the search engine looks up the database and lists the pages containing information on that topic. In displaying these pages, some mechanism for ranking the relevant pages is used. Ranking can be done using the parameters such as number of occur rences of the word (location/frequency), the number of links coming into and out of the web pages etc[1].

This paper deals with the design of a Tamil search engine, that searches for Tamil documents on the web.

Like any other search engine, it consists of a crawler, an indexing mechanism and a database to store the information. However the challenges faced for handling Tamil documents and the Tamil language are different. There are innumerable Tamil sites available on the net. Moreover, the format in which they are stored/represented are different. For instance, different fonts are

used by different sites, each font having its own encoding. In spite of the standardi-zation that has been enforced by Tamilnadu Government (these standards are detailed very well on the Tamilnet99 web page [2]) which requires everybody to use TAM or TAB fonts, sites that have been hosted earlier have different encodings. Thus, the effective retrieval of content requires collecting information in all the fonts and converting it to a standard format before storing it. Another challenge is with respect to the language itself. Tamil being a highly inflectional language, every root word takes on innumerable forms due to the addition of suffixes indicating person, gender, tense, number, cases etc. Thus the number of words to be handled is large. This can be an issue in the design of the database. This problem has to be handled effectively.

The following sections of this paper describe the design of the Tamil search engine that addresses these problems, to provide an efficient search mechanism for Tamil documents.

2. Design of Tamil Search Engine

The Tamil Search Engine consists of the following major components:

- Crawler
- The Database System
- Search

The searcher component includes the ranking mechanism and the user interface. The design of each of the modules is described below:

2.1 Crawler Module

The crawler module is responsible for retrieving pages from the web and handing them to the database management module. A list of seed URLs is taken as its input and sequentially traces through all the links in the list. Each document is downloaded and then the links contained in each document are extracted. These extracted links are added to the URL list if they are not already present. This process is repeated periodically to obtain the updated status of the web documents. The document are then processed to extract word information. The documents could be multilingual. However, only English and Tamil content are handled by the search engine. The language is first identified and the content processed accordingly. Tamil words and English words are maintained separately.

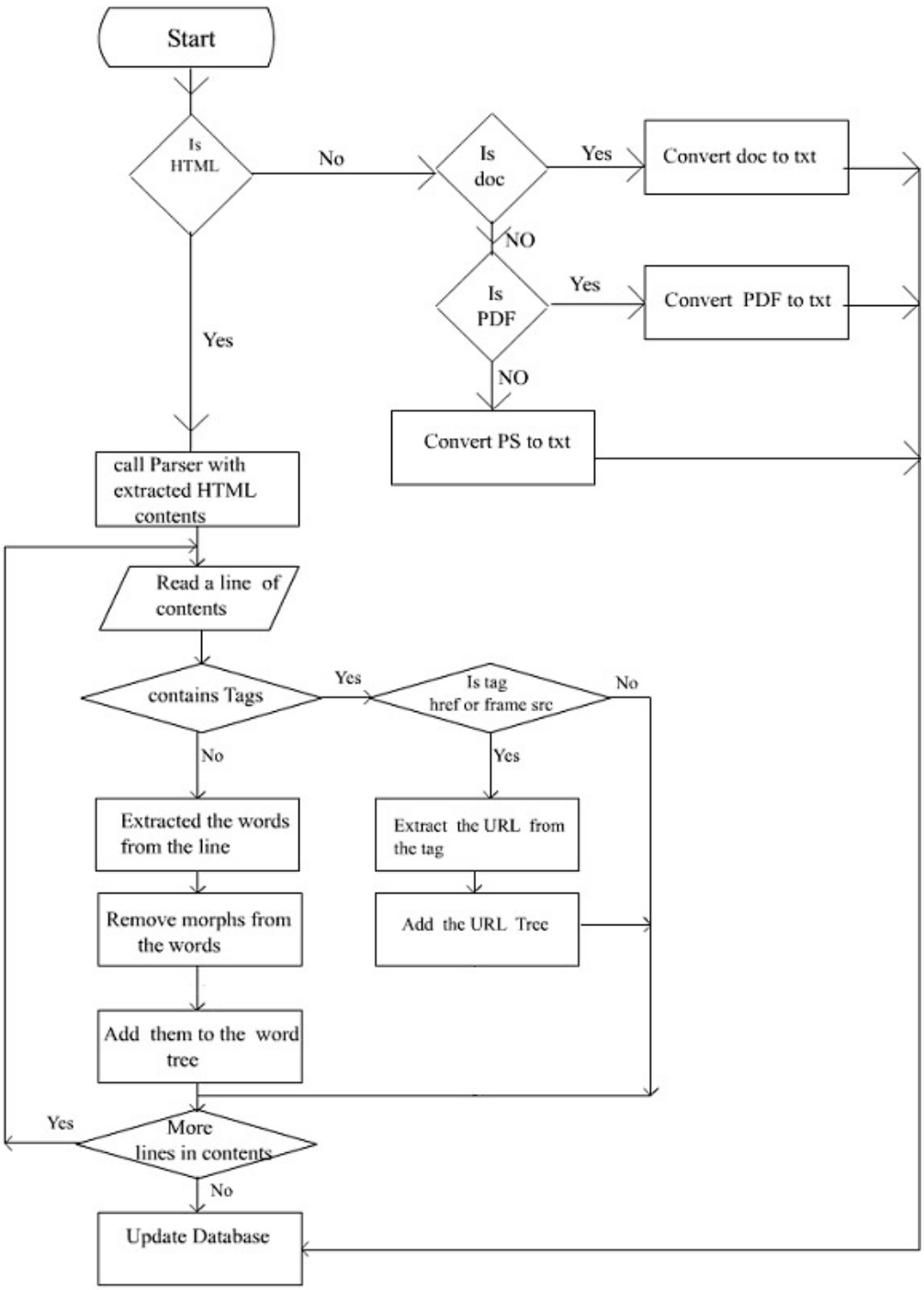
The effective retrieval of content requires collecting information in all the fonts and converting it to standard format before processing. Words in different fonts are converted to TAB font using generic font converters. These font converters take care of most of the popular fonts (such as TAM kumudam, Vika tan, Kalki, TMNews, LT-TM-Barani, LT-TM-Lakshman, LT-TM-Kurinji, Amudham, Elango-TML-Pan chali-Normal, Tboomi, TM-TTKapilan, TM-TT Valluvan, Sarukesi, Tamilweb, Aabohi, Anantha_shan mugathas, Bamini/Baamini, DenukaPC, Eelanadu, Inaimathi, Inaikathir, Arulmathi, Webtamil, Anjal-Sys tem, Anjal-Text, Tamilnet). The documents may be in HTML, doc, PDF or PS formats. The processing of the HTML documents is done as follows. The documents are parsed and words from the body part of the HTML file are retrieved. Then the stop words (whose content value is very low) are removed. For the rest of the words, the corresponding root words are extracted using the morphological analyzer. The root word is stored into a word BTree and then into the database. Extraction of root words and storage of only root words in the database is an important design decision. Since, Tamil is an inflectional language, each root word combines with numerous suffixes, and appears in many different forms. Storage of each of these forms would prove to be too costly, and is also unnecessary. Often, a user may specify one form of the word, but typically expects all information regarding the root word. That is documents containing other forms of the word are also normally desired.

Thus, identifying the root words, and indexing the documents based on the root word, makes the task of retrieving all forms of the root word easy.

Once the text part of the html contents is processed, the URL links in that page are identified. The URLs are identified by using the HTML tag such as <a href…>, <frame src…>, <base href…> and <area href…>. The URL links are then inserted into the URL list which is maintained in a URL BTree. For PDF, doc and PS file, the first step in processing is the conversion of these files to text files. PDF and PS files are converted using the existing utilities.

Once the text file is obtained, the same process of removing stop words and identifying root words using the morphological analyser is carried out. For all types of files, in addition to storing the root words in the database, the entire file is also stored locally as text files. This is required to present the content of the desired word in the document.

The following figure (1) depicts the over all flow of the crawler.



2.2 The Database System Module

The database management module is responsible for storing the latest version of every page retrieved by the crawler. Its goal is to store only the latest version of any given page. The Crawler and Searcher Modules use it for storing data and retrieving data respectively. Two sets

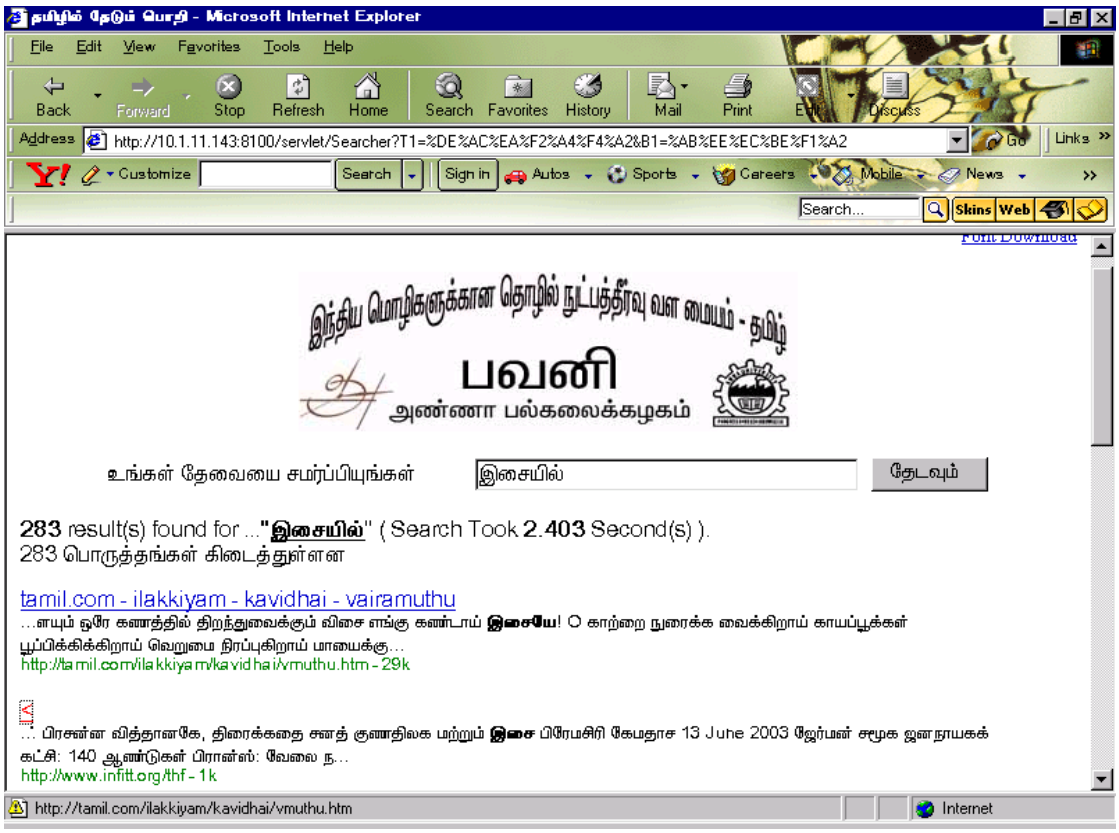
of tables are used in the data base module. One set is to store information regarding the words and the other to store information regarding the URLs.

The main functionality of the database lies in the first set of tables. Specifically, for every word in the documents the Document ID and the position of the word in the document are stored. The position of the word in various HTML tags is maintained and used for ranking analysis. English words and Tamil words are maintained in different tables. Thus, for a given word as a query, the database returns the information such as the documents in which it appears and the positions of the word in those documents very quickly.

The second set, namely the URL table is maintained to keep track of the list of URLs to be crawled. The field in the database are document ID, URL, last_modified date, outgoing links and the title of URL. Every time the crawler is invoked, this table is updated. The document ID and URL is maintained for indexing and the outgoing links is used for ranking analysis.

2.3 Search Module

In the search module, when a user queries the search engine for a particular topic, the database is looked up and pages containing information on that topic are listed. In displaying these pages, ranking mechanism is used. There are two major components in this module - the user interface and ranking algorithm. The user interface is shown in fig(2). A text field is provided for the user to enter the query in Tamil or English. For Tamil words, morphological analysis is performed on each word of the query to obtain its root form. The database is then queried based on the root word, and the records(pages) which satisfy the query are retrieved. A page ranking algorithm is then performed on these pages. The criterion used for ranking is the frequency count of the word in the document. The pages are then displayed in order of frequency count. A query may contain more than one word. In this case, documents containing all the words (i.e., corresponding to Boolean AND function of the words), would be first listed, followed by document containing a subset of the words (corresponding to OR function) Here again the ranking based on frequency count will be used. The results are displayed in lists of ten pages. The user has the facility to navigate through this list.



யீig (2)

3. Conclusion

This paper has described search engine for extracting and presenting Tamil content available on web site. The highlights of this search engine are its capability to handle multiple fonts, and the use of the morphological analyser. While the former broadens the search of the search engine, the latter improves the depth and efficiency of the search engine.

The current status of the search engine is as follows. 1100 sites have been crawled and the time taken for a query is 3 seconds. The performance could be improved by adapting more suitable searching and ranking algorithm.

References

1. Dell Zhang, Yisheng Dong, An Efficient Algorithm to Rank Web Resources, http://www9.org/w9cdrom/251/251.html
2. http://www.tamilnet99.org/
3. Brian E. Brewington, George Cybenko - How dynamic is the web? http://www9.org/w9cdrom/264/264.html
4. Search Engine Technology web site, http://searchenginewatch.com/
5. Daniela Florescu, Alon Levy and Alberto Mendelzon - Database Techniques for the World Wide Web: A Survey, http://gaston.snu.ac.kr/xweet/seminar/990728-jmjeong/www.html