

***iAgent* : A System for Managing  
Networked Tamil and Multilingual Information Resources**

K. Rajaraman and Kok F. Lai  
Information Technology Institute  
11 Science Park Road, Singapore 117685  
Republic of Singapore  
kanagasa, kflai@iti.gov.sg

**ABSTRACT**

The advent of World Wide Web(WWW) has created a novel means for information dissemination whereby information resources all over the world can be made available to a user connected to the net anywhere and anytime. As more and more information resources are becoming available on the WWW, providing easy access to these information resources has become a significant service. In this paper we present *iAgent*, a system for information collection and retrieval designed specifically for networked Tamil and multilingual information resources. It fulfils the critical role of an information search tool to access the rapidly growing pool of WWW resources provided in Tamil or other Asian languages. The system is build around a multilingual architecture that confines language processing requirements into specialised modules. These modules share a standard library which allows the user to collect, index, retrieve and set profiles on networked information resources. We demonstrate the usefulness of this system by showing how it can be applied to search and retrieve Tamil webpages from all over the world. The engine can also support several major languages in Asia, including English, Malay, Bahasa Indonesia, Thai, Chinese, Japanese and Korean.

Keywords: Tamil on WWW, Tamil information processing, Multilingual information resources, Information retrieval, Search engines.

**1. INTRODUCTION**

The revolution brought about by Internet is now a well-known and fully felt phenomenon. With the availability of services like World Wide Web (WWW) , Newsgroups and Email, Internet has made it possible to disseminate information to a user connected to the net anywhere and anytime. Among these servcies, WWW has grown explosively during the recent years resulting in a novel networked information resource. The growth of the web is positive on one hand because more information can be accessed now. However, as more and more information resources are becoming available on the WWW, locating the desired information has become difficult [1]. Hence, providing easy access to such networked information resources has become a significant service. Currently, several commercial search sites such as AltaVista (<http://www.altavista.digital.com>), Excite (<http://www.excite.com>), Infoseek (<http://www.infoseek.com>) and Yahoo (<http://www.yahoo.com>) are very popular, generating millions of hit daily.

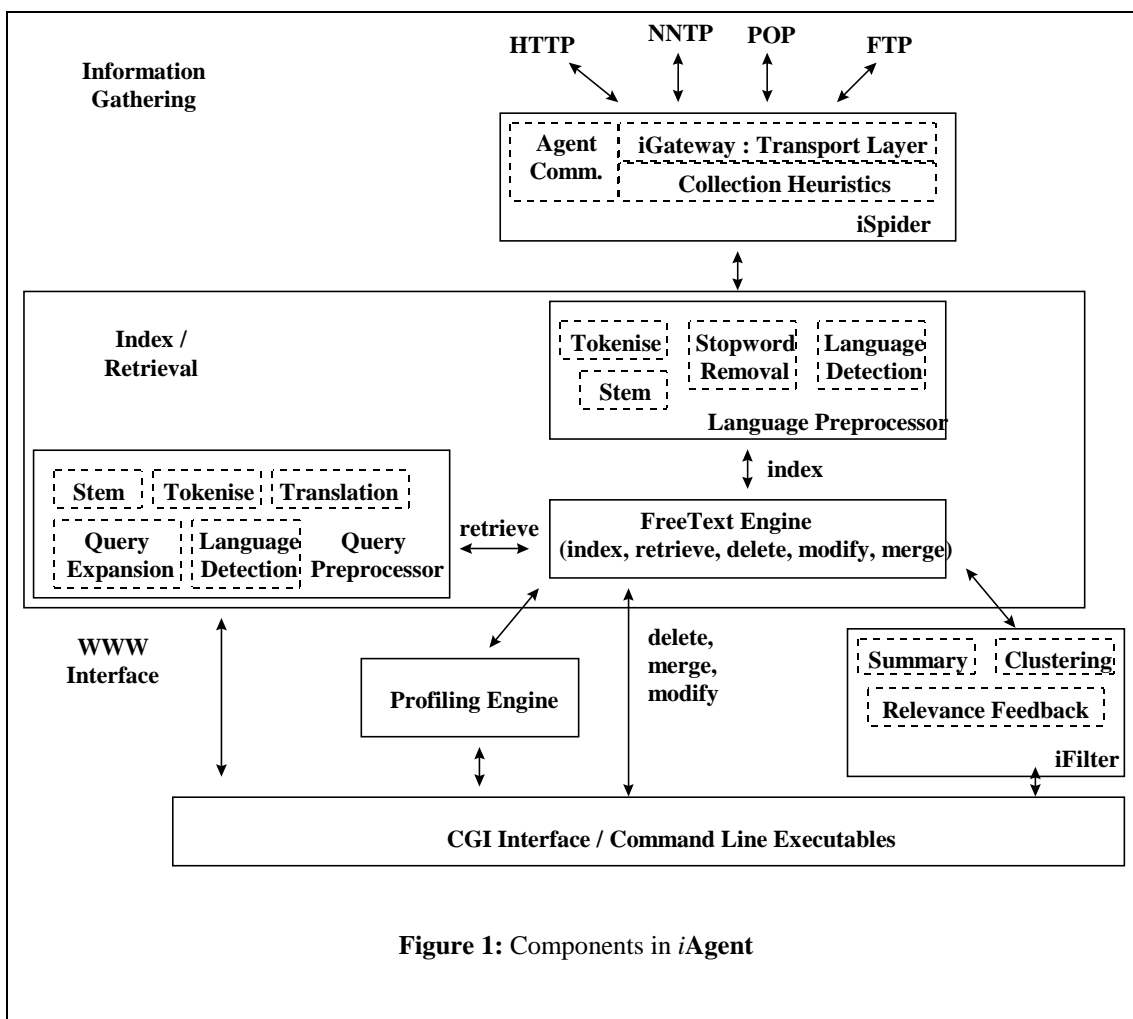
Having originated from the West, these search engines have not been developed with multilingualism in mind. As a result, native languages often take a back-seat due to technology limitations. For the vision

of international information superhighway to be realised, providing multilingual search capability is important. Currently, there are few search engines available for tackling native languages [2]. (These engine are sometimes confused with engines like, for example, Theni Search [3], which actually search in English and hence cannot be called native.) However, most of them are simplistic because they are either tailored for one specific language or do not have spidering capabilities. As an example, for Tamil, a search engine has been developed by Internet Research and Development Unit(IRDU), Singapore for searching Tamil documents [4]. The engine accepts search strings in Tamil through a JAVA applet front-end and employs a simple WAIS-SF indexer to search and retrieve documents. It does not have spidering capabilities. Instead, it assumes all documents to search are available a priori on the local machine. Also, in its present form, it can search only Tamil documents.

This paper describes ***iAgent*** (<http://iagent.iti.gov.sg>), an information collection and retrieval system designed specifically for multilingual information resources. We designed and built a multilingual architecture which confines language processing requirements into specialised modules. These modules share a standard library which allows the user to collect, index, retrieve and set profiles on networked information resources. We have demonstrated the usefulness of this system by showing how it can be applied to search and retrieve Tamil webpages from all over the world. The engine can also support several other major languages in the South East Asia, including English, Malay, Bahasa Indonesia, Thai, Chinese, Japanese, Korean and Filipino Tagalog.

## 2. SYSTEM DESCRIPTION

Figure 1 shows the system architecture for ***iAgent***. The various components are described in the following sections.



**Figure 1:** Components in *iAgent*

## 2.1 Information Gathering

*iAgent* currently supports the following protocols, which allows it to automatically collect information widely distributed on the Internet:

- *HyperText Transport Protocol (HTTP)*. Based on this protocol, a knowledge robot, or web spider is developed. The robot starts from a list of known sites, and automatically traverses the web's hypertext structure by retrieving a "valid" document, and recursively retrieving all valid documents that are referenced. A "valid" document is that which is judged to have been produced by a website selected for information gathering. The system employs simple heuristics which rely on domain name restrictions (e.g. `http://*.gov.sg`), directory level restrictions, as well as "kill list" (e.g. do not visit `http://abs.xyz`) to determine if a document is valid.
- *Network News Transport Protocol (NNTP)*. The system uses the NNTP protocol to interrogate USENET news server for selected newsgroups. During each access, the system retrieves new postings from these newsgroups and formats them for subsequent indexing and retrieval.
- *Post Office Protocol (POP)*. Using the POP protocol, the system is able to retrieve electronic mail from personal mailbox for subsequent indexing and retrieval.

With these tools, a system administrator has the flexibility to target selected websites, newsgroups and mailboxes for information gathering. For example, by providing a list of desired Tamil website URL's, an archive of Tamil webpages can be easily constructed by sending the web spiders to collect all web pages at these URL's.

## 2.2 Free-Text Engine

This component creates the inverted index [5] of all documents collected by the information gatherer. The inverted index will then be used during document retrieval. In our multilingual architecture, language-specific processing is confined to dedicated modules. The database layer is unaware of the language type of the *tokens* being indexed and retrieved. Logically, the inverted index table has the following structure (Table 1):

Token	Inverted Entries
Compute	DocID, tf, pos1, pos2, ....., DocID, tf, pos1, pos2, ....., .....
ஊர்	DocID, tf, pos1, pos2, ....., DocID, tf, pos1, pos2, ....., .....
.....	....

**Table 1** : Logical Structure of Inverted Index

Thus, if a user wishes to retrieve documents containing the word “compute”, the corresponding document id's (DocID) in the inverted table are retrieved, together with their term frequencies (tf) and positions (pos1, pos2, etc.). The term frequencies are used in the ranking of the document, while the positions are used to determine if consecutive query terms constitute a phrase (e.g. ஊர் அமைக்கப்பட்டது).

The following sections describe the various modules in more detail.

### 2.2.1 Indexing

The first step of automatic indexing is lexical analysis; it converts an input stream of characters into a stream of words or tokens, then generates lists of index terms. A *tokeniser* is used to identify words within the document structure that contribute to the content. Tokenising is easy for Tamil whereas it is nontrivial for some Asian languages like Chinese. Chinese text has its own characteristics; it includes a large set of characters, about several thousands for common use.

Most words are indexed except for very common words, which are maintained in a stopword list. Stopword removal is required to discard words that are non-content bearing and hence do not contribute to the description of the document [6]. They typically consist of terms that are too frequently present to be a good discriminator; a search using one of these words is likely to retrieve almost every item in a database regardless of its relevance. Examples of such words in English include “the”, “is”, “with”, “this”, “and”, etc. Similar stopwords also exist in Chinese text. The selection of stopwords depends on

the database and features of the users and the indexing process. It may vary from application to application. For example, a computer literature database probably need not use index terms like “computer”, “program” and “language”. However, these words should be indexed in other applications such as movie stories.

Stemming is another way to improve the performance of the information retrieval. It is used to reduce the size of index files. Content bearing words are then stemmed to remove the redundancy of word variants and also to reduce the indexing overhead. So in English document, rather than storing the words “computer”, “computers”, “compute”, and “computing” separately, these words are stemmed to a common root form “comput”. Only this stemmed word or index term is stored. There are several approaches to stemming, such as table lookup stemmer, successor variety stemmer, n-gram stemmer, and affix removal stemmer, etc. Porter's stemming algorithm [7] is used here, because it is more compact than other algorithms. This algorithm belongs to affix removal ones, and it consists of a set of condition/action rules. It may be noted that, for some Asian languages like Chinese, stemming is not applicable due to the different nature of characters.

During indexing, important sentences from the input documents are extracted and stored as document summaries. The summary consists of a set of sentences that is judged to be representative of the document. The number of sentences to be extracted can be specified by users. In addition, the titles and the URL's of the original documents are also stored so that users can easily retrieve them.

### 2.2.2 Retrieval

After indexing all the required documents, the system is then ready to accept queries and retrieve documents. The query strings can consist of phrases in any of the languages supported, as well as Boolean operators (AND, OR, NOT). The system will return the titles, summaries, URL's and the matching scores of documents found. The scores are computed using the dot product of term frequency (*tf*) and inverse document frequency (*idf*) weighting. *idf* is measured as follow:

$$idf_i = \log N / n_i \quad (1)$$

where  $N$  = the number of documents in the collection,  $n_i$  = the total number of occurrences of index term  $i$  in the collection. Thus, the score of document  $j$  due to term  $i$  is given by :

$$\text{score}(doc_j, term_i) = tf_{j,i} * idf_i \quad (2)$$

Fuzzy boolean operators are used. Thus

$$term_1 \text{ and } term_2 \rightarrow \min(\text{score}(-, term_1), \text{score}(-, term_2)) \quad (3.1)$$

$$term_1 \text{ or } term_2 \rightarrow \max(\text{score}(-, term_1), \text{score}(-, term_2)) \quad (3.2)$$

$$term_1 \text{ not } term_2 \rightarrow \min(\text{score}(-, term_1), 1 - \text{score}(-, term_2)) \quad (3.3)$$

Retrieved documents are ranked in decreasing order of similarity to the query. For phrase search (e.g.  $\emptyset' \tilde{\text{A}} \tilde{\text{O}} ; \tilde{\text{U}}' \hat{\text{A}} \tilde{\text{O}}$ ), the system will perform full-text search for documents containing the each of the constituent words (e.g.  $\emptyset' \tilde{\text{A}} \tilde{\text{O}} ; \tilde{\text{U}}'$ ,  $\hat{\text{A}} \tilde{\text{O}}$ ). The documents retrieved are then checked for adjacency of these terms.

In addition, the system provides a suite of information retrieval tools for the users to query the databases :

- *Group into clusters.* The system analyses the contents of the retrieved articles, and automatically groups these articles into clusters. A cluster consists of two or more articles with significant overlap in the contents. Users can therefore quickly inspect the different categories produced to locate the right piece of information.
- *Document extracts.* The system automatically extracts the most important sentences from a document to form the document extract. This allows users to quickly form opinions about usefulness of the document before retrieving the entire length.

Apart from the above content-based search capabilities, field search capability is also available in the system. A field structure file is defined to describe the document fields such as document identification number, database name, title, URL, date, and summary, etc. Each of these fields can be set to be searchable.

### **2.3 WWW Interface**

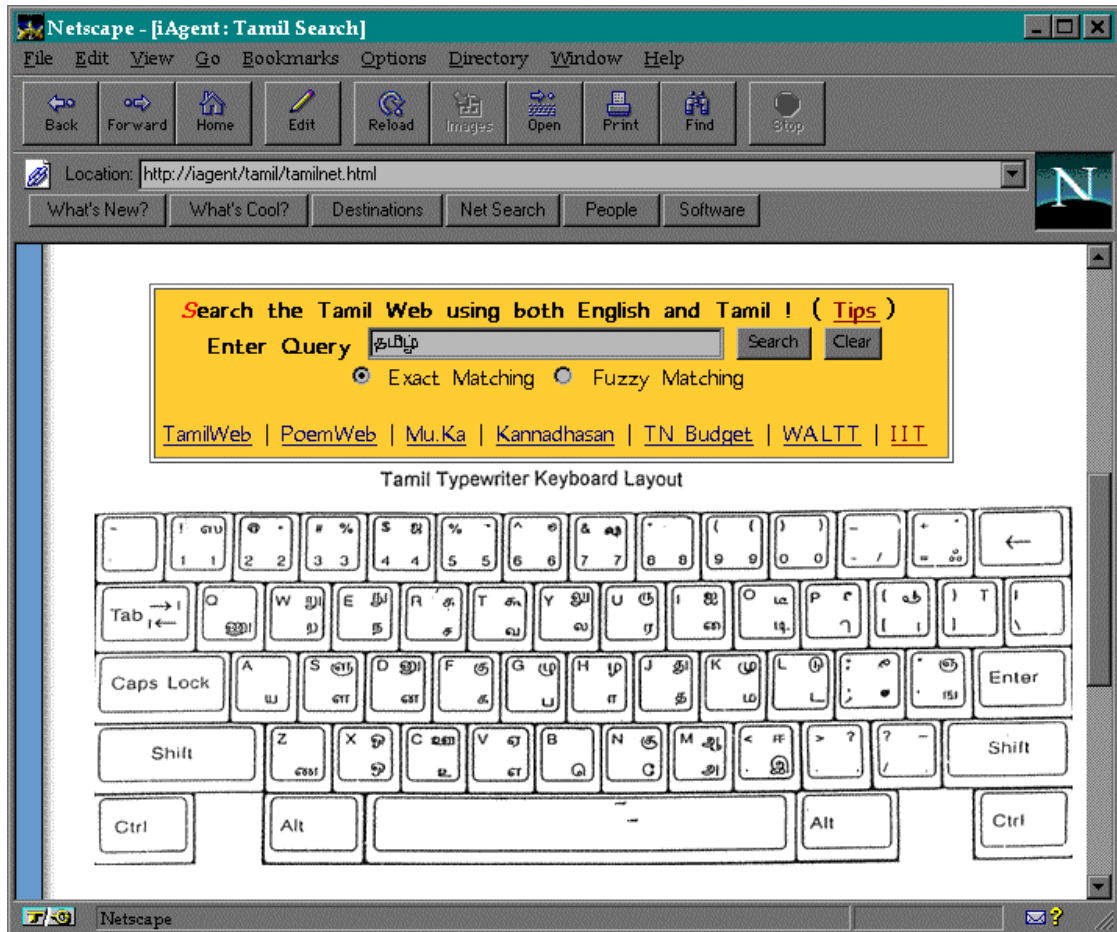
Users interact with *iAgent* via simple World-Wide-Web (WWW) interface, using standard web browsers such as Netscape or Microsoft's Internet Explorer. Search requests are submitted using HTML forms. To input Tamil search strings, at present, we are using the Keyboard Manager software available for free download from the IRDU, Singapore's TamilWeb site (<http://irdu.nus.sg/tamilweb>). In this non-commercial version, the keyboard manager uses the Tamil typewriter keyboard layout. Since this is a standard keyboard layout and followed worldwide for Tamil input, there is no need to learn a new layout. Our interface can also support the *Kanian Keyboard* layout [8] and the Romanized Tamil Keyboard layout (<http://www.murasu.com>) through suitable keyboard management utilities. In future, Tamil input will be provided through a JAVA applet and consequently, no keyboard management software needs to be present at the client side.

After the search request is submitted, the results returned are formatted into HTML pages. Users can retrieve the full-text of any document by clicking on the hyperlink; the browser will then fetch the document based on the Uniform Resource Locators (URL) of the document.

## **3. EXAMPLE: Tamil Web Page Search**

We use *iAgent* to collect Tamil webpages from a list of specified websites and gather the information regularly using the web spider, create inverted index of webpages collected, and allow users to submit query to the information resource.

Figure 2 shows a snapshot of the search interface. User can select various information resources to search and enter their search string in the box provided.



**Figure 2** : A Form Interface for WWW Search

Figures 3.1 and 3.2 show the results from the sample query “»¾Ö°”. It can be seen that search results from various Tamil websites satisfying the query are returned. The titles, summaries and URL’s are shown to the user.

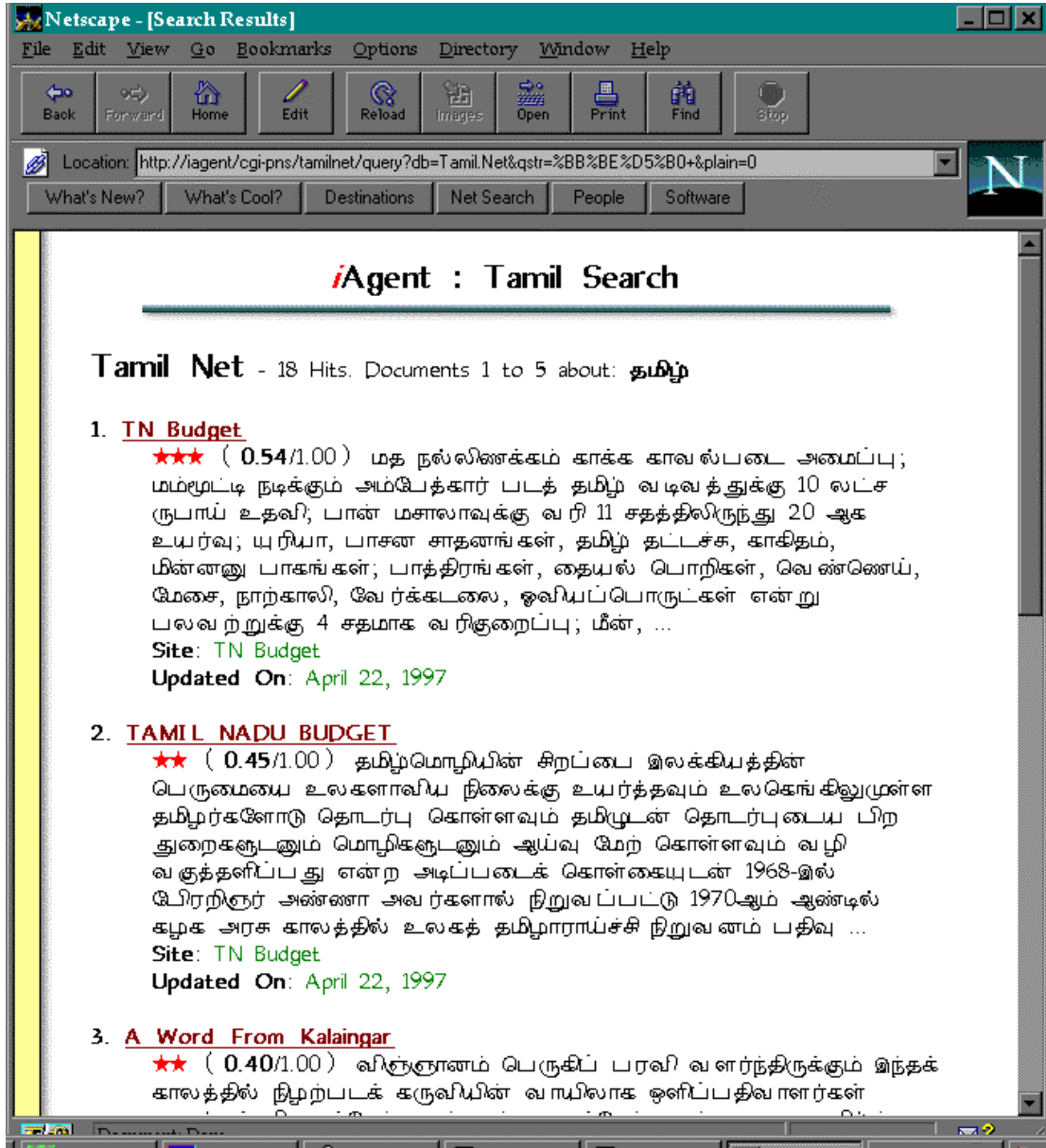


Figure 3.1: Search Results for a Sample Query

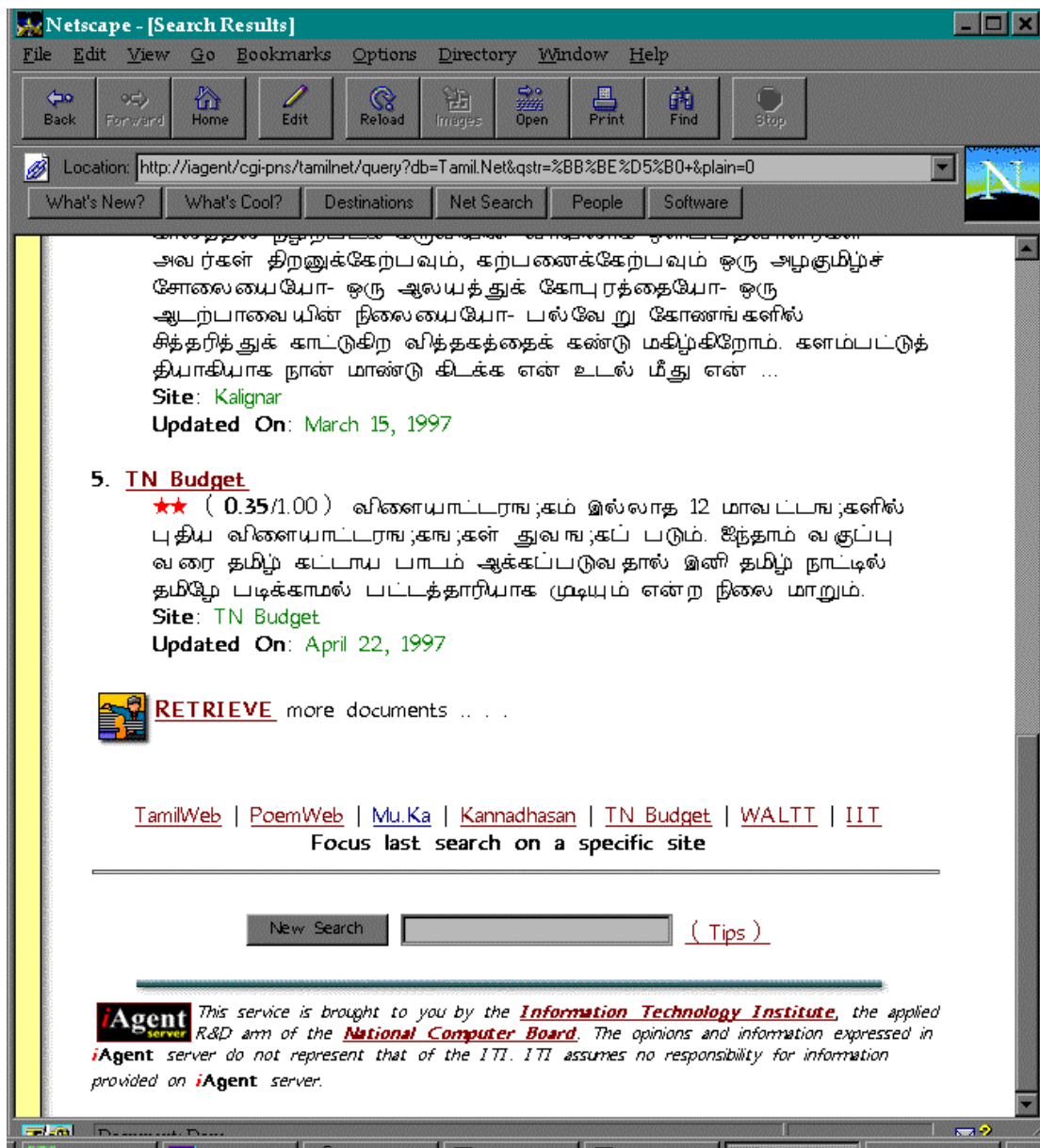


Figure 3.2: Search Results for a Sample Query

It can be seen that there were 18 hits for the sample query above and the results are returned from various Tamil websites. Suppose it is desired to use the same query but focus the search to get results from a specific website, say Kalaingar's web home. We have provided a means for implementing this

focussed search. For example, a focussed search on the “Mu. Ka” site can be implemented by just clicking on the link pointing to “Mu. Ka” at the bottom of the page. (See Figure 3.2.) The refined search results are shown in Figure 4.

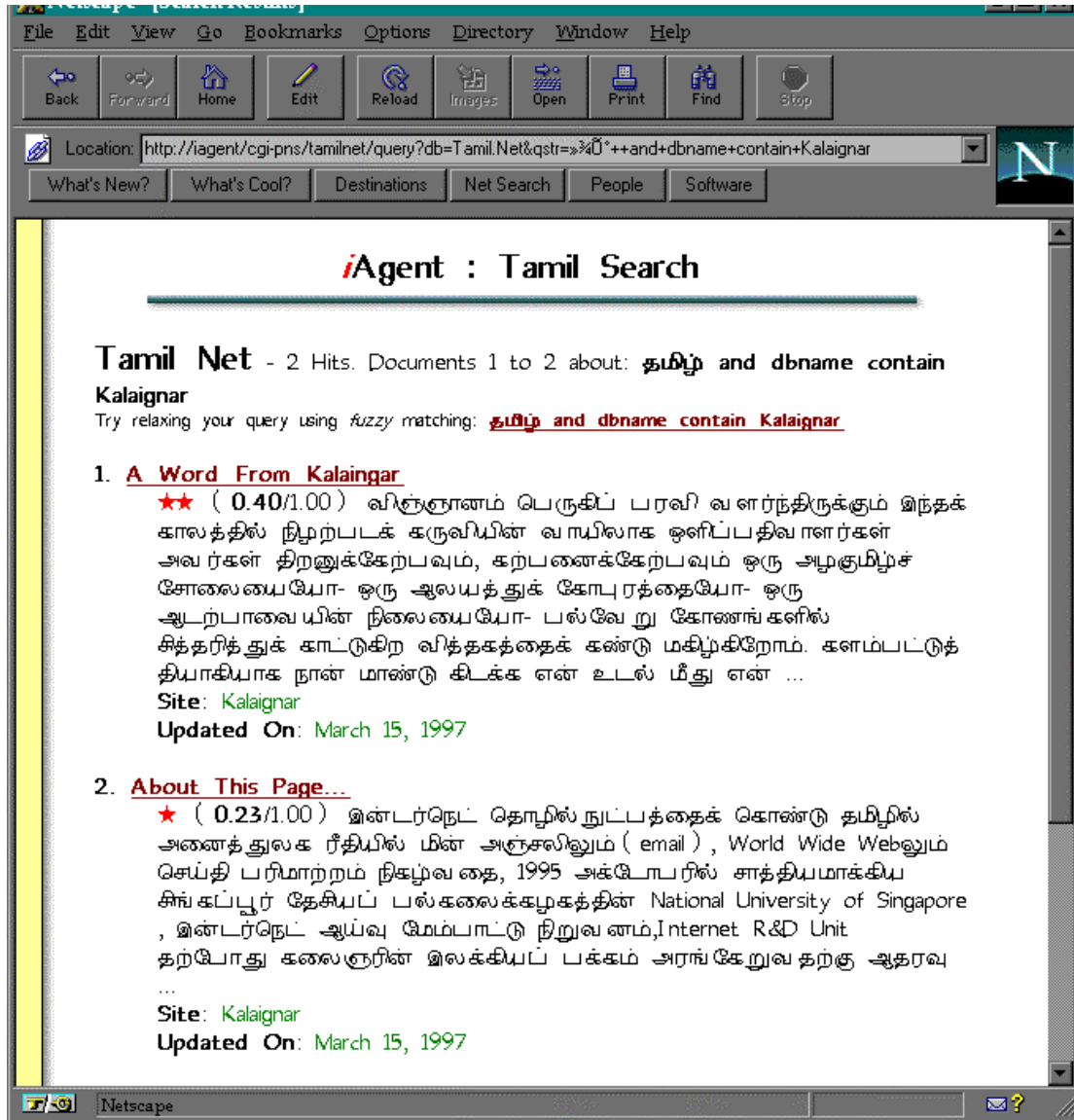


Figure 4: Results of the Focussed Search on Kalaignar's web home

#### 4. CONCLUSION

We have described *iAgent*, an information collection and retrieval system designed specifically for multilingual information resources. We have designed and built a multilingual architecture which confines language processing requirements into specialised modules. These modules share a standard library which allows the user to collect, index, retrieve and set profiles on networked information resources. We have demonstrated the usefulness of *iAgent* by showing how it can be applied for searching information resources containing Tamil web pages. The user interface has been designed to accept search strings directly in the native language. We have illustrated this by showing how Tamil can be input through the standard Tamil typewriter layout.

Ours is the first attempt at designing such a search service for networked Tamil resources. A special feature of our engine is that it is insensitive to the encoding standard used for composing the documents. The engine can support any bilingual encoding standard (i.e. the lower ASCII range is in English and the upper range in the respective language), e.g. Tamil Kavian (<http://irdu.nus.sg/tamilweb>), Murasu (<http://www.murasu.com>), Nalinam (<http://www.au.malaysia.net/tamil>). It may be noted that the same Tamil typewriter layout can be used for input under all these encodings. However, *iAgent* currently does not use a single internal representation and hence the system needs to treat each encoding separately. Our current work involves generalising the *iAgent* architecture so that multiple encoding standards can be handled under a unified formalism. We plan to adopt Unicode (<http://www.unicode.org>) as the base encoding and employ character mapping tables for handling the different encoding standards, especially for Tamil, prevalent on the WWW today. Adopting Unicode also enables future compatibility. We are also investigating encoding detection algorithms which can be deployed in *iAgent* so that documents can be searched without even knowing the encoding standard of the documents.

With the explosive growth of the WWW, more and more webpages will be published in Asian languages in the near future. A multilingual search engine like *iAgent* fulfills an important educational role for users to search for relevant information they need. The provision of search tools like *iAgent* is a first step towards supporting users in their information needs. Users also need to know how to use them effectively. For information resources in the English language, learners need to learn the skills and practice of information search such as how to refine their search based on the current hit-list<sup>1</sup>. The advent of search engines for Tamil and multilingual information resources may also bring about the need for different user search strategies for looking up information in each different language.

## 5. REFERENCES

- [1] Oren Etzioni, "The World-Wide Web: Quagmire or Gold Mine?", *Communications of the ACM*, Vol 39, No. 11, 1996.
- [2] *Search Engines*. URL: [http://www.yahoo.com/Computers\\_and\\_Internet/Internet/](http://www.yahoo.com/Computers_and_Internet/Internet/)

---

<sup>1</sup> Or for example, why a query returns 0 hit, but when slightly amended can return over 1,000 hits.

World\_Wide\_Web/Searching\_the\_Web/Search\_Engines/

[3] *Theni Search*. URL: <http://www.tamilweb.com/theni/>

[4] Leong Kok Yong, Tan Tin Wee, Naa Govindasamy, and Lee Teck Chee, "Multiple language support over the Word Wide Web", INET'96 Workshop, Montreal, Canada, 1996.

URL: <http://www.irdu.nus.sg/tamilweb/inet96/inet96paper.html>

[5] G.Salton, *Automatic Text Processing*, Addison-Wesley, 1989.

[6] William Frakes and Ricardo Baeza-Yates, *Information Retrieval : Data Structures and Algorithms*, Prentice Hall, 1992.

[7] M. F. Porter, "An Algorithm for Suffix Stripping," *Program*, V14, pp. 130-137, 1980.

[8] N. Govindasamy, "Kanian Keyboard", Tamil and Computer Conference Proceedings, Anna University, Madras, India, August 1994.

## Authors

Dr K. Rajaraman obtained his Ph.D. from the Indian Institute of Science, Bangalore, India in 1997. His research interests include Information Retrieval, Machine Learning, Data Mining and Intelligent Agents. He is also interested in prompting Tamil on the Internet. He is currently a Member of Technical Staff at the Advanced Technology (Publishing) Group in the Information Technology Institute, Singapore.

Email: [kanagasa@iti.gov.sg](mailto:kanagasa@iti.gov.sg)

Webpage URL: <http://www.iti.gov.sg/staff/kanagasa>

Dr Lai Kok Fung currently manages the Advanced Technology (Publishing) group in the Information Technology Institute, Singapore. He is one of the creators of *iAgent*, a multilingual Internet search engine supporting major Asian languages. Dr Lai obtained his Ph.D. from the University of Wisconsin - Madison in 1994. His research interests include information retrieval, pattern recognition and computer vision.

Email: [kflai@iti.gov.sg](mailto:kflai@iti.gov.sg)

Webpage URL: <http://www.iti.gov.sg/staff/kflai>