Dictionary-less Search Engine for the Collaborative Database

Virach Sornlertlamvanich[1], Pongtai Tarsaku[2], Prapass Srichaivattana[2], Thatsanee Charoenporn[1] and Hitoshi Isahara[1]

[1]Thai Computational Linguistics Laboratory, CRL Asia Research Center, 112 Phahon Yothin Rd., Klong 1, Klong Luang, Pathumthani 12120, Thailand
Email: {virach, thatsanee}@crl-asia.org, isahara@crl.go.jp

[2]National Electronics and Computer Technology Center, Thailand Science Park, 112 Phahon Yothin Rd., Klong 1, Klong Luang, Pathumthani 12120, Thailand
Email: {ptarsaku, prapass_s}@notes.nectec.or.th

Abstract

This paper presents the probabilistic-based dictionary-less search engine, called Sansarn (means document pick-up in Thai), which is different from the traditional word-based indexing search engine. It is designed to solve the ambiguous word boundary problem of non-word break languages, such as Thai, Japanese and Chinese. Without relying on the word segmentation program in word indexing process, the approach adopts the mutual information (MI) in determining the word possibility for ranking the retrieved documents. It is effectively applied in constructing the collaborative database for the contents of Thailand research project, The Best and the Brightest (B&B). The Dublin Core Metadata is introduced to facilitate data exchange between databases and to realize sophisticated search.

1. Introduction

A search engine is an important tool for the Internet world. Tons of data are available on the World Wide Web (WWW). The search engine serves people by allowing them to access the desired data easily and quickly. Nowadays, there are a lot of efficient search engines on the Web such as Google, Yahoo, MSN, Altavista, etc. These search engines consider a lot of data around the world which are often in many different languages. Thus a good search engine should support all languages. Developing a multilingual search engine is an expensive task especially when dealing with non-word break languages such as Thai, Chinese, Japanese, etc. Specifying a word boundary in non-word break language is a crucial task. For Thai, there have been a lot of approaches proposed to solve the word segmentation problem, as discussed in [4], [6], [8], and [13]. Most of them are dictionary-based approaches. In search engine viewpoint, there are two major disadvantages in the dictionary-based approach, namely the language dependency and the unregistered word manipulation. Concerning the language dependency problem, the word segmentation strongly depends on language information because it requires dictionary and some linguistic knowledge of the language. On the other hand, the unregistered word problem is crucial because new coined words regularly appear throughout the Web messages. The number of unregistered words in the Web data has a great effect on the performance of word segmentation algorithms. Consequently, the performance of the dictionary-based search engine is directly affected by the accuracy of the word segmentation module. Some dictionary-based search engines for non-word break languages are currently available in Google, Siamguru, Catcha, etc.

Previous works on dictionary-less word segmentation for Thai have been done in [5], [9], [10], [11], and [12]. These approaches use some local context statistical values such as character pair probability, left/right condition probabilities, mutual information, n-gram, left/right entropies, left/right variations as the input attributes for the decision tree algorithm. The derived decision tree is then used to specify word boundary. Sornlertlamvanich and Potipiti [9] proposed the algorithm for word extraction from Thai texts without relying on word segmentation at all. They employed the C4.5 machine learning algorithm in selecting the appropriate features for yielding the word candidates. They used several features such as string length, mutual information and entropy for training decision tree. We apply this method to our proposed search engine framework.

The remainder of this paper is organized as follows. We first review the traditional dictionary-based search engine (In Section 2). In section 3, we introduce our approach. Section 3.1 describes the word score measurement, which is the most important part of our approach. Section 3.2 describes our dictionary-less search engine. Section 4 shows the analysis result of comparison between dictionary-based and dictionary-less approaches. The application of the search engine in constructing the collaborative database for the contents of Thailand research project is explained in section 5. Section 6 gives the conclusion and future work.

2. Dictionary-based Search Engine

For a non-word break language, most search engines adopt word segmentation modules to determine the word entry for making indexes. A word segmentation module is used to specify word boundaries. Then a word list is extracted to generate the indexes. The word list is a list of all words and their information (document ID, position in a document). During the searching process, a query string is segmented into words by the word segmentation module. Then the segmented words are used to search in the index file. When the segmented words from the query string are found, the word information and the links to the target documents are retrieved. The resulting documents are finally ranked according to the predefined scoring scheme. The typical processes of the dictionary-based search engine are shown in Figure 1.

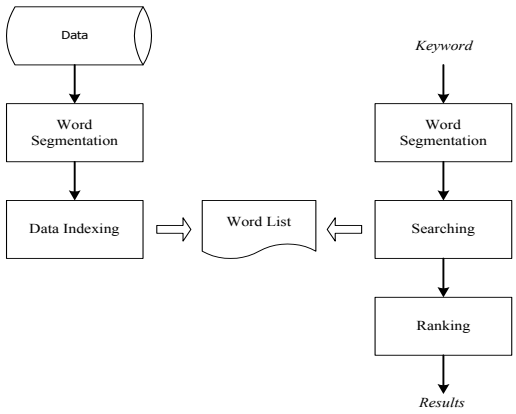

Figure 1. Dictionary-based Search Engine

The accuracy of the word segmentation module directly affects the performance of the search engine. In Thai (a non-word break language), the accuracy of the word segmentation module varies according to the number of the unregistered words in a data set. The performance of the search engine mainly depends on two processes, namely, word segmentation and document ranking.

The following examples show some possible errors being caused by the word segmentation.

Assuming that a dictionary contains 6 words: ***a, b, c, ac, bc*** and ***cb***.

Case 1:

The content of document A is ***abcbcb***. By using a word segmentation module, the content is segmented into ***a|bc|bc|b*** (assuming that the correct segmentation is ***a|b|cb|cb***). In this case, if the query is ***cb,*** it cannot be found in document A or if the query is ***bc***, the document A will be incorrectly returned.

Case 2:

The content of document A is ***abcdac***. By using the word segmentation module, the content is segmented into ***a|bc|d|ac|*** (assuming that the correct segmentation is ***a|b|cd|ac|*** and ***cd*** is an unregistered word to the word segmentation). If the query is ***bc***, the result from this document will be incorrect or if the query is ***cd***, it cannot be found in the document A.

These problems apparently decrease the accuracy of the dictionary-based search engine. In case 2, the unregistered word problem frequently occurs because the Web documents have a tendency to include proper nouns, abbreviations, and newly introduced terms.

3. Our approach in Sansarn

In the search engine for a non-word break language, we improve the accuracy of search engine in two levels, namely word level and document level. In this work, we focus on the improvement of word level. Our approach is different from the conventional approaches in that the word segmentation does not get involved at all. We determine word boundaries by adopting the statistical approach proposed by Sornlertlamvanich and Potipiti [9]. We provide a function to measure the word possibility to elaborate the word frequency. The enhanced suffix array [14] is applied to index the documents since it provides efficiency in term frequency computation for determining the word boundary. Based on the word possibility, the word score is measured as described in Section 3.1.

3.1 Word score measurement

A major problem of the non-dictionary based search engine is how to specify a word boundary. During searching process, the found strings can be located and only the meaningful strings are preferred. For example, in Thai, if a query is a short string and likely to be a part of other strings such as "ยา" *(drug),* it is difficult to locate the meaningful position. If it is contained in the string "กินยา" (*take drug*) then "ยา" (*drug*) is a meaningful string. If it is

contained in the string "พัทยา" (*Pattaya*) then it is exactly a part of the word ("พัทยา" (*Pattaya*) is a name of district in Thailand). In the given example, the meaningfulness of the string can be decided by its surrounding, namely, "กิน" (*take*) and "พัท" (*Pat: no meaning*).

Mutual information [2] is a statistical value that we use to measure the degree of the co-occurrence of the query and its context. Let ***xy*** be a query, ***ab*** is the left context and ***cd*** is the right context of the string ***xy***, the mutual information (MI) can be determined by Equation 1-4.

$$MI_L(abxy) = \frac{p(abxy)}{p(ab)p(xy)} \quad (1)$$

$$MI_L(abxy) \approx \frac{Count(abxy)}{Count(ab)Count(xy)} \quad (2)$$

$$MI_R(xycd) = \frac{p(xycd)}{p(xy)p(cd)} \quad (3)$$

$$MI_R(xycd) \approx \frac{Count(xycd)}{Count(xy)Count(cd)} \quad (4)$$

If the *MI* value is high, ***xy*** should be a part of the context. In other words, the word score of ***xy*** is low. On the other hand, if the MI value is low, ***xy*** should be independent from the context. In other words, the word score of ***xy*** is high. We can measure the word score of a string from the *MI* by Equation 5-6.

$$wscore_L(xy \mid ab) = 1 - norm(MI_L(abxy)) \quad (5)$$

$$wscore_R(xy \mid cd) = 1 - norm(MI_R(xycd)) \quad (6)$$

norm (.) is the normalizing function which normalizes the argument from 0 to 1.

We use the word score (wscore) to determine the word boundary. The word score assigns the possibility of being a word of the string. Thus the word score is a value that represents the word possibility. In addition, the word score is also beneficially used in weighting the term in different contexts. For example, the word score of the query "พลาสติก" (*plastic*) in the string "ถุงพลาสติก" (*plastic bag*) is lower than the one of the string in "ทำจากพลาสติก" (*made from plastic*). In the above example, the word score can be used to weigh the term in a more precise manner. Though "พลาสติก" (*plastic*) in both cases are acceptable word, "type of bag" in the first case is likely to become a part of a compound word rather than "material" in the second case. The word score is very useful in refining the document ranking process.

3.2 Dictionary-less Search Engine

In non-word break languages, determining the word boundary is a crucial task. Most of the search engines for non-word break languages rely on the word segmentation module to solve the word boundary problem. To overcome the word boundary problem, we propose the word score for measuring the word possibility of the string in the context.

The architecture of dictionary-less search engine is shown in Figure 2.

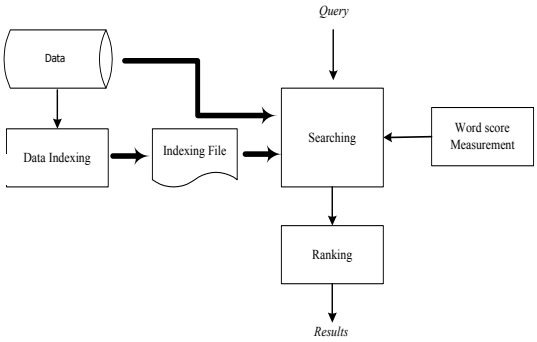

Figure 2. Architecture of the dictionary-less search engine

The search engine is composed of 3 major modules.

- Data Indexing
- Searching
- Document ranking

3.2.1 Data Indexing

In typical search engines, the data are segmented into words to provide a word list for generating the indexes. In our approach, the data is referred to the sequence of characters and indexed character by character. We adopt the enhanced suffix array for indexing the data. All suffixes of the data string are indexed. Thus the number of indexes is equal to the data size. The advantage of this indexing method guarantees all search strings to be found while word indexing method depending on the word segmentation yields a lower recall. This indexing method can be applied to all languages because no dictionary and language depending knowledge are required.

3.2.2 Searching

The searching method is based on the enhanced suffix array. The method requires O(mlogN) to access the string in the data, where m is a length of the search string (or query) and N is the number of indexes. The word score is used to measure the word possibility of the found string (discussed in Section 3.1). For each found string, the word score is calculated for both of its left and the right contexts. If the found string and its context is "***…abxyzcd…***", where "***xy***" is the search string, "***ab***" and

"***cd***" are the left and right contexts respectively. We calculate two word score values, namely, $wscore_L(xy|ab)$ and $wscore_R(xy|cd)$ following Equations 5 and 6. The minimum value of the two word score is assigned to be the word score of the string "***xy***". This procedure is shown in Figure 3.

```
do all location of found string
    left_context = get left context from location(i)
    right_context = get right context from location(i)
    lscore = wscore(keyword|left_context)
    rscore = wscore(keyword|right_context)
   awscore[i]  = min(lscore,rscore)
   increase i
loop
```

Figure 3. **Pseudo codes of the word score calculation**

The *awscore[]* represents the term weight in document ranking process.

3.2.3 Document Ranking

We use *tf***idf* [7] based method for ranking the document. The term frequency (*tf*) is defined in Equation 7.

$$tf(i) = \sum_{all\ index_i} awscore\ [index_i] \qquad (7)$$

where $index_i$ is the index of *awscore[]* which is a word score from document *i*.

The value of being a term in our approach is varied from 0 to 1 while terms in general is defined according to their existence. Document frequency (*df*) of a term is defined as the number of documents that contains the term. We can compute the inverted document frequency (idf) by the Equation 8.

$$idf = \log(\frac{N}{df}) \qquad (8)$$

where N is the number of all documents in the data set.

4. Analysis

Considering the drawbacks of the dictionary-based approach,

- language dependency problem
- unregistered word problem

can be comparatively analyzed with our approach as described below:

4.1 Language dependency problem

This is a major drawback of dictionary-based approach because the search engine needs word segmentation and dictionary for each language. In general, word segmentation requires a dictionary and some linguistic knowledge. Considering the multilingual ability of the search engine, our approach can better fulfill the task and be extensible to any language.

4.2. Unregistered word problem

The accuracy of the word segmentation module is decreased corresponding to the increasing number of unregistered words in the database. In this case, the word segmentation module tries to segment the unregistered word with the registered word. If the search engine uses these segmented words to create the index, the undesirable results will be returned. For example, in case of proper noun, the word "ประชาอุทิศ" (*Pracha uthit*) which is an unregistered word (this proper noun is composed of two registered common words, namely, "ประชา" (*Pracha, people*), "อุทิศ" (*dedicate*)), the error occurs when the query is either "ประชา" or "อุทิศ". The string "ประชาอุทิศ" will be undesirably returned. In case of phrases, the phrase "ที่นาซ่า" (*at NASA*) composes of two word "ที่" (*at*), a preposition, and "นาซ่า" (*NASA*), an unregistered word. Accidentally this phrase is segmented into "ที่นา" (*farmland*) and "ซ่า" (*show of*) which are undesirable. To improve the accuracy, the dictionary needs to be timely up-to-date. In our approach, the word score of the search string is computed. In the first example, the word score of the search string "ประชา" with context "อุทิศ" and "พูด" (*speak*) are comparatively computed. Word score in Equation 5 and 6 can naturally reflect the dependence of "ประชา" in "ประชาอุทิศ" rather than in "ประชาพูด" context. Since "พูด" is a common word and "ประชาอุทิศ" is a compound word. The common word "พูด" is likely to appear with other words than "อุทิศ" which is likely to a part of the compound word.

5. Application of Sansarn in the construction of collaborative database for the contents of Thailand research project

There are several heterogeneous research databases available on the Internet in Thailand. The databases are really constructed depending on the database system and the designed structure. The inconsistency causes difficulties in exchanging data and providing one-stop service for the thorough databases access. The requirement occurs to not only the end-users but strongly from the research planning and management points of view.

Sansarn is extendedly designed to serve such the requirements under the framework of multilingual and non-word break language accessibility. To allow the heterogeneous database systems and the distributed database maintainability, the concept of collaborative

database based on the Dublin Core Metadata standard is introduced. The preliminary experimental service of search for the contents of Thailand research project, "The Best and The Brightest", is available at http://www.nstda.or.th/ grants.

5.1 System Architecture

As shown in Figure 4, the system consists of two subsystems, namely, the database preparation subsystem and the search engine subsystem. The database preparation subsystem functions as follows:

- Collect the contents of research projects regularly from the sources that conform to the provided data format standard via the web robot program.
- Make indexes by using the Sansarn algorithm.
- Distribute the indexed database to several index servers.

The search engine subsystem functions as the following:

- Obtain queries from users.
- Request the relevant research projects to the queries simultaneously from the distributed servers.
- Rank the returned projects from the distributed servers according to the score of relevance.
- Return the results with the links to each of the original web pages.

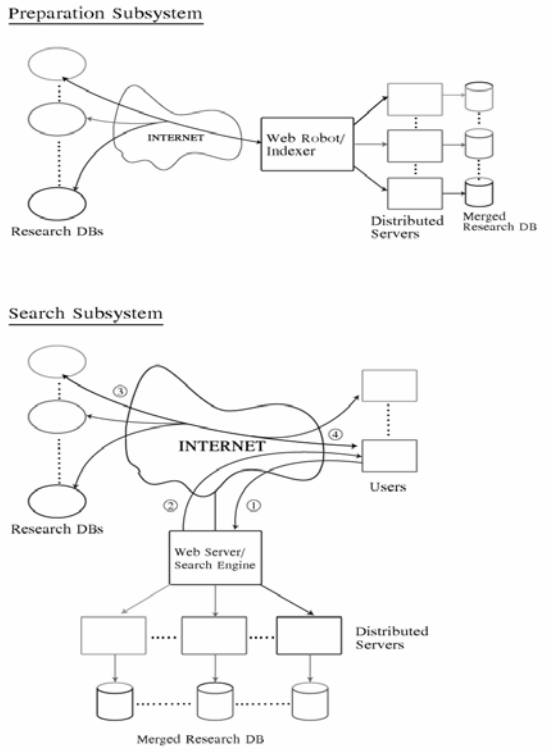

Figure 4. The System Architecture

5.2 Metadata Format

We implement eight elements of the Dublin Core Metadata Element Set V1.1 [3] as shown in Table 1. The metadata for the contents of research projects are allowed to be annotated in two forms. One is inserted at the head part of the HTML files and the other one is encoded in RDF/XML conforming to the Expressing Simple Dublin Core in RDF/XML standard [1]. The Figures 5 and 6 are the examples of the two forms used in our system.

Table 1. The Metadata Elements for the contents of research projects

Metadata Elements	Description
DC.Title	Title
DC.Creator	Researcher(s)
DC.Publisher	Institution
DC.Contributor	Source of Funds
DC.Type	Status
DC.Coverage	Beginning and ending date
DC.Date	Actual completed date
DC.Description	Abstract

```
<html>
<head>
<meta name="DC.Title" content="IPv6 Test Bed">
<meta name="DC.Creator" content="Mr.Robert Elz">
<meta name="DC.Publisher" content="Songkla University">
<meta name="DC.Contributor" content="NECTEC">
<meta name="DC.Type" content="On-going">
<meta name="DC.Coverage" content="2002-07-31 to 2003-07-30">
<meta name="DC.Date" content="-">
<meta name="DC.Description" content="The IPv6 address ……..">
</head>
```

Figure 5. Example of the metadata in the head part of HTML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
    rdf:about="http://mali.nectec.or.th/det?id=B055">
<dc:title>IPv6 Test Bed</dc:title>
<dc:creator>Mr.Robert Elz</dc:creator>
<dc:publisher>Songkla University</dc:publisher>
<dc:contributor>NECTEC</dc:contributor>
<dc:type>On-going</dc:type>
<dc:coverage>2002-07-31 to 2003-07-30</dc:coverage>
<dc:date>-</dc:date>
<dc.description>The IPv6 address ……</dc:description>
</rdf:Description>
</rdf:RDF>
```

Figure 6. Example of the metadata in RDF/XML

6. Conclusion and Future Work

We propose a new approach of dictionary-less search engine called Sansarn to solve the ambiguous word boundary of non-word break languages, e.g., Thai, Japanese, and Chinese. We implement the search engine to construct the collaborative database for the contents of Thailand research projects in order to facilitate the project of The Best and The Brightest. Sansarn introduces the Dublin Core Metadata to markup the contents for facilitating the data exchange. For future work, we are going to extend features of the existing system, i.e., facilitating the creation of Thailand researcher profile database and providing the service in searching for researchers who have expertise in the required subjects by using the technique of citation indexes in ranking the expertise.

7. References

[1] Beckett, Dave, Eric Miller and Dan Brickley. "Expressing Simple Dublin Core in RDF / XML", http://dublincore.org/documents/2002/07/31/dcmes-xml/, July 2002.

[2] Church, K.W., Robert L. and Mark L.Y., "A Status Report on ACL/DCL", Proceedings of the 7th Annual Conference of the UW Centre New OED and Text Research: Using Corpora, 1991, pp. 84-91.

[3] Dublin Core Metadata Initiative, "Dublin Core Metadata Element Set, Version 1.1: Reference Description", http://dublincore.org/documents/2003/02/04/dces/, February 2003.

[4] Meknavin S., Charoenpornsawat P. and Kijsirikul B., "Feature-based Thai Word Segmentation", Proceedings of the Natural Language Processing Pacific Rim Symposium 1997, December 1997, pp.41-46.

[5] Potipiti T., Sornlertlamvanich V., and Chaloenporn T. , "Towards Building a Corpus-based Dictionary for Non-word-boundary Languages", Workshop on Terminology Resources and Computation, Workshop Proceedings of the Second International Conference on Language Resources and Evaluation (LREC2000), May 2000, pp. 82-86.

[6] Raruenrom S., "Dictionary-based Thai Word Separation", Thesis, Engineering Faculty, Chulalongkorn University, 1991.

[7] Salton G. and McGill M.J., Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

[8] Sornlertlamvanich V., "Word Segmentation for Thai in Machine Translation System", Machine Translation, National Electronics and Computer Technology Center, Bangkok, 1993, pp. 50-56.

[9] Sornlertlamvanich V., Potipiti T. and Chaloenporn T., "Automatic Corpus-Based Thai Word Extraction with the C4.5 Learning Algorithm", Proceedings of the 18th International Conference on Computational Linguistics Vol. 2, Jul 2000, pp. 802-807.

[10] Thanaruk T., Thanasan T., Duangrumol P. and Arunthep S., "Non-Dictionary-Based Word Segmentation Using Local Context Statistics",. Proceedings of SNLP-Oriental COCOSDA 2002, May 2002, pp. 81-88.

[11] Theeramunkong T. and Usanavasin S., "Non-Dictionary-Based Word Segmentation Using Decision Tree", Proceedings of Human Language Technology (HLT 2001), March 2001.

[12] Theeramunkong T., Usanavasin S., Machomsomboon T. and Opasanont B, "Thai Word Segmentation without a Dictionary by Using Decision Tree", Proceedings of The Forth Symposium on Natural Language Processing, May 2000, pp.165-175.

[13] Varakulsiripan R., Ngamvivit J., Janwan S., Jiwattayakul S. and Thipjaksurat S., "Word Segmentation from Thai Sentence by Longest Matching Method", Papers on Natural Language Processing complied by Virach Sornlertlamvanich, 1995, pp. 279-290.

[14] Yamamoto, M. and Church, K.W., "Using Suffix Arrays to Compare Term Frequency and Document Frequency for All Substrings in Corpus", Proceedings of the Sixth Workshop on Very Large Corpora, August 1998, pp. 28-37.