BENGALI-ENGLISH RELEVANT CROSS LINGUAL INFORMATION ACCESS USING FINITE AUTOMATA

Avishek Banerjee, Swapan Bhattacharyya
Department of Computer Science & Engineering and Information Technology
Asansol Engineering College Sen Raleigh Road
Kanyapur, Asansol, 713304, WB, India
Email: avishekbanerji@gmail.com, swapanbhattacharyya@ieee.org

ABSTRACT

CLIR techniques searches unrestricted texts and typically extract term and relationships from bilingual electronic dictionaries or bilingual text collections and use them to translate query and/or document representations into a compatible set of representations with a common feature set. In this paper, we focus on dictionary-based approach by using a bilingual data dictionary with a combination to statistics-based methods to avoid the problem of ambiguity also the development of human computer interface aspects of NLP (Natural Language processing) is the approach of this paper. The intelligent web search with regional language like Bengali is depending upon two major aspect that is CLIA (Cross language information access) and NLP. In our previous work with IIT, KGP we already developed content based CLIA where content based searching in trained on Bengali Corpora with the help of Bengali data dictionary. Here we want to introduce intelligent search because to recognize the sense of meaning of a sentence and it has a better real life approach towards human computer interactions.

KEYWORDS

Query Translation,Co-OccurrenceTendency, Training Corpora, indexing, ranking, relevancy, NLP, CLIA etc.

INTRODUCTION

This paper describes a system that uses cross-language information retrieval (CLIR) methods and NLP to provide search engines with capability of automatic, relevant intelligent bilingual search. Here we want to introduce the intelligence constructing different Bengali grammar rules or semantic by automata. Bilingual Machine Readable Dictionaries is a good alternative. We have used a Bengali-English pair as an application for the conducted experiments. Here is a big issue for which we have used frequency patterns to count the relevancy avoiding exceptional cases by programming logic. Though the complete implementation of rules of Bengali Grammar is a very big job we here introduced to recognize the sense of sentences from the different forms of verb from SOV (Subject Object Verb) pattern of Parts of Speech. Here we detected that verbs are playing great role to detect the sense of a sentence so we emphasis on verb of a sentence and detection of suffix form of Verb list.

The whole process can be summarized as

Feature terms are first extracted from Web documents for each category in the source and target languages. The CLIR methods are implemented to the search sentences. After that, we use further measures to evaluate ranked retrieval results, thus, developing reliable and informative test collections. We have used frequency patterns to count the relevancy avoiding exceptional cases by programming logic. Given a variety of keyword occurrences in a document, the goal is to choose fragments, which are:

(i) Maximally informative about the discussion of those terms in the document.
(ii) Self-contained enough as to be easy to read, and
(iii) Short enough to fit within the normally strict constraints on the space available for summaries.

The system would be useful for Web users, expanding the international scope of the Web.

CLIR METHODS

CLIR methods basically includes the following steps
1) Bengali Data dictionary creation.
2) Parsing according to the parts of speech (POS).
3) Creation of stop word list.
4) Bengali Grammar rules (machine understandable) Construction.
5) Web Intelligent Searching Technique. (Sense Recognition Techniques)
6) Indexing and Ranking using relevancy.

1) Bengali Data dictionary creation.

1. Data dictionary is created mainly keeping mind to detect the words of Bengali parts of speech (like Nouns, Adjectives, Verb, and Adverb).
2. These words are included in our data dictionary separately in their combinational form of Unicode Characters (The Unicode Standard 5.0).
3. The present Data Dictionary is created in HTML files as it provide us an easy way to examine the combinational form of the Unicode Characters whether they correctly represent the ‘Bengali’ words as it is the Standard Dictionary or not.

2) Parsing according to the parts of speech.

The main reasons behind Parsing (Dividing) are:

1) There are various forms of the parts of speech in Bengali, which are required to parse so that we can detect the suffixes, prefixes and/or inflections at the end of the words and detect the stop word list.

2) Dividing the search sentences enables a structured way of look since the words are categorized into Nouns, Verbs, Adverbs and Adjectives.

3) Creation of Stop Word List.

1. The main purpose of using such stop word list is to separate the non useful words like Conjunctions, Prepositions, Articles, Suffix, and Prefix etc from the query. So that main meaningful Root words can be identified and search can be made using the selected root words.

3) Construction of rules according to Bengali Grammar

1) We detected the list of total 110 suffixes that are user after verb in Bengali and which create sense of tense in Bengali sentences.Among those 110 suffix 44 are not used in today's Bengali and called "SADHU" another 66 suffixes are used which are called "CHALIT" in Bengali grammar. Among those "CHALIT" suffixes some are most frequently used and some are used rarely.

2) There are some specific forms in Bengali to determine first, second and third person singular number or plural number and depending upon those some specific suffixes are used,similar like English grammar.The corresponding suffixes are fixed for particular subject term such as when subject is first person singular like "AMI" means I the tense form of go verb "JABO","JAI" , JACHCHI" etc are used but is not used in case of subject is second person singular like "TUMI" means you . So the suffixes forms are fixed for subject person.

3) Depending upon the different types of suffix list we can determine the relationship or mode of talking in any conversion.

ই	অ	ও	এন	ন	ইস	এ	ইতেছি	ইতেছ
ইতেছেন	ইতেছিস	ইেতেছ	ছি	চ্ছি	ছ	চ্ছ	ছেন	চ্ছেন
ছিস	চ্ছিস	ছে	চ্ছে	চ্ছেন	ইয়াছি	ইয়াছ	ইয়াছিস	ইয়াছেন
ইয়াছে	এছি	এছ	এছেন	এছিস	এছে	ইলাম	ইলে	ইলেন
ইলি	ইল	লাম	লে	লেন	লি	ল	লেম	লুম
ইতেছিলাম	ইতেছিলে	ইতেছিলি	ইতেছিল	ইতেছিলেন	ছিলাম	ছিলে	ছিলি	ছিল
ছিলেন	চ্ছিলি	চ্ছিল	চ্ছিলেম	ছিলেম	ছিলুম	ইয়াছিলাম	ইয়াছিলে	ইয়াছিলি
ইয়াছিল	ইয়াছিলেন	ইয়াছিলেম	এছিলাম	এছিলে	এছিলি	এছিল	এছিলেন	এছিলেম
এছিলুম	ইতে	ইতাম	ইতেন	ইত	তাম	তেম	তুম	তে
তেন	ত	তিস	ইব	ইবে	ইবেন	ইবি	ব	বে
বেন	বি	ইতেথাকিব	ইতেথাকিবে	ইতেথাকিবেন	ইতেথাকিবি	তেথাকব	তেথাকবে	তেথাকবেন
তেথাকবি	ইয়াথাকিব	ইয়াথাকিবে	ইয়াথাকিবেন	ইয়াথাকিবি	এথাকব	এথাকবে	এথাকবেন	এথাকবি

Suffix list of verb in Bengali

অ	অনা	আ	দর	কু	নির	নি	সু
না	পাতি	বি	ভর	ভরা	রাম	স	হা

Prefix list of noun, adjective

প্র	পরা	অপ	সম	নি	অব	অনু	নির	দুর	বি
অধি	সু	উৎ	পরি	প্রতি	অভি	অতি	অপি	উপ	আ

Bengali prefix list (comes from "Sanskrit")

অন্তঃ	আবিঃ	বহিঃ	প্রাদুঃ	তিরঃ	পুনঃ	পূর্ব

Prefixes of some indeclinable-words

ফি	দর	হর	না	ব	বে
বদ	নিম	বর	কার	খাস	খোশ

Bengali prefix list (comes from "Pharsi")

হেড	ফুল	হাফ	সাব

Bengali prefix list (comes from "English")

টি	টা	খানা	খানি	গুলো
গুলি	রাশি	রাজি	বৃন্দ	গণ
সমূহ	আবলী	জাতা	জাতি	

Bengali suffix list of noun (based on "Numbers").

এর	রা	টার	কে
এ:	তে	ও	ই
র	এরা		

Bengali suffix list of noun/adjective

এ:+রা + ই	এ:+রা + ও	এ:+ র + ই
এ:+ টার	এ:+ ও	এ:+রা
এ:+ কে	এ:+ কে + ও	এ:+ র + ও
কে + ও	কে + ই	রা + ও

Bengali concatenated suffix list of noun/adjective

Steps for Parts Of Speech (POS)

i) Go through the search quarry.

ii) Decide types the word whether Noun, Verb, Adverb, Adjective or Pronoun from the dictionary format.

iii) The Noun, Verb and Adjective will be treated as keywords.

iv) From the suffix list of verb we recognize the sense of sentences.

v) Rather than Noun, Adjective, Adverb and Verb all words are treated as Stop Word.

We design a finite state Automata which is described below to detect the sense of a particular sentence in Bengali Language.

Finite Automata :

- Σ is the input alphabet (a finite, non-empty set of symbols).
- S is a finite, non-empty set of states.
- s_0 is an initial state, an element of S.
- δ is the state-transition function: .
- F is the set of final states, a (possibly empty) subset of S.

➢ $\Sigma \rightarrow \{a,b,c,d,\ldots\}$

Where ,

a → Checking condition whether contaminated with some specific suffix term in verb.

b → Checking condition whether contaminated with some specific suffix term in verb.

c → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in first person.**

d → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in second person .**

e→ Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in third person.**

e→ Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in third person.**

f → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in first person present tense form.**

g → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in first person past tense form.**

h→ Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in first person future tense form.**

i→ Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in second person present tense form.**

j → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in second person past tense form.**

k → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in second person future tense form.**

l → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in third person present tense form.**

m → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in third person past tense form.**

n → Checking condition whether contaminated with some specific suffix term in verb and determine the **subject set in third person future tense form.**

➢ $S \rightarrow \{s_0, S_{1,}, S_2, C, D\}$

S_1 – "SADHU" form of Bengali language ,

S_2 – "CHALIT" form of Bengali Language ,

S_3 – Verb form used with **subject set in first person.**

S_4 – Verb form used with **subject set in Second person.**

S_5 – Verb form used with **subject set in Third person.**

S_6 – Verb form used with **subject set in first person present tense form.**

S_7 – Verb form used with **subject set in first person past tense form.**

S_8 – Verb form used with **subject set in first person future tense form.**

S_9 – Verb form used with **subject set in second person present tense form.**

S_{10} – Verb form used with **subject set in second person past tense form.**

S_{11} – Verb form used with **subject set in second person future tense form.**

S_{12} – Verb form used with **subject set in Third person future tense form.**

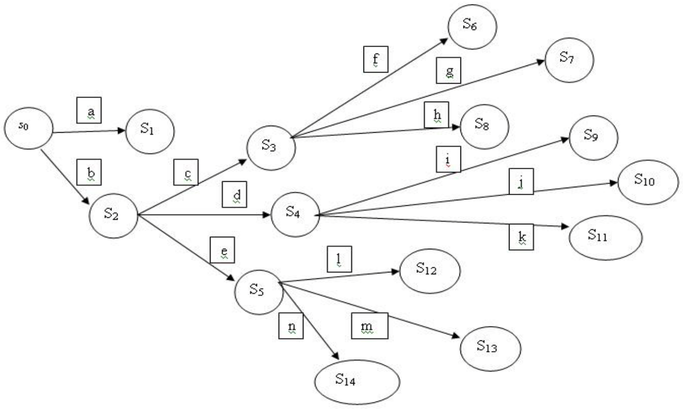
S_{13} – Verb form used with **subject set in Third person future tense form.**

S_{14} – Verb form used with **subject set in Third person future tense form.**

➢ δ **:** $S X \Sigma \rightarrow S$

➢ s_0 → {Bengali sentence in SOV pattern}

➢ F_0 → {non empty set of final states }



subject set →
{{AMI,AMRA,AMADER},{TUMI,TOMRA,TOMADER},
{SE,TARA,TADER}}

5. Searching Technique

1) The retrieved keywords are matched with the keyword field of database table and corresponding multiple links is retrieved from the database table.

2) The link is referred to the particular HTML file address of the repository and gets the document retrieved from the repository.

6. Indexing and Ranking using Relevancy

1) The retrieved documents are indexed first with out relevancy checking.
2) The indexed documents are ranked according to their relevancy.

IMPLEMENTATION

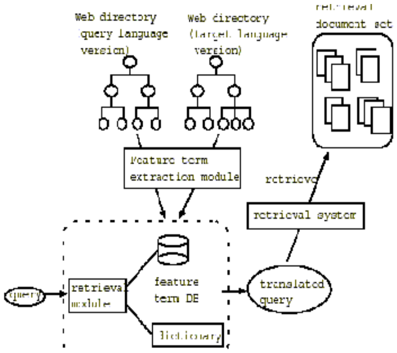


Figure 1. Outline of the proposed system.

Figure 1 illustrates the outline of the proposed system. This system consists of query and target language versions of Web Directory, each language versions of feature term database, bilingual dictionary, and retrieval target document set. The part surrounded by a dotted line illustrates components of translation processing for query. The processing on our system can be divided into two phases. One is the **preprocessing phase**, which extracts feature terms from each category of a Web directory, and stores it in the feature term database in advance. Another is the **retrieval phase**, which translates the given query into the target language, and retrieves documents.

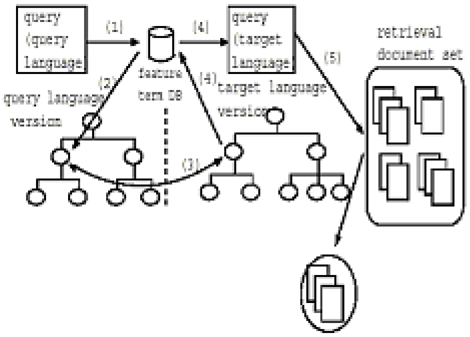


Figure 2. Flow of retrieval.

a. Preprocessing Phase

Figure 2 shows the processing flow of the proposed system. In the preprocessing phase, the system conducts feature term extraction and category matching between the query and target languages in advance. The following procedure is used for this phase:

Feature Term Selection:

The feature term selection is very crucial to the document representations. The main goal of feature selection is to enhance the performance of system, to minimize the use of raw feature and to reduce the dimensions of document representation space. Given a collection of training documents, all unique terms found in the collection obtained after stop-words removal and stemming words are too large to be applied directly to learning algorithm. In addition, the experimental results have shown that using all terms could not produce the satisfactory performance. So the irrelevant and redundant terms must be removed and an optimal feature term subset must be selected. The algorithms about feature selection have been investigated extensively. Sometimes, the relation between feature terms needs be analyzed and is used to reduce the dimension of feature term space. This analysis can be done by domain-specific expert or automatic technique for example term clustering [7].

In case of English language we can say that the heuristics used to select the candidate feature terms identify base noun phrases according to the following patterns:

Base Noun Phrase (BNP). This pattern restricts the candidate feature terms to one of the following patterns: NN, NN NN, JJ NN, NN NN NN, JJ NN NN, JJ JJ NN, where NN and JJ are nouns and adjectives. Definite Base Noun Phrase (dBNP). This pattern restricts candidate feature terms to definite base noun phrases, which are noun phrases (BNP) preceded by the definite article the. But in case of Bengali language we know that there is no concept of Article so for this language we can take help of another set of suffixes like “ta”,”ta ke” , “gulo” etc. Also we have to take the help of the prefix list of the Noun and Adjective to determine the sense of the sentences.

Definite Base Noun Phrase (dBNP). This pattern restricts candidate feature terms to definite base noun phrases, which are noun phrases (BNP) preceded by the definite article the.

Beginning Definite Base Noun Phrase (bBNP). bBNPs are dBNPs at the beginning of a sentence followed by a verb phrase.

1. **Feature-term extraction**

For each category in all language versions of a Web directory,

(a) Extract terms from Web documents in the required category and calculate the weight of the terms.

(b) Extract the top *n* ranked terms as the feature terms of the category.

(c) Store the feature terms in the feature-term database.

The features of each category are represented in the feature-term set. The feature-term set is a set of terms that are judged to represent the features of the category. The feature-term set for each category is extracted as follows:

(1) The system extracts terms from Web documents that belong to a given category;

(2) The system calculates the weights of the extracted terms.

(3) The top *n* ranked terms are extracted as the feature terms of the category.

Weights of feature terms are calculated by TF·ICF (term frequency · inverse category frequency). TF·ICF is a variation of TF·IDF (term frequency · inverse document frequency). TF·IDF is calculated by multiplying the term frequency by the inverse document frequency. Instead of using a document as the unit, TF·ICF calculates weights by category. TF·ICF is able to calculate term weights considering the content of the category. It is calculated as follows:

$tf \cdot icf(ti, c) = f(ti)\ Nc \cdot \log N\ ni + 1$

Where *ti* is the term appearing in category *c*, *f*(*ti*) is the term frequency of term *ti*, *Nc* is the total number of terms in category *c*, *ni* is the number of categories that contain the term *ti* and *N* is the total number of categories in the directory.

b. Retrieval Phase

Figure 2 illustrates the processing flow for retrieval. First, the system estimates the relevant category of the query from the query language version. Secondly, the system selects a category corresponding to the relevant category. Thirdly, the system translates the query terms into the target language using the feature term set for the corresponding category. Finally, the system retrieves documents using the translated query. The procedure for the retrieval phase is as follows:

(1) For each category in the query language version, calculate the relevance between the query and the feature term set for the category.

(2) Determine the category with the highest relevance as the relevant category for the query.

(3) Select the category corresponding to the most relevant category from the target language version.

(4) Translate the query terms into the target language using the feature-term set of the corresponding category.

(5) Retrieve documents using the translated query.

Selection of Relevant Category

In our system, queries consist of keywords, not sentences. We define the query vector *_q* as follows:

$\_q = (q1, q2, \ldots, qn)$

where *qk* is the weight of the *k*-th keyword in the query.We define the values of all *qk* as 1. First, the system calculates the relevance between the query and each category in the query language version, and determines the most relevant category to the query in the query language version. The relevance between the query and each category is calculated by multiplying the inner product between the query terms and the feature-term set of the target category by the angle of these two vectors. The relevance between query *q* and category *c* is calculated as follows:

$rel(q, c) = \_q \cdot \_c\ ((\_q \cdot \_c) / (|\_q| \cdot |\_c|))$

Where *_c* is a vector of category *c* and is defined as follows:

$\_c = (w1, w2, \ldots, wn)$

where *wk* is the weight of the *k*-th keyword in the feature term set of *c*.

When there is more than one category whose relevance to the query exceeds a certain threshold, all are selected as relevant categories for the query.

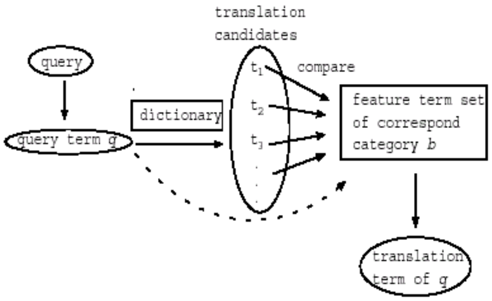
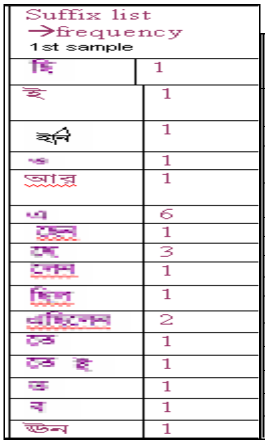


Figure 3. Translation of a query.

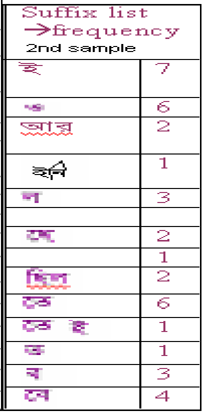
Query Translation

Figure 3 illustrates the processing flow for query translation. First, for each query term *q*, the system looks up the term in a bilingual dictionary and extracts all translation candidates for the feature term. Next, the system checks whether each translation candidate is included in the feature-term set of the corresponding category. If it is, the system checks the weight of the candidate in the feature-term set. Lastly, the highest-weighted translation candidate in the feature-term set of the corresponding category is selected as the translation of the feature term. If there is no translation candidate for a feature term in the feature-term set of the corresponding category, that term is ignored in the retrieval. However, in some cases, the source language term itself is useful as a feature term in the target language. For example, some English terms (mostly abbreviations) are commonly used in documents written in other languages (e.g. “WWW”, “HTML”). Therefore, when there is no translation candidate for a feature term in the feature-term set of the corresponding category, the feature term itself is checked to see whether it is included in the feature-term set of the corresponding category. If it is, the feature term itself is treated as the translated term. In the next section, we propose some revisions of the query translation method described above.

Results



Suffix list →frequency 1st sample	
ছি	1
ই	1
ইনি	1
ও	1
আর	1
এ	6
ছেন	1
ছে	3
লেন	1
ছিল	1
এছিলেন	2
তে	1
তে ই	1
ত	1
ব	1
উডেন	1



Suffix list →frequency 2nd sample	
ই	7
ও	6
আর	2
ইনি	1
ল	3
ছে	2
	1
ছিল	2
তে	6
তে ই	1
ত	1
ব	3
লে	4

DHATU → Ll

1st Sample →2044 words (in chalit form)

2nd Sample → 4087 words (in chalit form)

Total sadhu suffix of verb→ 44

Total chalit suffix of verb → 66.

Steps to extract sense of Bangla sentence:

We consider the following example sentence

আমি গবেষনা করি

i)The context of this sentence can be judged by developing a rule for the verb and its parts of speech. We can see that the word 'করি' is a form of the word 'কর' which implies doing something. However as in Bangla language, the grammar rules specify that the suffix determines the subject. As in this case the subject would be 1st person. With this information in mind and with the aim of transforming specifically Bangla words to a machine understandable code, we can see that only the following words can imply 1st person subject.আমি, আমরা, আমার, আমাদের, etc.We can do a context search on this list to see which state the subject is in. We can also understand if it implies a singular or plural form using the suffix information of 1st person subject in Bangla language. Since we have identified the subject here it is but trivial to understand that ' গবেষনা ' is the object of this sentence. The thing we have done is divided the sentence into fields which can be represented by states in a finite state automata.Now the task is to understand the tense and the parts of speech of the sentence. Again the suffix information can be used here. Consider this, the word ‘করেছিলাম’is in past tense because of the suffix, ‘ছিলাম’, and this form is similar in other verbs too. Thus we have deciphered the tense of the sentence by examining the verb. To understand the parts of speech we can again look into the suffix of the word for Bangla language. For example the word 'করছিস' with the suffix 'ছিস' indicates an interrogative sentence. An assertive sentence however is like example (i) which has a 'ই' suffix. So here is the short classification in the form of various states.

CONCLUSION

As Bengali is a well-known and renowned language and Bengali versed population in our country is very high if we can facilitate cross language translation from Bengali to English and vise versa we can achieve to the goal of information searching by religion language and motivate general people toward the use of Information Technology.

FUTURE SCOPE

Relevancy should be a highly sensitive issue in this regard so if we concentrate in this regard and work with large corpora then the system will be checked or tested or verified but for that we need financial help from Government or Corporate House who are interested in this field.

REFERENCES

[1] Stephen Robertson, “TREC Techniques for Cross-language Information Retrieval”, in *Proceeding of the International Workshop on Research Issues in Digital Libraries in co-operation with ACM SIGIR at Vedic Village,Kolkata,*on December 2006.

[2] D. A. Hull. Using structured queries for disambiguation in cross-language information retrieval. *Electronic Working Notes of the AAAI Symposium on Cross-Language Text and Speech Retrieval*, 1997.

[3] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real user queries on the Web. *Information Processing & Management*, 36(2):207–227, 2000.

[4] F. Kimura, A. Maeda, J. Miyazaki, M. Yoshikawa, and S. Uemura. Cross-Language Information Retrieval Using Web Directory as a Linguistic Resource. *Proceedings of Asia Information Retrieval Symposium (AIRS)*, pages 297–300, 2004.

[5] F. Kimura, A. Maeda, M. Yoshikawa, and S. Uemura. Cross-Language Information Retrieval using Web Directories. *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM ’03)*, pages911–914, 2003.

[6] C.-J. Lin, W.-C. Lin, G.-W. Bian, and H.-H. Chen. Description of the NTU Japanese-English cross-lingual information retrieval system used for NTCIR workshop. *First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 145–148, 1999.

[7] Keywords Extraction, Document Similarity and Categorization , Huaizhong KOU and Georges Gardarin PRiSMLab,Universitof Versailles .