Iterative Disambiguation Algorithm for Removing Ambiguity Problem in CLIR

[1.]Aparna Joshi,[2.] Rupali bagate,[3.] Savita mangalore
[1,2,3.] Asst. Prof. AIT, Pune

***Abstract*— This paper describes an English-Marathi Cross-Lingual Information Retrieval System. The system retrieves Marathi documents in response to a query given in English. Marathi is one of the most prominent regional languages of Indian subcontinent. We take a document translation based approach using bi-lingual dictionaries. Query words not found in the dictionary are transliterated using a simple rule based approach which utilizes the corpus to return the 'k' closest Marathi transliterations of the given English word. The resulting multiple translation/transliteration choices for each query word are disambiguated using an iterative disambiguation algorithm which, based on term-term co-occurrence statistics, produces the final translated query.**

I. INTRODUCTION

Information Retrieval (IR) systems aim to retrieve relevant documents to a user query, where the query is a set of keywords. Cross Lingual Information Retrieval (CLIR) involves the retrieval of documents in a language other than the query language.Since the language of query and the documents to be retrieved are different, the queries need to be translated. This translation step may cause reduction in the retrieval performance of CLIR system as compared to monolingual IR system. The main reasons for this reduced performance are missing specified vocabulary, missing general terms and wrong translation due to ambiguity [1].
The three main approaches of query translation include dictionary-based machine translation; parallel corpora based statistical lexicon and ontology-based methods [2]. The basic idea of machine translation is to replace each term in the document with an appropriate term or a set of terms from the lexicon. This approach is used in our experiment. Our cross-lingual task involves Marathi document retrieval in response to queries in English. All major news papers, Government departments, public sector organizations have started websites in Marathi language. CLIR helps to break the barrier of languages and helps to access information in different languages.

The organization of the paper is as follows: Section II, gives the literature survey of CLIR system. Section III, explains the architecture of our CLIR system. Section IV, describes the experiment and discuss the result. Section V, concludes the paper highlighting some potential direction for future work.

II. LITERATURE SURVEY

In Indian languages CLIR is still in its primitive state. CLIR works have been reported for Chinese-English [6], Arabic- English [5] and European languages like German-English [12], French-English [13]. The first major work involving Hindi occurred during TIDES Surprise Language exercise [3]. The objective of this work was to retrieve Hindi documents in responds to English queries. Similar work has been reported for Bengali [2] and Tamil. But nothing has been reported for Marathi, even though it is a prominent regional language. Some of the language specific obstacles of CLIR are proprietary encoding of text, lack of availability of corpora and variability in Unicode encoding [11].
There are three approaches to CLIR systems.

A. Machine Translation Approach

In CLIR, Machine Translation (MT) can be implemented in two different ways. The first way is to use an MT system to translate foreign language documents in the corpora into the language of the user's query. This is done off-line beforehand. This approach is not viable for large document collections, or for collections in which the documents are in numerous languages. For example, in experiments performed on German-Spanish CLIR, was not able to find direct German/Spanish MT so he had to use German/English MT, then English/Spanish MT. Not all the terms in the original German documents could be translated by this "triangulation" process. In the second method of using MT in CLIR, the users query in the "source" language is translated into the "target" language (the language of the documents in the stored collection). The "target" language query is then used to retrieve "target" language documents using classical IR techniques with both methods; the MT stage is separate from the retrieval stage. An ambiguity problem exists in the MT component, since the translated query does not necessarily represents the sense of the original query. For instance, translating the English query *big bank* to another language could produce an inappropriate translation since it is not clear whether "bank" means the institution or the edge of a river. MT systems normally attempt to determine the correct word sense for translation by using context analysis. However, a typical search engine query lacks context as it consists of a small number of keywords. MT is more efficient in document translation with clear context.

B. Dictionary-based query translation approach

In dictionary based query translation the query keywords are translated to the target language using Machine Readable Dictionaries (MRD). MRDs are electronic versions of printed dictionaries, and may be general dictionaries or specific domain dictionaries or a combination of both. The major problem in the bilingual dictionary approach is translation ambiguity in addition to problems of word inflection,

problems of translating word compounds, phrases, proper names, spelling variants and special terms.

Word inflection

A common problem with query translation is word inflection. This can be solved by lemmatization, where every word is reduced to its uninflected form or lemma. Another technique is called stemming, where different grammatical forms of a word are reduced to a common shorter form (not necessarily the lemma) called a stem, by the successive removal of word endings. The stemming rules remove "-ion", "-at" and "-ity" . For. Ex. **Gravitation** and **Gravity** transform **to** "Grav-" .

Phrases

For the success of CLIR, translation of phrases in their entirety, rather than individual word-for-word translation, is crucial. (Hull and Grefenstette 1996). Phrases matched against a manually built multi-word (phrase) dictionary showed higher precision than those translated by single word-based dictionaries.

Compound words

A compound word is a word formed from two or more words; compound words are not widely available in English, but very much used in other languages such as German, Finnish, etc. A compound word can be decomposed to two or more words, where each has a meaning are called compositional compounds, for example a Finnish word *kaupunginhallitus* (*city government*) is decomposed into two components, each of which has a meaning, *kaupungin* (*city*) and *hallitus* (*government*); but the problem occurs with nondecomposable compounds whose meaning can't be deduced on the basis of its components, or semi-compositional compounds with meanings that in part could have meaning but not related to the full compound meaning, for example the Finnish compound *krokotiilinkyyneleet* (*crocodile tears*). Compound splitting can be performed effectively by means of a lexicon-based morphological analyzer.

Proper names and spelling variants

In many documents technical terms and proper names are important text elements. Their translation is crucial for a good CLIR system. MT lexicons and general bilingual dictionaries lack translations for proper names and spelling variants. A common method used to handle untranslatable keywords is to include them untranslated in the target language query. If this word does not exist in the target language, the query will be less likely to retrieve relevant documents. Alternative methods exist to solve this problem for languages of the same writing system such as Transformation rule based translation (TRT). In TRT a word in one language is matched to a word in other language based on regular correspondences between the characters of the two languages. Thus the source language vocabulary and the target language vocabulary are regarded as spelling variants of each other.

Special terms

Special terms are most likely to be technical or scientific terms that are not widely available in general dictionaries. Special terms can be matched against a special dictionary, e.g. a medical term can be matched against a medical dictionary. Combining both general and specific domain dictionaries enhances the retrieval results. Two techniques are used to combine both dictionaries. Sequential translation translates the query keywords against the specific domain dictionary. If it fails to match, it uses the general dictionary, and a parallel translation that matches query keywords against both general and specific dictionaries. Both these techniques reduce the special terms translation problem but don't solve it altogether. For instance, translating a newspaper article that contains scientific terms, technical terms, political terms etc. needs more than a domain specific dictionary.

C. Corpus-based Approach

A Corpus is a repository of a collection of natural language material, such as text, paragraphs, and sentences from one or many languages. Two types of corpora (plural of "corpus") have been used in query translation:

Parallel Corpora

Parallel corpora consist of the same text in more than one language. An aligned parallel corpus is annotated to show exactly which sentence of the source language corresponds with exactly which sentence of the target text. When retrieving text from a parallel corpus, the query in this does not need to be translated, since a source language query can be matched against the source language component of the corpus, and then the target language component aligned to it can be easily retrieved. Parallel corpora can be populated using human translation, websites in more than one language or using MT methods. "Spider" systems have been developed to collect documents that have translation equivalents over the internet to produce corpora. The alignment process can be done by comparing documents by the presence of indicators. The indicator could be an author name, document date, source, special names in the document, numbers or acronyms, in fact anything which clearly corresponds in both the source and target language texts. Another example of parallel corpora alignment is the PTMiner tool (Nie, Simard, Isabelle and Durand 1999 [14]). The system first determines candidate sites, and then identifies a set of web pages on each web site that are indexed by a search engine. The next step is to construct pairs of web pages on the bases of pattern matching between URLs (index.html vs. indexf.html). The final step is to filter the candidate parallel page. Yet another alignment method was developed for bilingual reports of election results (Braschler and Sch¨auble 1998 [12]).

Comparable Corpora

Comparable corpora contain text in more than one language. The texts in each language are not translations of each other, but cover the same topic area, and hence contain an equivalent vocabulary. A number of statistical techniques can be used to derive topic-specific (often technical) bilingual dictionaries from parallel corpora.

III. SYSTEM ARCHITECTURE

The architecture of our CLIR system is shown in Figure 1.Our document Translation based approach initially stem the query words before looking up their entries in the bi lingual dictionary. In case of a match, all possible translations from

the dictionary are returned. In case a match is not found, the word is assumed to be a proper noun and therefore transliterated by the English-Devanagari transliteration module. The above module, based on a simple lookup table and corpus, returns the best three Marathi transliterations for a given query word. Finally, the translation disambiguation module disambiguates the multiple translations/transliterations returned for each word and returns the most probable Marathi translation of the entire query to the monolingual IR engine.

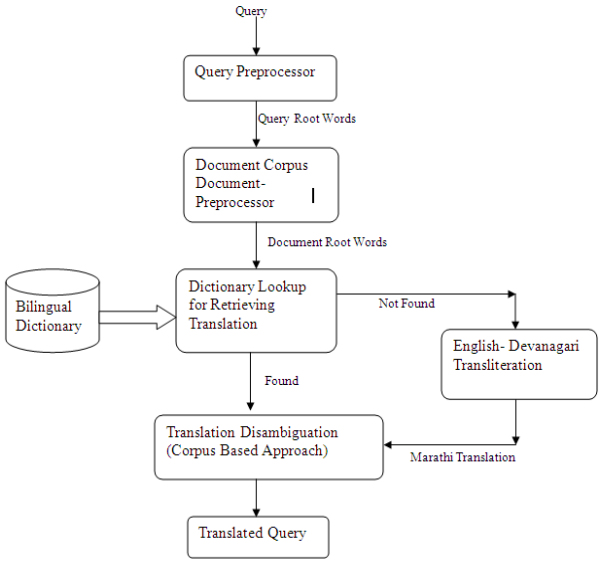

Figure. 1. System Architecture of English-Marathi CLIR System

The Algorithm clearly depicts the entire flow of proposed system.

Algorithm for Query Translation Approach

1: Remove all the stop words from query
2: Stem the query words to find the root words
3: **for** $stem_i$ 2 stems of query words **do**
4: Retrieve all the possible translations from bilingual dictionary
5: **if** list is empty **then**
6: Transliterate the word using to produce candidate transliterations
7: **end if**
8: en**d for**
9: Disambiguate the various translation/transliteration candidates for each word
10: Submit the final translated English query to Marathi

The individual module are explained in the subsequent sections

Query Preprocessor

It accept query in English. This query is passed through processes like tokenizing, stop word removal and stemming. The output is a bag of weighted query words. Proper nouns and nouns weigh the highest. The StringTokenizer class of java.util is used for tokenizing. High frequency words or function words such as articles, prepositions, and conjunctions are excluded from the tokenized text. The stemming process reduces tokens to their corresponding root form. The stemming algorithm used in our test is Robert Krovetz's KSTEM. It uses a list of words and a set of rules for handling inflectional and derivational morphology.

Document Preprocessor

The documents from the document corpora are subjected to processes like tokenizing, stop word removal and stemming. This helps in reducing the number of words to be stored. These words are indexed and stored in a hash table called the inverted index file.

English to Devanagari Transliteration

Many words of English used as part of the English or Marathi query, are not likely to be present in the English-Marathi bi-lingual dictionaries. So convert that words which is in English to Marathi using transliteration method. Simple rule based approach is used which utilizes the corpus to identify the closest possible transliterations for a given Marathi word. Create a lookup table which gives the Devanagari letter transliteration for each English letter. An English word is scanned from left to right replacing each letter with its corresponding entry from the lookup table [10].

Translation Disambiguation

The aim of the Translation Disambiguation module is to choose the most probable translation of the input query Q. With the bilingual dictionary, use corresponding words for one entry as the translations for that word. Generally, using the bilingual dictionary is similar to using the trained translation model. That is, given a term in the source language, the dictionary produces its translational equivalences in the target language yet without any probability. The word sense disambiguation, the sense of a word is inferred based on the company it keeps i.e based on the words with which it co-occurs. Similarly, the words in query, although less in number, provide important clues for choosing the right translation/transliterations. Assuming we have a query with three terms, s1, s2, s3, each with different possible translations/ transliterations, the most probable translation of query is the combination which has the maximum number of occurrences in the corpus. However, this approach is not only computationally expensive but may also run into data sparsity problem. We use a iterative disambiguation algorithm which examines pairs of terms to gather partial evidence for the likelihood of a translation in a given context.

Iterative Disambiguation Algorithm

Consider three words si, sj , sk, as shown in Figure 2, with multiple translations. Let their translations be denoted as {{ti,1}, {tj,1, tj,2, tj,3}, {tk,1, tk,2}}.

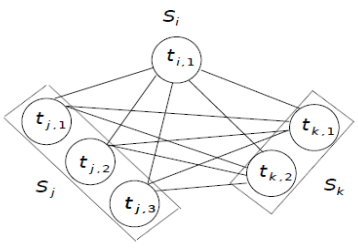

Figure 2: Co-occurrence Network for Disambiguating Translation/Transliteration

Given this, a co-occurrence network is constructed as follows: the translation candidates of different query terms are linked together. But, no links exist between different translation candidates of a query term. In the above graph, a weight w(t|si), is associated to each node t which denotes the probability of the candidate being the right translation choice for the input query Q. A weight, l(t, t0), is also associated to each edge(t, t0) which denotes the association measure between the words t and t0. Initially, all the translation candidates are assumed to be equally likely.
Initialization step:

$$w0(t|si) = 1/ |tr(si)| \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Table 1: Mathematical symbols involved in translation disambiguation

Symbol	Explanation
s_i	Source word
$tr(s_i)$	Set of translations for word s_i
t	Translation candidate, $t \in tr(s_i)$
$w(t|s_i)$	Weight of node t, where s_i is the source word
$l(t, t')$	Weight of link between nodes t and t'
$t_{i,m}$	m^{th} translation of i^{th} source word

After initialization, each node weight is iteratively updated using the weights of nodes linked to it and the weight of link connecting them.
Iteration step:

$$w^n(t|si) = w^{n-1}(t|si) + \sum_{t' \in inlink(t)} l(t, t') * w^{n-1}(t'|s) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where s is the corresponding source word for translation candidate t0 and inlink(t) is the set of translation candidates that are linked to t. After each node weight is updated, the weights are normalized to ensure they all sum to one.
Normalization step:

$$w^n(t|si) = w^n(t|si) / \sum_{m=1}^{|tr(si)|} w^n(t_{i,m}|si \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Steps 2 and 3 are repeated iteratively till convergence. Finally, the two most probable translations for each source word are chosen as candidate translations.
Netbeans 6 with JDK 1.6 isused for developing the user interface.

IV. EXPERIMENTS AND EXPECTED RESULTS

The user can run the system by entering the topics which user wants to search. Our systems provide the list of related documents which will be followed by the ranked document containing the information. As we have created our own document collection which is a domain specific & generalized English to Marathi machine translation dictionary. The bi-lingual dictionaries available with us also have Parts-Of-Speech (POS) information for each word. POS tagging the input query may help in reducing the ambiguity since translations of only matching POS will be retrieved. We will use following standard measure for evaluation: Mean Average Precision (MAP), R-Precision. Expected result for our system is translated document which is free from ambiguity using iterative disambiguation algorithm.

V. CONCLUSION

We present our English→Marathi CLIR systems using domain knowledge based. Our approach is based on document translation using bi-lingual dictionaries. Transliteration of words which are not in the dictionaries is done using simple rule based approach. Disambiguation the various translations/transliterations is performed using an iterative disambiguation algorithm which is based on term- term co-occurrence statistics.

REFERENCES

1. Anna R. Diekema., "Translation Events in Cross-Language Information Retrieval," In ACM SIGIR Forum, vol.3,No. 1, June 2004.
2. Debasis Mandal, Sandipan Dandapat, Mayank Gupta, Pratyush Banerje
3. and Sudeshna Sakar, "Bengali and Hindi to English Cross-language Text Retrieval under Limited Resources," CLEF 2007. Available at http://www.clef-campaign.org/2007/working_notes/ mandalCLEF2007.pdf
4. Leah S. Larkey, Margaret E. Connell and Nasreen Abduljaleel, "Hindi CLIR in thirty days," ACM Transactions on Asian Language Information Processing (TALIP), vol. 2, Issue 2. June 2003, pp. 130- 142.
5. Ballesteros, L. and Croft, B., "Dictionary Methods for Cross-Lingual
6. information Retrieval," In the Proceeding of the 7th International DEXA Conference on Database and Expert Systems Applications, pp. 791-801.1996.
7. Aljlayl, Ophir Frieder and David Grossman, "On Arabic-English Cross-Language Information Retrieval: A Machine Translation Approach," Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'02) 8-10 April 2002 pp. 2- 7.
8. Feng Yu, Dequan Zheng, Tiejun Zhao, Sheng Li and Hao Yu, "Chinese English Cross-Lingual Information Retrieval based on Domain Ontology Knowledge," Int. Conf. on Computational Intelligence and Security, 2006 vol. 2, 3-6 Nov. 2006, pp. 1460 – 1463
9. Robert Krovets, "Viewing morphology as an inference process," In Proceedings of the 16th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval., pp. 191-202, 1993.
10. G. Salton, A. Wong and C.S. Yang "A Vector space Model for Automatic indexing", Communication of the ACM, vol. 18 , Issue 11, November 1975, pp. 613 – 620.
11. Jorg Becker "Topic based VSM", Business information systems, proceedings of BIS 2003, Colorado Springs, USA.
12. Dik L. Lee, Huei Chuang and Kent Seamons, "Document Ranking and the Vector-Space Model", IEEE Software3, vol. 14, Issue 2, March 1997, pp. 67 – 75.
13. Seetha, Anurag Das, Sujoy and Kumar, M "Evaluation of the English Hindi Cross Language Information Retrieval System Based on Dictionary Based Query Translation Method", 10th Int. Conf. on Information Technology, (ICIT 2007). 17-20 Dec. 2007, pp. 56-61.
14. D. Neumann.,"A cross-language question answering system for German and English," CLEF –2003.