Cross lingual Information Retrieval

Chapter 1. CLIR and its challenges

A large amount of information in the form of text, audio, video and other documents is available on the web. Users should be able to find relevant information in these documents. Information Retrieval (IR) refers to the task of searching relevant documents and information from the contents of a data set such as the World Wide Web (WWW). A web search engine is an IR system that is designed to search for information on the World Wide Web. There are various components involved in information retrieval. IR system has following components:

- Crawling: Documents from web are fetched and stored.
- Indexing: An index of the fetched documents is created.
- Query: Input from the user.
- Ranking: The systems produces a list of documents, ranked according to their relevance to the query.

Information on the web is growing in various forms and languages. Though English dominated the web initially, now less than half the documents on the web are in English. The popularity of internet and availability of networked information sources have led to a strong demand for Cross Lingual Information Retrieval (CLIR) systems. *Cross-Lingual Information Retrieval (CLIR)* refers to the retrieval of documents that are in a language different from the one in which the query is expressed. This allows users to search document collections in multiple languages and retrieve relevant information in a form that is useful to them, even when they have little or no linguistic competence in the target languages. Cross lingual information retrieval is important for countries like India where very large fraction of people are not conversant with English and thus don't have access to the vast store of information on the web.

1.1 Approaches to CLIR

Various approaches (Amelina & Taufik, 2010) can be adopted to create a cross lingual search system. They are as follows:

1.1.1 Query translation approach

In this approach, the query is translated into the language of the document. Many translation schemes could be possible like dictionary based translation or more sophisticated machine translations. The dictionary based approach uses a lexical resource like bi-lingual dictionary to translate words from source language to target document language. This translation can be done at word level or phrase level. The main assumption in this approach is that user can read and understand documents in target language. In case, the user is not conversant with the target language, he/she can use some external tools to translate the document in foreign language to his/her native language. Such tools need not be available for all language pairs.

1.1.2 Document translation approach

This approach translates the documents in foreign languages to the query language. Although this approach alleviates the problem stated above, this approach has scalability issues. There are too many documents to be translated and each document is quite large as compared to a query. This makes the approach practically unsuitable.

1.1.3 Interlingua based approach

In this case, the documents and the query are both translated into some common Interlingua (like UNL). This approach generally requires huge resources as the translation needs to be done online.

A possible solution to overcome the problems in query and document translations is to use query translation followed by snippet translation instead of document translation. A snippet generally contains parts of a document containing query terms. This can give a clue to the end

user about usability of document. If the user finds it useful, then document translation can be used to translate the document in language of the user.

With every approach comes a challenge with an associated cost. Let us take a look at the general challenges in CLIR.

1.2 Challenges in CLIR

We face the following challenges in creating a CLIR system:

1. Translation ambiguity:
    - While translating from source language to target language, more than one translation may be possible. Selecting appropriate translation is a challenge.
    - For example, the word मान (maan, respect/neck) has two meanings neck and respect.
2. Phrase identification and translation
    - Indentifying phrases in limited context and translating them as a whole entity rather than individual word translation is difficult.
3. Translate/transliterate a term:
    - There are ambiguous names which need to be transliterated instead of translation.
    - For example, भास्कर (Bhaskar, Sun*)* in Marathi refers to a person's name as well as sun. Detecting these cases based on available context is a challenge.
4. Transliteration errors:
    - Errors while transliteration might end up fetching the wrong word in target language.
5. Dictionary coverage

- For translations using bi-lingual dictionary, the exhaustiveness of the dictionary is important criteria for performance on system.

6. Font:
    - Many documents on web are not in Unicode format. These documents need to be converted in Unicode format for further processing and storage.
7. Morphological analysis (different for different languages)
8. Out-of-Vocabulary (OOV) problems
    - New words get added to language which may not be recognized by the system.

1.2.1 Factors affecting the performance of CLIR systems

Among the different challenges, the major factors which influence the performance of CLIR systems are given in detail below:

1.2.1.1 Limited size of Dictionary

The limited size of dictionary contributes to translation errors. New words get added to the language quite frequently and maintaining the dictionary up to date with these new words is difficult. Also compounds and phrases can be formed from existing words in the language. No dictionary can contain all possible compounds and phrases. A specific domain can generate a specific terminology which might not be present in general dictionary. Inflected word forms are not included in dictionary. Thus normalization process like stemming becomes essential.

1.2.1.2 Query translation/transliteration performance

The phenomenon of translation ambiguity is common in cross lingual information retrieval and refers to increase of irrelevant search key senses due to lexical ambiguity in source and target languages. A search key may have more than one sense in source language which may be expressed by dictionary by providing several alternatives. During the translation process, extraneous senses may be added to the query due to the fact that the translation alternatives

may also have more than one sense. Thus lexical ambiguity appears in both source and target language.

Chapter 2. Related work

Over the last decade a lot of research has been done on information retrieval. Several approaches have been proposed for many sub problems which exists in the task of information retrieval. Some of them focussed on improving performance in terms of computation time, quality while some of the discussed innovative strategies and architectures for information retrieval system.

In this section we will look at research work related to both offline and online processing of our cross lingual search system. Section 2.1 describes various strategies used for maintaining freshness of crawl. For optimal crawling process, we need to utilize the resources effectively. We have to adopt different strategies for resource constrained crawling which limits the maximum URLs that can be crawled in each depth. Section 2.2 describes various methods used for selecting URLs to be fetched in each depth. In online processing of *Sandhan*, accuracy of translated query plays an important role in deciding the quality of retrieved results. In section 2.3 we describe the methods of improving the translation quality.

2.1 Incremental Crawling

With millions of pages getting updated every day, it is important to keep the crawl up-to-date with latest pages from the web. Gone are the days where you manually keep the system for crawling for fixed number of depths and merge the crawl separately every time. The number of crawled pages increases exponentially in each depth. This demands an architecture which incrementally adds new/updated pages to the existing crawl.

Cho, J. and Garcia-Molina, H. (1999) gives insight on how to develop an incremental web crawler. This crawler periodically updates the crawl in a batch mode. The report mentions about two categories of crawlers *viz.,* periodic crawler and incremental crawler. Periodic crawler which is also called as snapshot crawler crawls a certain number of pages till a sufficient depth and stops crawling. It recrawls the same set of pages after certain amount of time and replaces the old crawl with new one. On the other hand, incremental crawler keeps on crawling

pages refreshing the existing crawl and replacing old pages with new ones. With the help of incremental crawling, one can estimate the periodicity with which a page changes and hence optimize the crawling process. Another important difference between the two strategies is that periodic crawler can index a new page only when the next crawling cycle starts while incremental crawler updates the page as soon as it is found. To design an incremental crawler, study of change rate of web pages is essential. If all the pages change at equal intervals, then periodic crawler may be as effective as incremental crawlers.

While exploring different crawling strategies it is important to define the metrics to be used for measuring the freshness of the crawl. Cho & Garcia-Molina (2000) has defined freshness of the crawl as follows:

Let S = (e_1, e_2 ... e_N) be the local database with N elements. The freshness of a local element e_i at time t is

$$F(e_i, t) = \begin{cases} 1 & if\ e_i\ is\ up-to-date\ at\ time\ t \\ 0 & otherwise \end{cases}$$

Then, the freshness of the local database S at time t is

$$F(S; t) = \frac{1}{N} \sum_{i=1}^{N} F(e_i, t)$$

Another metric used for evaluation of crawling strategy is age. To capture \how old" the database is, we define the metric age as follows:

Age of the local element e_i at time t is

$$A(e_i, t) = \begin{cases} 1 & if\ e_i\ is\ up-to-date\ at\ time\ t \\ 0 & (t\ -\ modification\ time\ of\ e_i) otherwise \end{cases}$$

The age of S tells us the average age of the local database. Using both these metrics we can evaluate different crawling strategies. Design of an incremental crawler is dependent on the crawling strategy used.

Sigurosson (2005) describes one such web crawler called "*Heritrix*". It is world's first open source web crawler. It checks whether the crawled page has changed or not and accordingly updates its wait interval (fetch interval). The fetch interval is divided by a constant if the page has changed while it is multiplied by a constant if it has not changed. Such an adaptive strategy for scheduling helps in learning the average rate of change of pages.

2.2 Resource constrained crawling

While incremental crawling and scheduling helps in maintaining crawl up-to-date, it requires a lot of resources which may not be available in a small scale organization. In small organizations or academic institutions, resources available are limited which puts limit on the throughput of the crawler. In such a scenario, crawling important URLs first is required. Scheduling policies help in optimizing the bandwidth required for crawling by increasing the crawl period of pages which are not updated frequently. However, even within URLs, which have more frequency of updates, we need to have a priority on the URLs to be fetched first.

A lot of work exists on classifying URLs based on page content. Page content based classification can be helpful in deciding the next fetch interval of the URL. However, classification of URLs before fetching is required for prioritizing URLs in current depth.

There are many research contributions for prioritizing URLs during fetching. Some of them use purely URL based features, while some of them use parent information, contextual information, *etc.* Let us look at few of them.

Min-Yen Kan (2004)quantified the performance of web page classification using only the URL features(URL text) against anchor text, title text and page text, showed that URL features when treated correctly, exceeds the performance of some source-document based features.

Min-Yen Kan *et al.* (2005) added URL features, component length, content, orthography, token sequence and precedence to model URL. The resulting features, used in supervised maximum entropy modelling, significantly improve over existing URL features.

Fish search(Bra, et al. 1994) ,one of the first dynamic Web search algorithms, takes as input a seed URL and a search query, and dynamically builds a priority list (initialized to the seed URL) of the next URLs (hereafter called nodes) to be explored. As each document's text becomes available, it is analyzed by a scoring component evaluating whether it is relevant or irrelevant to the search query (1-0 value) and, based on that score. A heuristic decides whether to pursue the exploration in that direction or not.

Shark search algorithm(Hersovic, et al. 1998), a more aggressive algorithm, instead of binary evaluation of document relevance, returns a score between 0 and 1 in order to evaluate the relevance of documents to a given query, which has direct impact on priority list. Shark search calculates potential score of the children not only by propagating ancestral relevance scores deeper down the hierarchy, but also by making use of the meta-information contained in the links to documents.

Jamali et al. (2006) used the link structure analysis with the similarity of the page context to determine the download pages priority, while Xu & Zuo (2007)use the hyperlinks to discover the relationships between the web pages.

2.3 Query Translation and Transliteration

In CLIR, either the query or the document or both need to be converted into a common representation to retrieve relevant documents. Translating all documents into the query language is less desirable due to the enormous resource requirements. Normally the query is translated into the language document collection. Three methods (Jagarlamudi and Kumaran 2008) are generally used for translating the query *viz.* machine readable bilingual dictionaries, parallel texts and machine translation systems. Most of the queries in IR are short in nature and lacks necessary syntactical features which are required for machine translation. Most of the approaches use bilingual dictionaries for translation of queries. Such bilingual dictionaries may not be exhaustive. If the translation for a word is not available, then transliteration of the word is generally used.

Much of the work for transliteration in Indian languages has been done from one Indian script to another. Om transliteration scheme (Madhavi, et al. 2005) provides a script representation which is common for all Indian languages. The display and input are in human readable Roman script. Transliteration systems from Indian languages to English have to overcome the problem of spell variations. Indian languages have lot of spell variations. This variation is much more when words in Indian languages are written using Latin script.

(Surana, Singh and Kumar 2008) highlighted the importance of origin of the word and proposed different ways of transliteration based on its origin. A word is classified as Indian or foreign using character based n-grams. They have reported their results on English-Hindi and English-Telugu datasets. (Malik 2006) tried to solve a special case of Punjabi machine transliteration. They converted Shahmukhi to Gurumukhi using rule based transliteration. (Gupta, Goyal and Diwakar 2010) used an intermediate notation called as WX notation to transliterate the word from one language to another. They divided the problem of transliteration into two sub problems – transliterating source word to WX notation and transliterating WX notation to target notation. (Janarthanam, S and Nallasamy 2008) also proposed similar idea of using intermediate form for transliteration. They converted source word to an intermediate form called compressed word form and then used modified Levenshtein distance to match the right candidate from target language index.

(Rama and Gali 2009) proposed the use of SMT system for machine transliteration. In this approach, words are split into constituent characters and each character acts like a word in a sentence. Our approach uses a different word segmentation technique, which, when combined with SMT gives better results.

References

Amelina, Nasharuddin Nurul, and Abdullah Muhamad Taufik. "Cross-lingual Information Retrieval." *Electronic Journal of Computer Science and Information Technology (eJCSIT)* 2 (2010).

Bra, Paul De, Geert-jan Houben, Yoram Kornatzky, and Reinier Post. "Information Retrieval in Distributed Hypertexts." *Proceedings of RIAO, Intelligent Multimedia, Information Retrieval Systems and Management.* New York, 1994.

Cho, J., and H. Garcia-Molina. *The Evolution of the Web and Implications for an Incremental Crawler.* Stanford, 1999.

Cho, Junghoo, and Hector Garcia-Molina. "Synchronizing a database to Improve Freshness." *MOD.* Dallas, TX USA: ACM, 2000.

Gupta, Rohit, Pulkit Goyal, and Sapan Diwakar. "Transliteration among Indian Languages using WX notation." *Semantic Approaches in Natural Language Processing.* 2010.

Hersovic, Michael, Michal Jacovi, Yoelle S. Maarek, Dan Pelleg, Menanchem Shtalhaim, and Sigalit Ur. "The shark-search algorithm. An application: tailored Web site mapping." *Proceedings of the seventh international conference on World Wide Web.* Amsterdam: Elsevier Science Publishers, 1998. 317-326.

Jagarlamudi, Jagadeesh, and A. Kumaran. "Cross-Lingual Information Retrieval System for Indian Languages." *Advances in Multilingual and Multimodal Information Retrieval.* Berlin Heidelberg: Springer-Verlag, 2008. 80-87.

Jamali, Mohsen, Hassan Sayyadi, Babak Bagheri Hariri, and Hassan Abolhassani. "A Method for Focused Crawling Using Combination of Link Structure and Content Similarity." *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence.* Washington, DC, USA: IEEE Computer Society, 2006. 753-756.

Janarthanam, Srinivasan C, Sethuramalingam S, and Udhyakumar Nallasamy. "Named entity transliteration for cross-language information retrieval using compressed word format mapping algorithm." *Proceedings of the 2nd ACM workshop on Improving non english web searching.* ACM, 2008. 33-38.

Kan, Min-Yen. "Web page classification without the web page." *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters.* New York: ACM, 2004.

Kan, Min-Yen, and Hoang Oanh Nguyen Thi. "Fast webpage classification using URL features." *Proceedings of the 14th ACM international conference on Information and knowledge management.* ACM, 2005.

Madhavi, Ganapathiraju and Mini, Balakrishnan and Balakrishnan, N and Raj, and Reddy. "Om: One tool for many (Indian) languages." *Journal of Zhejiang University-Science A* (Zhejiang University Press) 6, no. 11 (2005): 1348--1353.

Malik, Muhammad G. "Punjabi machine transliteration." *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2006. 1137-1144.

Rama, Taraka, and Karthik Gali. "Modeling machine transliteration as a phrase based statistical machine translation problem." *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration.* Association for Computational Linguistics, 2009. 124-127.

Sigurosson, Kristinn. "Incremental crawling with Heritrix." *In Proceedings of the 5th International Web Archiving Workshop (IWAW '05).* 2005.

Surana, Harshit and Singh, and Anil Kumar. "A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages." *IJCNLP.* 2008. 64-71.

Xu, Qingyang, and Wanli Zuo. "First-order focused crawling." *Proceedings of the 16th international conference on World Wide Web.* New York, NY, USA: ACM, 2007. 1159-1160.