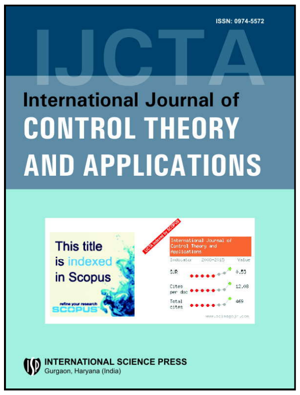

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 19 • 2017

Query Optimization using Query Sequence for Hindi-English Code-Mixed Query

Amit Jena[1] and Rakesh Chandra Balabantaray[1]

[1] *Department of Computer Science and Engineering, IIIT Bhubaneswar, Bhubaneswar, Odisha, India,*
Emails: amitcuj@gmail.com, rakeshbray@gmail.com

Abstract: In this study, we are addressing two aspects of web search query. First, we tried to address the problem of Hindi-English code mixed search engine query. Code-mixing is frequently observed in user generated content on social media and also in web search engines, especially from multilingual users. We have considered Hindi-English mixed term queries written in common roman script. Second, we also studied the possibility of exploring the query sequence of users to automate the "Query Reformulation" process. Web search users frequently modify their queries till they get relevant results. Previous research work has mainly concentrated on suggesting queries for users. Some research has also been carried out to identify the concept behind the queries and propose a query reformulation. The query once fired into our system, will undergo language identification to find out the language of each query term, normalize the query terms to remove ambiguities and then the normalised terms are converted to correctly spelled English words. Then the query is analysed for inherent sequential continuity and final reformulation is done.

Keywords: Query Reformulation, Query Refinement, Query Sequence, Language Identification

1. INTRODUCTION

Google now processes over 40,000 search queries every second on average, and translates to over 3.5 billion searches per day and 1.2 trillion searches per year worldwide. Out of that approximately 30% are reformulation to the previously entered query. For example, a user may search for "capital of India" and then gives the query "population of the country". Then the existing search engines treat each query as independent queries. The search result in Google for second query gives us a general population statistics of different countries of the world. But the user intention was to search for the population of India.

The search engine side of query reformulation is in place for quite some time now. Using our system we can do searching like a man-to-man conversation by asking subsequent questions in a way we could never do with regular search engines. We try to imitate the way humans converse, where there is a link between the present and previous query. Ongoing query refinement techniques are helpful, but text based conversational search is a step ahead. It's not out only padding terms to the original query but about parsing the language. We parse the query so as the search engine understands, really comprehends, what we intend to search.

Our system has two modules:

1) Handling of Hindi-English mixed query written in roman script
2) Query reformulation using query sequence

1.1. Query Reformulation

Web search process involves the user entering a query, learning from the results, and reformulating the queries till the information need is satisfied. When a query is entered to the search system, it makes an attempt to return the closest possible results to that query. Many a times it has been observed that the user reformulates the query with a hope to retrieve better results. Generally users tend to ignore the low ranked results to the query. They scroll through the high ranked results and chose to modify the query than going down to the successive results to search for the information (Jansen and Spink, 2006). Research into query reformulation and user behaviour have shown that 37% of search queries are reformulations to previous queries (Jansen e al., 2007) and that 52% of users reformulate their queries (Jansen et al., 2005) to get a better result.

Generally there is an inherent continuity in the user queries. The research work till now has been to identify and understand the query reformulation behaviour. The intention behind this was to refine the query auto suggestions. Also, if we are able to predict the query reformulation with high accuracy then we will be able to evaluate the user satisfaction with query results. This will cross validate our traditional approach to evaluate user satisfaction based on click through information.

1.2. Handling of Mixed Language query terms

Bilingual speakers (users having knowledge of both English and Hindi) widely use code-mixing and code-switching in their regular use of language on social media and search platforms. This tendency is highly observed in search query logs too. Users tend to enter the query in casual manner. Code-Mixing can be formally defined as embedding of phrases, words or morphemes of one language into a query of another language. And code-switching refers to the co-occurrence of words belonging to two different grammatical systems (Gumperz., 1982). We are referring to both code-mixing and code-switching by Code-Mixing.

Hindi-English speakers generate high volume of code-mixed query. Vyas et al. (2014) found that it is highly complex to analyse such query stems because they do not generally follow any formal grammar, have spelling variations and typos, lack of annotated data, and query text in inherently conversational in nature.

2. RELATED WORK

2.1. On Query reformulation

Traditionally Query reformulation is done using three global methods:

1. assisting the user to reformulate the query using vocabulary tools
2. using a manual thesaurus
3. by automatic thesaurus generation

2.1.1. Using vocabulary techniques for Query reformulation

While a user is searching, we can take help of various vocabulary tools to guide the user. There can be many support techniques to show the user which search query is giving accurate result and which need some refinement. The user can be provided with the following list about their search:

1. omitted words form the query (stop words)
2. stemmed words
3. number of matches for each term and phrase
4. dynamic conversion of words to phrases

The search system may also suggest alternate queries by taking help from a thesaurus or some existing vocabulary. We can also provide the user with the inverted index list of words, which would help the user in finding appropriate word substitutions from the indexed terms.

2.1.2. Query expansion using a manual thesaurus

The manual query expansion technique works in close association with the user. The user provides reinforced information for the search query terms or phrases. The user may suggest some additional terms to update the query. Some wen based search engines [17] like Google and Bing uses auto suggestion in response to the entered query. The user gets a list of closely related queries for the entered query.

The most widely used query expansion technique uses global analysis of query using thesaurus. For all the query terms entered for the query, the query should be reformulated by substituting the terms with their synonyms to match all similar results. This can be achieved by using a thesaurus. This idea when combined with the term weighting factor, can boost the search result significantly.

2.1.3. Dynamic generation of thesaurus

The manual formulation and maintenance of thesaurus is highly cost intensive. This can be substituted with dynamic generation of thesaurus by analysing the existing collection of documents to make it cost effective. There are two main techniques to create a thesaurus dynamically:

- Finding out the co-occurrence of words: Words co-occurring in a document or paragraph are more likely to be semantically similar or synonymous. We just then use simple text analytics approaches to find the similar words.
- The second approach is to use a grammar oriented approach. Here we try to analyse the grammatical relations and how words are grammatically dependant on each other. For example, entities that are woven, stitched, ironed, hooded, collar, are more likely to be dress materials. Simply using word co-occurrence may be misled by the parser errors, but if we take the help of grammatical relations then it is more robust.

There are some other techniques used by researchers for Query reformulation:

2.1.4. Query Reformulation using Anchor text

Query reformulation techniques using the query logs is widely accepted technique to understand user intention behind the query and improving the result retrieval effectiveness. In the paper the author suggested that the readily available anchor text can be an effective alternative to the query log technique. They studied various existing query refinement and reformulation techniques based on stemming, query log based substitution, and query expansion using available TREC collections. The methods demonstrated by highlights that for standard collections, log based query reformulation techniques are more efficient. But research suggest query expansion to be much more effective form of query reformulation than mere word substitution. Their results showed that for the above discussed techniques using anchor text as substitute for query log does not give any significant improvement.

2.1.5. Query reformulation using concept based matching

Here the main idea was to do concept based term matching using a list of words. In their work, they showed that relying on surface level term similarity results into many false positives. There are cases where queries with similar topics but different intent are falsely identified as being reformulated. The researchers proposed a changed representation of web based search queries by identifying the semantics and user intention behind the query. This model showed significant improvement in query reformulation performance by using features of query semantics.

2.2. On Code-mixed query terms

Liu and Solorio (2008) used existing taggers for both English and Spanish language over code-switched domain for POS tagging of English- Spanish log. They achieved an accuracy of 93.48%. But the data used was manually transcribed hence failed to address the inherent problems of code-mixed query. Vyas et al. (2014) formally studied the problem, reported challenges in processing Hindi-English code-mixed query. He performed initial experiments on POS tagging. Their study showed that identification of code-mixed term's language and normalization are critical for the POS tagging task.

3. PROBLEM DEFINITION

Below we are defining the problem and the relevant terms that we will be using throughout.

Definition: Query Reformulation can be formally defined as the act of entering a refined query Qi to modify the previous search query Q_{i-1} with the hope of getting better results to meet the information need of the user.

Definition: We maintained a search window of 10 queries $\{Q_i, Q_{i+1}, \ldots Q_{i+10}\}$ i.e. we will keep track of 9 previous queries while analysing the current query. In this work, we automate the query reformulation. In this work we are handling the problem: First, convert the code-mixed search query to Standard English words. Then Given a sequence of queries $\{Q_i, Q_{i+1}, \ldots, Q_{i+9}\}$, and the current query Q , predict whether Q should be reformulated. If yes, reformulate it and send the new reformulated query to the search system in order to get more relevant results.

4. APPROACH

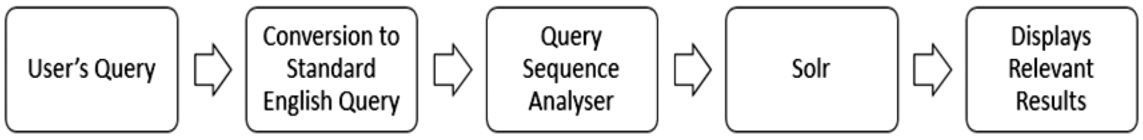
User's Query
Conversion to Standard English Query
Query Sequence Analyser
Solr
Displays Relevant Results

Figure 1: Architecture of our Proposed System

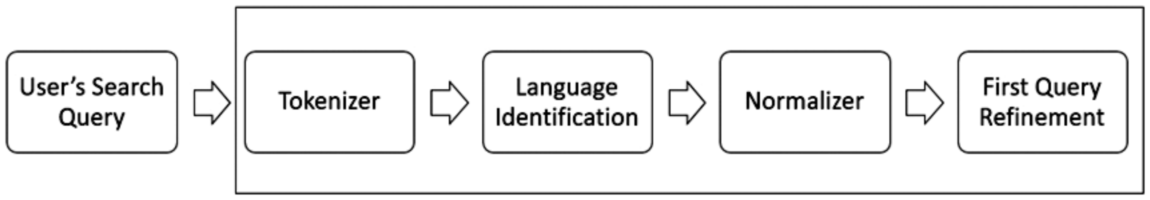

Figure 2: Pipeline stages of Hindi-English mixed query to Standard English Query

4.1. Conversion of code-mixed search query to standard words

This stage has the Language Identification and Normalization layers. Our system takes the user entered query as input on which each module runs sequentially. The query may be bilingual code-mixed Hindi-English query or monolingual query, but the script is Roman. Twokenizer (Owoputi et al., 2013) was used to tokenize the oriinal query into individual tokens. The Language Identification stage processes the tokens and tags then with a label from our predefined language labels. Based on the language label tag of the individual token, the Normalizer runs the corresponding Hindi normalizer or the English normalizer.

4.1.1. Language Identification

Each token of the original query is given a tag out of the 'en', 'hi' and 'rest' to identify its language. Words that a Hindi-English bilingual speaker could identify as either Hindi or English were marked as 'hi' or 'en'. The label 'rest' was given to everything else.

The feature set comprised of:

1) BNC: We took the normalized frequency of the 10, 00,000 English words in British National Corpus (BNC).
2) LEXNORM: It's a binary feature related to the presence of the word in the Han et al. (2011) dataset.
3) HINDI DICT: It's also a binary feature marking the presence of the word in a Hindi corpus of 30,823 transliterated words, released by Gupta (2012).
4) NGRAM: We have taken the n-grams of tokens up to n=3.

We used CRF to perform the Language identification work by formulating the problem as a sequence labelling task.

4.1.2. Normalization

Words in Roman script that were given the tag 'hi' in the language identification stage were labelled with their standard form in the native script of Devanagari. Similarly, English words with language tag 'en' were corrected with their standard spelling. Words with language tag 'rest' were left unaltered.

After the language identification task is over, the noisy non-standard tokens, like Hindi words inconsistently written in many ways using the Roman script, in the entered query were transformed into standard words in the identified language. To address this, a normalization model which does language-specific transformations, resulting into correctly spelled words for a given token is built.

The Hindi normalizer, converted the words identified as Hindi to Devanagari script from roman script. We used GIZA++ and CRF to obtain Hindi words in Devanagari script from roman script. GIZA++ was used to obtain character alignment between non-standard Hindi words written in roman to normalized words format. Then CRF was employed for conversion from Roman script to Devanagari Script using learnt letter transformation. Then we used a standard Hindi dictionary to convert normalised Hindi word from Devanagari script to equivalent English words in roman script.

The English normalizer too worked on the same principle. Using edit distance and comparing with BNC (British National Corpus), we converted the words to Standard English words. These words were free from spelling errors and have no abbreviations.

4.2. Query Reformulation

First, we need to identify if there is need of query reformulation i.e. we need to find out if the query is auto refined if this will result to better results. The reformulation strategy will take into consideration, the inherent continuity in the query sequence, the flow of concept and user intention behind the query. Multiple strategies exist to reformulate a query resulting in to different types of query reformulations:

1) Generalization: A generalization reformulation occurs when the second query is intended to seek more general information compared to the first query.

2) Specification: A specification reformulation occurs when the second query is intended to seek more specific information compared to the first query.
3) Same Intent: A same intent reformulation occurs when the second query is intended to express the same intent as the first query.

We used the features given below to predict if query reformulation is needed:

1) Length of the queries and difference between them.
2) Count of out-of-vocabulary words in Q_1, Q_2 and the difference between them.
3) Count of common "exact match" concepts.
4) Count of common "approximate match" concepts.
5) Count of common "semantic match" concepts.
6) Count of concepts in query window and the difference between them.
7) Count of concepts in current query but not in previous query.
8) Count of concepts in previous query but not in current query.
9) Then do the evaluation by replacing the concepts with keywords.

We are maintaining a sliding window size of 10 for searched queries $\{Q_i, Q_{i+1}, \ldots, Q_{i+10}\}$, i.e. we will keep track of 9 previous queries while analysing the current query. At present our system is able to handle the following types of queries:

- Pronoun replacement: If the subsequent query have any pronoun then it is substituted with the noun from the previous query.
- Name entity identification: If there is no pronoun in the subsequent query, but there is some named entity in the queries then it tries to combine the queries belonging to the same named entity.
- Continuity of concept: The query is segmented into semantic phrases depending upon the mutual information score. Whenever the point wise mutual information between two consecutive words drop below a predefined threshold, a segment break is inserted.

4.3. Pronoun replacement

4.3.1. All subsequent queries are linked

We used the Stanford CoreNLP tool in our proposed model. When the first query is fired, it is analysed and the POS, named entities, normalized dates, times, and numeric quantities, and subject are recorded. When second query is searched, using the above technique we search for pronouns, if found, it is substituted with the noun from previous query and reformulated. The newly generated query is stored in query log. When the third and subsequent queries are fired, the pronouns are replaced with the corresponding nouns identified from the previous queries.

4.3.2. Linked queries are not consecutive

Let the query Qi+1 and Qi+4 are linked and queries Qi+2 and Qi+3 are independent queries. Then it involves two sub tasks: identifying if queries are independent and then reformulating the dependant query based upon the noun information of previous query.

The independent queries are identified by the absence of any pronouns in the query. The occurrence of pronoun in any subsequent query is marked as dependant query and the noun to be substituted in Qi+4 should be of Qi+1 but not of Qi+3.

Types of pronouns resolved:

1) Personal Pronouns
   - First Person: The person in action. {I, me and we, us}
   - Second Person: The person at which the action is directed to. {you}
   - Third Person: The person, people, or things about which we are talking or writing {she, her he, him it and they, them}
2) Relative Pronouns
   - {that, which, who, whom, whose, whichever, whoever, whomever}
3) Demonstrative Pronouns
   - Refers to nearby things {this and these}
   - Refers to distant things {that and those}

4.4. Named entity identification

For the search query we identify the named entities present in the query. With each entered query, we search for presence of named entities like company name, hotel, monuments and establish a link between them and the place (city or town), if present in the previous query. For e.g. if Q_1 is “Hotels at New Delhi” and Q_2 is “Hotel Taj”, then the reformulated query becomes “Hotels Taj at New Delhi”.

4.5. Continuity of concept

To capture concept similarity, we compute the concept similarity by measuring the semantic similarity between two queries in consideration. We used the following formula to compute similarity between two sequence of words, $Q = \{q_1, q_2, \ldots, q_I\}$ and $S = \{s_1, s_2, \ldots, s_J\}$.

$$P(S|Q) = \prod_{i=1}^{I}\prod_{j=1}^{J} P(S_i|Q_j)P(Qj|Q)$$

where P(t|Q) is the unigram probability of word ‘t’ in the entire Query Q.

5. RESULTS

The Hindi-English mixed query, was fired into our solr search system. The solr system had data indexed in it. We indexed pdfs, text files and doc files. The precision value for the original query and the auto-reformulated query was calculated. We randomly chose 2 users to use our system and fire Hindi-English mix query. They were asked to search for 100 queries. We had to manually ask the users to fire the query, since there is no query log available for Hindi-English mixed query for search engine.

The accuracy of language identification using CRF is 88%. And the accuracy of the Hindi and English normalizer was 76.5% and 81% respectively. The normalization module yields an overall accuracy of 78%.

The reformulated queries were tested by putting them into our solr based local search platform. In this, we have indexed 100 pdf and doc files related to India. The files indexed cover a wide range of details about India starting from history, geography, politics and culture. The precision for each of the code mixed query that was reformulated is given in the Table - 1. For precision calculation we have taken top 5 documents fetched for each query. Each document (pdf/doc file) is given a relevancy score as follows:

- Document is relevant : 1
- Document is partial relevant : 0.5
- Document is not relevant : 0

The relevancy score is awarded on the basis of what the user intend to search but not on what search query he is entering alone. In the Table - 1 we give a subset of the search query.

Table 1
Reformulated Query Result

Query #	*Original Query*	*P@3*	*P@5*	*Final Reformulated Query*	*P@3*	*P@5*
Q_1	bharat ki rajdhani	0	0	bharat's capital	0.6	0.5
Q_2	capital of the country	0.5	0.35	capital of bharat	0.6	0.5
Q_3	last viceroy of india	1	1	last viceroy of india	1	1
Q_4	ind ka prim minister kaun hai	0	0.2	india's prime minister who is	1	0.95
Q_5	unka kitna age	0	0	india's prime minister how much age	1	1
Q_6	gov jobs jiska exam hota hai	0	0.2	india govt job whose exam have	0.65	0.7
Q_7	festivals here	0	0	festivals india	1	1
Q_8	mickel jackson fav step	0	0	mickel jackson favourite step	0	0
Q_9	mumbai to kolkata kese jana hai	0.3	0.28	mumbai to kolkata how to go	0.6	0.73
Q_{10}	what people here eat	0	0	what people india eat	0.6	0.5

6. CONCLUSION

Our effort was to address the following issues pertaining to search system:

1) The popular search engines, perform terribly when Hindi-English code mixed queries are searched.
2) They give a list of recommended queries for our query, but do not auto refine and reformulate them.
3) Each query is treated as individual entity and the inherent connection between them is ignored.
4) Thus, each time the user has to reformulate themselves and enter a new query to get relevant results.
5) Multi-lingual users and social media addicted users, can search using mixed query of Hindi and English.
6) Higher user satisfaction.

In the future, we intend to work to include Odia language into our study. We also intend to create a query log of code mixed multilingual search queries. We would like to improve upon the accuracy of language identification and normalization module. Our extend aim is to create a personalized search product that can search our computer system intelligently understanding or intention. This can also be integrated with websites for better intra-site search experience for multilingual users. We intend to further extend our work and build a shallow parser for Hindi-Odia-English, which will benefit the local users and help in NLP research.

REFERENCES

[1] Sharma, Arnav, et al. "Shallow Parsing Pipeline for Hindi-English Code-Mixed Social Media Text." *arXiv preprint arXiv:1604.03136* (2016).

[2] BNC Consortium. "The British National Corpus, version 3 (BNC XML Edition); 2007." *Distributed by Oxford University Computing Services on behalf of the BNC Consortium. URL: http:// www. natcorp. ox. ac. uk (last accessed 25th May 2012).*

[3] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL (System Demonstrations)* (pp. 55-60).

[4] Huang, Jeff, and Efthimis N. Efthimiadis. "Analyzing and evaluating query reformulation strategies in web search logs." *Proceedings of the 18th ACM conference on Information and knowledge management.* ACM, 2009.

[5] Huang, Jeff, and Efthimis N. Efthimiadis. "Analyzing and evaluating query reformulation strategies in web search logs." *Proceedings of the 18th ACM conference on Information and knowledge management.* ACM, 2009.

[6] Pass, G., A. Chowdhury, and C. Torgeson. "A picture of search. Infoscale'06, Hong Kong." *Proceedings of the 1st international conference on Scalable information systems. ACM, New York.* 2006.

[7] Mogotsi, I. C. "Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval." *Information Retrieval* 13.2 (2010): 192-195.

[8] Dang, Van, and Bruce W. Croft. "Query reformulation using anchor text." *Proceedings of the third ACM international conference on Web search and data mining.* ACM, 2010.

[9] Cucerzan, Silviu, and Eric Brill. "Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users." *EMNLP*. Vol. 4. 2004.

[10] Kraft, Reiner, and Jason Zien. "Mining anchor text for query refinement." *Proceedings of the 13th international conference on World Wide Web.* ACM, 2004.

[11] Hassan, A. (2013). Identifying Web Search Query Reformulation using Concept based Matching. In *EMNLP* (Vol. 13, pp. 1000-1010).

[12] Baeza-Yates, R., Hurtado, C., Mendoza, M., & Dupret, G. (2005). Modeling user search behavior. In *Third Latin American Web Congress (LA-WEB'2005)* (pp. 10-pp). IEEE.

[13] Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., & Vigna, S. (2008, October). The query-flow graph: model and applications. In *Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 609-618). ACM.

[14] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, *19*(2), 263-311.

[15] Gao, J., He, X., Xie, S., & Ali, A. (2012, July). Learning lexicon models from search logs for query expansion. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 666-676). Association for Computational Linguistics.

[16] Teevan, J., Adar, E., Jones, R., & Potts, M. (2006, August). History repeats itself: repeat queries in Yahoo's logs. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 703-704). ACM.

[17] White, R. W., Chu, W., Hassan, A., He, X., Song, Y., & Wang, H. (2013, May). Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 1411-1420). ACM.

[18] Anick, P. (2003). Learning noun phrase query segmentation. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 88-95).

[19] Bergsma, S., & Wang, Q. I. (2007, June). Learning Noun Phrase Query Segmentation. In *EMNLP-CoNLL* (Vol. 7, pp. 819-826).

[20] Mitra, M., Singhal, A., & Buckley, C. (1998, August). Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 206-214). ACM.