International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 7– No.6, July 2014 – www.ijais.org

Hybrid Approach for Query Expansion using Query Log

Lynette Lopes
M.E Student,
TSEC, Mumbai, India

Jayant Gadge
Associate Professor,
TSEC, Mumbai, India

ABSTRACT

Web search users usually submit short and ambiguous queries to specify their requirement. In order to improve performance of short and ambiguous queries, query expansion is used. Query expansion is as an effective way to improve the performance of information retrieval systems by adding relevant terms to the original query. After using search engine lots of data get accumulated, from which queries that have been used to retrieve documents are used. This data is stored as query log. These query logs provide valuable information to extract relationships between queries and documents that can be used in query expansion.

This paper proposes method first to determine ambiguous queries using Kullback leibler distance model. It measures difference between two probability distributions. Second, relevant or most suitable expansion terms are selected from the documents with the analysis of relation between queries and documents. The relation can be evaluated by calculating frequency co-efficient with respect to document and document collection.

Keywords

Search engine, information retrieval, query log, ambiguity, expansion terms, co-occurrence, and suitability.

1. INTRODUCTION

Due to short and ambiguity in the user query, retrieving the information as per the intention of user in large volume of web is not straight forward. Because of such ambiguities, search engine generally does not understand in what context user is looking for the information. Hence, it returns huge amount of information, in which most of the retrieved pages are irrelevant to the user. This huge amount of heterogeneous information retrieve not only increases the burden for search engine but also decreases its performance. Consider the query “apple”, which could refer to the fruit, the computer company, a record label, and other less common interpretations [4]. A user interested in one interpretation would not usually be interested in documents relevant to the others.

Query expansion is an alternative way for the system to interact with the users. Given the terms used in the original query and the documents retrieved based on the original query, relevant terms that might be useful for query expansion are suggested. It is the user’s task in such a system to examine the suggested terms and to manually reformulate the query given the information provided by the system. The original query and document related to it is used from query log. Log files of search engines are a promising resource for data mining, since they provide raw data associated to users and web documents [3].

To deal with the mismatching problem at its source, a possible way is to create relationships between the two sets of terms. User logs provide a resource exploitable for this end. Query logs keep track of information regarding interaction between users and the search engine [1].

Logs are valuable resources to explore the search behavior of users and have been found highly useful to improve their search experience. Click-through data is the main mean for capturing users' relevance feedback information. Every single kind of user action can be exploited to derive aggregate statistics which are very useful for the optimization of search engine effectiveness.

By entering ambiguous terms for search, user won’t get satisfactory result. Hence, the user keeps on adding terms which may drift from required result. So, it is important to identify such terms and help user by suggesting expansion terms that will direct him to result page. That may also save user’s search time [9]. A method is proposed which not only recognizes ambiguous terms but also, suggests additional terms to modify original query to get relevant document. Section 2 describes existing work done on query expansion. Section 3 states proposed method. Section 4 represents results and section 5 presents conclusion.

2. LITERATURE SURVEY

Query expansion is rationalized by the fact that initial query formulation does not always reflect the exact information need of the user. There are two key aspects in any query expansion technique: the source from which expansion terms are selected and the method to weight and integrate expansion terms [6].

There are many approaches used for query expansion. All these approaches can be classified into two major classes: global methods and local methods. Global methods are techniques for expanding query terms independent of the query and results returned from it, so that changes in the query wording will cause the new query to match other semantically similar terms [2]. Only individual query terms are considered for expansion.

Global methods include:

- Manual query expansion
- Automatic query expansion
- Interactive query expansion
- Co-occurrence Based Approach for selecting terms for query expansion

Local methods use documents that are retrieved using unmodified query. The basic methods are:

- Relevance Feedback
- Pseudo Relevance Feedback
- Indirect relevance feedback

Manual QE takes place when the user refines the query by adding or deleting search terms without the assistance of the IR system. New search terms may be identified by reviewing previous retrieval results.

Current automatic query expansion techniques can be categorized into global analysis and local analysis. A query expansion method based on global analysis usually builds a thesaurus to assist users reformulating their queries. A thesaurus can be automatically established by analyzing relationships among documents and statistics term co-occurrences in the documents.

In interactive QE, the relevant words are provided to the user in the form of lists and user selects the word that best describes their needs to search the document. These relevant words are obtained from the top relevant documents retrieved from the initial user query. After the user gets feedback it may either expand or reformulate the query.

The term variants for the original query words are used and feed them to user for their consideration. The user needs more information/support to choose better query terms. This is provided by giving the support for real time summaries to the users as they choose a particular term from the list [10].

In Co-occurrence Based Approach, main source of extracting co-occurring terms is the corpus from where documents are coming. The problem with the co-occurrence approach is similar terms identified occur very frequently in the collection and therefore, these terms are not good elements to discriminate between relevant and non-relevant documents. When the co-occurrence analysis is done on the whole collection discrimination does occur to a certain extent only on the top ranked documents [6].

Log-based query expansion deal with the mismatching problem at its source, i.e., the inconsistency problem between the terms used in the documents and those used in the queries, a possible way is to create relationships between the two sets of terms. User logs provide a resource exploitable for this end [3].

Relevance feedback [12] is a straightforward strategy for reformulating queries. In a relevance feedback cycle, the user is presented with a list of initial results. After examining them, the user marks those documents as relevant. The original query is expanded according to these relevant documents. The expected result is that the next round of retrieval will move toward the relevant documents and away from non relevant documents.

Pseudo relevance feedback provides with a mechanism which automates the process of local analysis. No user is involved to make any judgment regarding relevance or irrelevance. The feedback is automatically provided [11]. This helps user to get optimized performance without much involvement and saves time to get to the real results.

In this method, it finds the initial set of most relevant documents and finally does the relevance feedback as before. This uses the top K documents in the search results for the feedback. This may also lead to wrong results as the top K documents may not be relevant to the user query. The top K documents can be among the Wikipedia results for the query and that can be used to generate weighted terms for the query expansion.

Indirect sources of evidence rather than explicit feedback can also be used as the basis for relevance feedback. This is often called implicit feedback. Implicit feedback is less reliable than explicit feedback, but is more useful than pseudo relevance feedback, which contains no evidence of user judgments [12].

Table 1 Comparison of query expansion techniques

Type of query expansion	Source	Feedback type	Term selection done by
Manual	Thesaurus	Relevance	User
Automatic	Relevant documents	Pseudo-relevance	System
Interactive	Relevant documents	Relevance	User
Log-based	Query log	Relevance	User
Co-occurrence	Document corpus	Relevance	User

3. PROPOSED SYSTEM

The short and ambiguous query terms lead into lots of irrelevant documents. In order, to retrieve documents relevant to user query a new approach of query expansion is proposed. It is based on KLD and frequency co-efficient method. To identify ambiguous terms KLD is used.

To select expansion terms, frequency co-efficient method is used. For which, co-occurrence degree of user query term that co-occurs with document term is calculated. Based on that, the term with highest co-occurrence degree is selected. The steps of the proposed system depicted in fig.1 are explained in detail in this section.

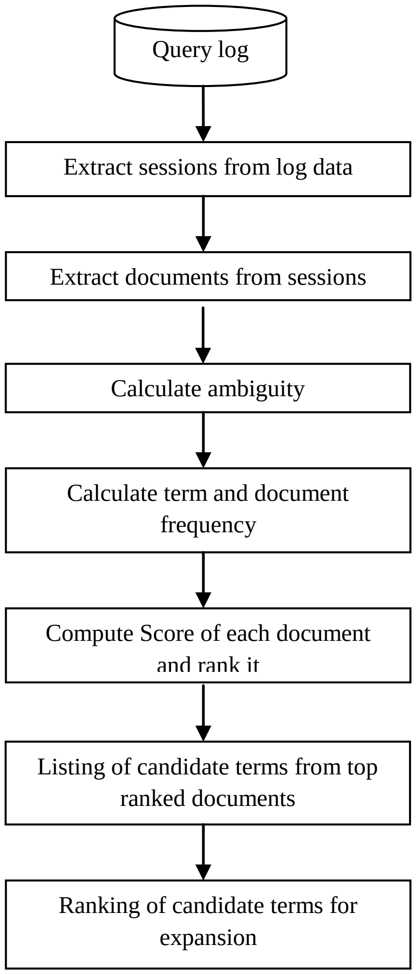
Query log
Extract sessions from log data
Extract documents from sessions
Calculate ambiguity
Calculate term and document frequency
Compute Score of each document and rank it
Listing of candidate terms from top ranked documents
Ranking of candidate terms for expansion

Fig 1: Proposed approach for expanding query

3.1 Preprocessing

The input used is query log. Query logs keep track of information regarding interaction between users and the search engine. They are valuable resources to explore the search behavior of users and have been found highly useful to improve their search experience. Preprocessing of data source is usually independent of the particular user query that is to be expanded but it is specific to the type of data source and expansion method.

One of the useful preprocessing steps is to remove very frequent words that appear in most of the documents and do not bear any meaningful content. They are called stop words.

And stemming the terms before building the inverted index has the advantage that it reduces the size of the index. These steps will be carried out after extracting text from document and creating a dictionary. After which unique id will be assigned to each term based on the page number.

3.2 Extract sessions from query log and documents from sessions

Sessions are created for the searched query term along with the links containing the documents. Each session is represented with a session id. A same query term may appear in two different sessions.Given a query q , get a collection of sessions from log data denoted by

$$S\ (q) = \{s1, s2\ldots\ sn\}\}. \tag{1}$$

The links are crawled to extract documents after which all the terms from the document are entered into a dictionary. And for each keyword from dictionary an id is assigned based on the page number from session to which it belongs.

$$si=\{d1,d2,\ldots,dm\} \tag{2}$$

3.3 Calculate ambiguity

Some terms are not able to retrieve documents correctly due to ambiguous nature. Finding out ambiguous terms i.e. term with more than one meaning is very crucial in such case. The ambiguity is calculated using kullback-leiber divergence method. The calculation of ambiguity degree can be considered as an evaluation of (KLD) Kullback- Leibler distance among these language models.

KLD is used to measure the divergence of two probability distributions in Information Theory, and it also can be used to evaluate the irrelevant degree between two language models. KL divergence basically is a non-symmetric measure of the difference between two probability distributions.

Each session contains a query term and sequence of clicked documents. If the occurrence of query is minimum which will be decided on the basis of threshold value then the query is ambiguous. So, it requires expansion for better results. The ambiguity degree can be calculated by using below equation (3)

$$\mathrm{A(q)} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{\forall \mathrm{j} \neq \mathrm{i}} \left(\frac{\mathrm{KLD}\big(\mathrm{p}(\mathrm{s_i})||\mathrm{p}(\mathrm{s_j})\big) + \mathrm{KLD}(\mathrm{p}(\mathrm{s_i})||\mathrm{p}(\mathrm{s_j}))}{2} \right) \tag{3}$$

Where A (q) is ambiguity of query q,
p(si) is conditional probability for each session,
n is number of sessions.

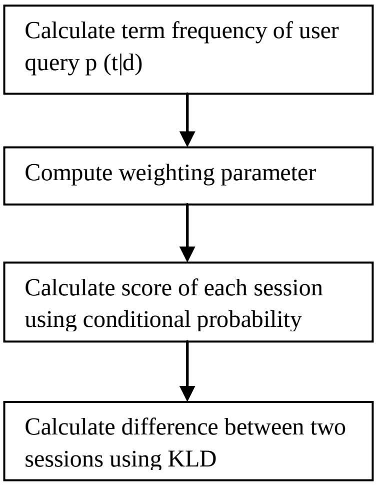
Calculate term frequency of user query p (t|d)
Compute weighting parameter
Calculate score of each session using conditional probability
Calculate difference between two sessions using KLD

Fig.2 Ambiguity calculation

Above fig. 2 states the steps carried out for ambiguity measurement.

3.4 Calculate document and term frequency

The computation of TF values for a web page is straight forward since it simply counts the occurrences for each term within the page. Term frequency is used to calculate the weight of a term in the document. However, some terms occur more rarely and they are more discriminative.

$$\mathrm{tf_i} = \frac{\mathrm{freq_i}}{\text{total no. of terms in document}} \qquad (4)$$

Where tf is term frequency and i is each term from document

The computation of IDF values however is more complex. Two values are mandatory:

1. The overall number of documents in the corpus and

2. The number of documents a term appears in.

If tf values are used, then the term will dominate because it may be more common term. To mitigate this effect, we use inverse document frequency. The idf of a term is the number of documents in the corpus divided by the document frequency of a term.

$$\mathrm{idf_i} = \frac{\log \mathrm{N}}{\mathrm{N_i}} \qquad (5)$$

Where idf is inverse document frequency,

N is total number of documents in corpus and n is documents that contain query term.

3.4 Compute score of each document and rank it

For selecting expansion terms, first document are to be selected from which terms are to be retrieved. Documents are to be selected based on score which is to be calculated using term and document frequency of each term in a document.

The document with highest score will be ranked on top. It will be given higher priority from which expansion terms will be selected. All extracted documents will be used for getting expansion terms.

$$\mathrm{Score(q,d)} = \sum\nolimits_{t\in q\cap \mathrm{d}} \mathrm{tf} * \mathrm{idf} \qquad (6)$$

Where tf is term frequency and idf is inverse document frequency.

The document with highest score is used for selecting expansion terms.

3.5 Listing of candidate terms from top ranked documents

The most likely terms to expand the query are those with a high probability in the top ranked set and low probability in the whole collection. Freq_coefficient, list out the candidate terms from top n documents. These candidate terms can be used for expansion. These coefficients measure the similarity between terms represented by the vectors. However there is a danger in adding these terms directly the query.

The candidate terms selected for expansion should co-occur with the original query terms in the top n documents by chance. The higher its degree is in whole corpus, the more likely it is that candidate term co-occurs with query terms by chance.

$$\mathrm{freq\_co}(\mathrm{t_i},\mathrm{t_j}) = \sum\nolimits_{\mathrm{d}\in \mathrm{D}} [\mathrm{f}(\mathrm{d},\mathrm{t_i}) \times \mathrm{f}(\mathrm{d},\mathrm{t_j})] \qquad (7)$$

Where d is the document from which expansion terms are to be selected,

ti is document term and tj is user query term,

freq_co is frequency co-occurrence of term ti and tj.

$$\mathrm{co\_degree}(\mathrm{c},\mathrm{t_j}) = \log(\mathrm{co}(\mathrm{c},\mathrm{tj}) + 1) * \mathrm{idf(c)}/\log(\mathrm{D}) \quad (8)$$

Where idf (c) is inverse document frequency of candidate term,

D is number of documents in corpus.

3.6 Ranking of candidate terms for expansion

In order to select suitable expansion term, obtain a value measuring how good candidate term is for whole query Q, it's needed to combine degrees of co-occurrence with all individual original query terms. After which score of expansion term is calculated based on suitability of query. Using which it decides whether selected expansion term is having higher weight or not.

$$\mathrm{S(Q)} = \prod\nolimits_{\mathrm{t\ in\ Q}} (\delta + \mathrm{co\_degree}(\mathrm{c},\mathrm{ti})))^{\mathrm{idf}(\mathrm{t_i})} \qquad (9)$$

Where S is Suitability.

δ is a simple smoothing technique.

4. EXPERIMENTAL RESULTS

A total of 15 queries are used to conduct the experiments. Some queries are extracted randomly from the query logs. Some others come from the query log. Other set of queries are added manually by us. The queries used for experiments are very close to those employed by the real web users and the average length of all queries is 2.1 words.

Table 1 Precision and Recall Value

User Query	Before expansion		After expansion	
	Precision	Recall	Precision	Recall
World time	0.5	0.667	0.75	1
Group Home	0.667	0.667	0.667	0.667
Java code	0.5	0.333	1	0.667
Jobs	0.333	0.5	0.667	1
Currency	0.4	0.4	0.6	0.6
Lottery tickets	0.4	0.5	0.8	1
Flower	0.333	0.5	1	0.75
Lyrics	0.5	0.333	0.5	0.6
Flash games	0.333	0.5	0.667	1
Spirit flight	0.333	0.4	0.667	0.8
Coffee flavors	0.6	0.75	0.8	1
Camel tours	0.2	0.4	0.6	0.75
Volleyball college team	0.333	0.333	0.667	0.667
Career guide	0.5	0.6	0.667	0.8
Kids today magazine	0.610	0.714	0.857	0.857
Average	0.443	0.479	0.727	0.792

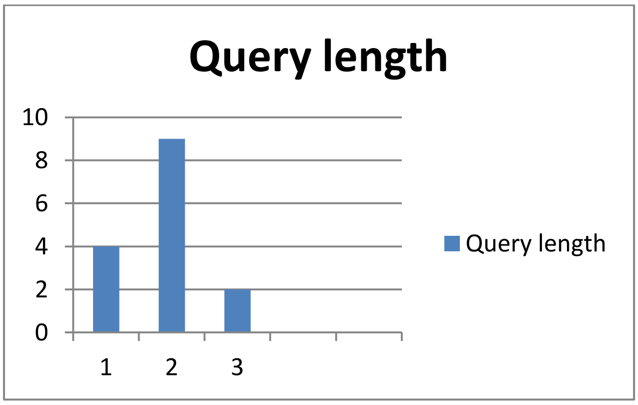

Fig. 3 Proportion of queries

Fig. 3 illustrates the distribution of query lengths based on the number of words. For our experiment, it is observed that 60% of the queries contain only one keyword and 26% of the queries contain two keywords. The average length of all queries is 2.16. The result shows that most people like to use short queries to retrieve information.

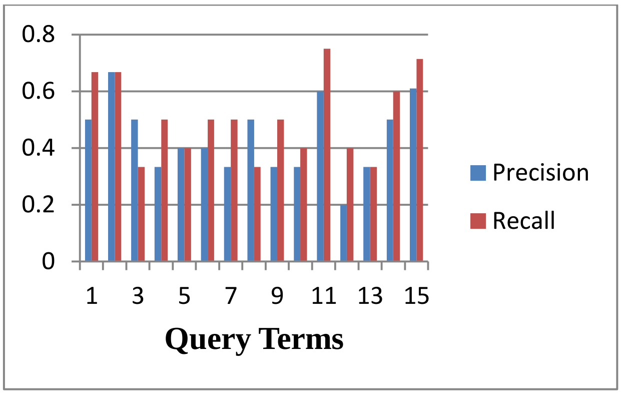

Fig. 4 Precision and Recall before query expansion

The above fig. 4 shows precision and recall value for all the 15 queries before query expansion. Below fig.5 shows precision and recall values after query expansion.

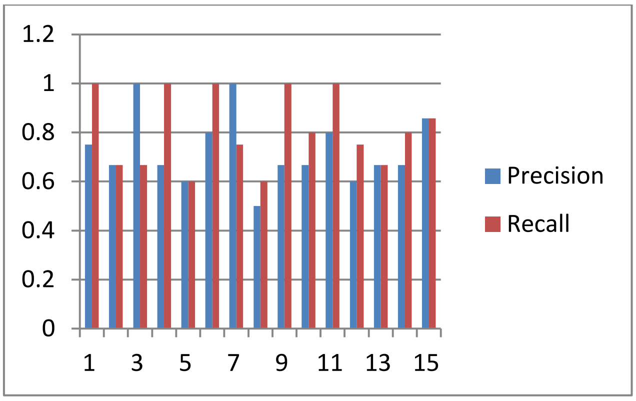

Fig. 5 Precision and Recall after expansion

From above fig. 5 it is observed that the precision and recall values of short queries are low. It is improved after expanding the short query by the terms suggested by proposed method.

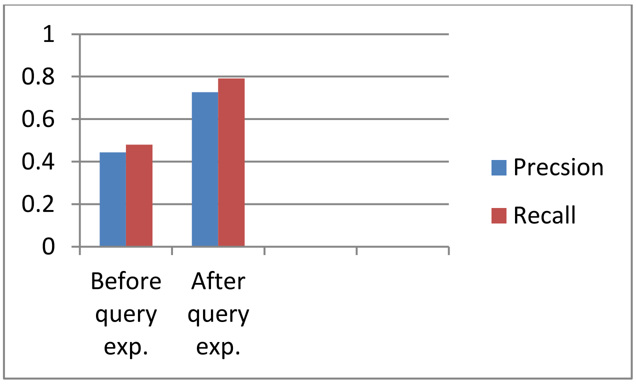

Fig.6 Comparison of Precision and Recall

From above fig. 6 it is observed that the precision and recall values of short queries are low. It is improved after expanding the short query by the terms suggested by proposed method.

5. CONCLUSION

The short and ambiguous query terms lead into lots of irrelevant documents. In order, to retrieve documents relevant to user query a new approach of query expansion is proposed. Itis based on KLD and frequency co-efficient method. To identify ambiguous terms KLD is used. To select expansion terms, frequency co-efficient method is used. For which, co-occurrence degree of user query term that co-occurs with document term is calculated. Based on that, the term with highest co-occurrence degree is selected. Further its suitability value is checked against whole user query. Top five terms are listed for user to select as expansion term.

Based on the data in table 1, it is seen that the precision value varies from 0.2-6 and recall value varies from 0.3-0.7 before expanding original queries. Whereas after expanding queries by our method precision value varies from 0.5-1 and recall value varies from 0.6-1. It is also observed that the average value of precision and recall before expansion is 0.443and 0.479. After query expansion the value of precision and recall is 0.727 and 0.792.

From above improvement in precision and recall values, it can be claimed that the proposed method provides better results in terms of Precision and Recall.

6. REFERENCES

[1] Hang Cui, Ji-Rong Wen,Jian-Yun Nie and Wei-Ying Ma, "*Probabilistic Query Expansion Using Query Logs*"

[2] Yogesh Kakde, "*A Survey of Query Expansion until June 2012*", Indian Institute of Technology, Bombay, 25th June 2012.

[3] Maarten van der, Heijden Max Hinne, Wessel Kraaij, "*Using query logs and click data to create improved document descriptions*"

[4] Zhu Kunpeng, Wang Xiaolong, Liu Yuanchao, "*A new query expansion method based on query logs*" Mining

[5] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China.

[6] Hazra Imran and Aditi Sharan, "*Thesaurus and Query Expansion*", IJCSIT, 2009

[7] Hang Cui, Ji-Rong Wen, Jian-Yun Nie and Wei-Ying Ma, "*Query Expansion by Mining User Logs*", IEEE transactions on knowledge and data engineering, vol. 15, no. 4, July/August 2003.

[8] Ziv BarYossef and Maxim Gurevich, "*Mining Search Engine Query Logs via Suggestion Sampling*".

[9] Burcu Yurekli, Gokhan Capan, Baris Yilmazel and Ozgur Yilmazel, "*Guided Navigation Using Query Log Mining through Query Expansion*".

[10] Rongmei Li, "*Improving Web Page Retrieval using Search Context from Clicked Domain Names*", 20th International workshop on database and expert system application, 2009.

[11] Ketan Singh, "*Study of Different Query Expansion Techniques*", Department of Computer Science and Engineering Indian Institute of Technology, Guwahati, April 2011.

[12] Dippasree Pal, Mandar Mitra, "*Query Expansion Using Term Distribution and Term Association*", Indian Statistical Institute, Kolkata, April 2013.