DYNAMIC QUERY EXPANSION BASED ON USER'S REAL TIME IMPLICIT FEEDBACK

Sanasam Ranbir Singh
Department of CSE, Indian Institute of Technology Guwahati, Assam, India

Hema A. Murthy
Department of CSE, Indian Institute of Technology Madras, Tamil Nadu, India

Timothy A. Gonsalves
Department of CSE, Indian Institute of Technology Mandi, Himachal Pradesh, India

Keywords: Query expansion, Real time implicit feedback, Query log, Relevant term, Search engine.

Abstract: Majority of the queries submitted to search engines are short and under-specified. Query expansion is a commonly used technique to address this issue. However, existing query expansion frameworks have an inherent problem of poor coherence between expansion terms and user's search goal. User's search goal, even for the same query, may be different at different instances. This often leads to poor retrieval performance. In many instances, user's current search is influenced by his/her recent searches. In this paper, we study a framework which explores user's implicit feedback provided at the time of search to determine user's search context. We then incorporate the proposed framework with query expansion to identify relevant query expansion terms. From extensive experiments, it is evident that the proposed framework can capture the dynamics of user's search and adapt query expansion accordingly.

1 INTRODUCTION

Term mismatch between query terms and document terms is an inherent problem that affects the precision of an *information retrieval* (IR) system. Majority of the queries submitted to Web search engines (WSE) are short and under-specified (Jansen et al., 2000; Craig et al., 1999). Short queries usually lack sufficient words to capture relevant documents and thus negatively affect the retrieval performance. Query expansion (QE) is a technique that addresses this issue (Xu and Croft, 1996), where original query is supplemented with additional related terms or phrases.

Existing query expansion frameworks have the problem of poor coherence between expansion terms and user's search goal. For instance, if the query `jaguar` be expanded as the terms {`auto`, `car`, `model`, `cat`, `jungle`,...} and user is looking for documents related to `car`, then the expansion terms such as `cat` and `jungle` are not relevant to user's search goal.

The simplest way to determine user's search goal is to ask users for explicit inputs at the time of search. Unfortunately, majority of the users are reluctant to provide any explicit feedback (Carroll and Rosson, 1987). The retrieval system has to learn user's preferences automatically without any explicit feedback from the user. Query log is a commonly used resource to determine user's preferences automatically without any overhead to the user (Kelly and Teevan, 2003; Agichtein et al., 2006; Joachims, 2002). However, such studies are not flexible enough to capture the changing needs of users over time. If we want to model the complete dynamics of user's preferences from query log, we will need an extremely large query log and huge computational resources. Moreover, user may always explore new search areas. This makes the task of modelling user's search dynamics an extremely difficult and expensive problem.

In this paper, we study a framework to expand user's search query dynamically based on user's implicit feedback provided at the time of search. It is evident from the analysis that, in many instances, user's

Ranbir Singh S., A. Murthy H. and A. Gonsalves T..
DYNAMIC QUERY EXPANSION BASED ON USER'S REAL TIME IMPLICIT FEEDBACK.
DOI: 10.5220/0003104901120121
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 112-121
ISBN: 978-989-8425-28-7

implicit feedback provided at the time of search provides sufficient clues to determine *what user wants*. For example, if the query `jaguar` is submitted immediately after the query `national animals`, it is very likely that user is looking for the information related to `animal`. Such a small feedback can provide a very strong clue to determine user's search preference. This is the main motivation of this paper. From extensive experiments, it is evident that the proposed framework has the potential to expand user's queries dynamically based on user's search pattern.

1.1 Problem Statement

Let q be a query and $\mathcal{E}^{(q)} = \{f_{q,1}, f_{q,2}, f_{q,3}, \ldots\}$ be the set of expansion terms for the query q returned by a traditional query expansion mechanism. In general, many of these expansion terms are not relevant to user's search goal. Now, the task is to identify the expansion terms in $\mathcal{E}^{(q)}$ which are relevant to user's search goal by exploiting user's implicit feedback provided by the user at the time of search.

The rest of the paper is organized as follows. In Section 2, we then discuss background materials. In Section 3, we present few observations of query log analysis which inspire the proposed framework. In Section 4, we discuss our proposed query expansion framework. Section 5 discusses evaluation methodologies. Section 6 present experimental observations. The paper concludes in Section 7.

2 BACKGROUND MATERIALS

2.1 Notations and Definitions

2.1.1 Vector Space Model

We use the vector space model (Salton et al., 1975) to represent a query or a document. A document d or a query q is represented by a *term vector* of the form $\mathbf{d} = \{w_1^{(d)}, w_2^{(d)}, \ldots, w_m^{(d)}\}$ or $\mathbf{q} = \{w_1^{(q)}, w_2^{(q)}, \ldots, w_m^{(q)}\}$, where $w_i^{(d)}$ and $w_i^{(q)}$ are the weights assigned to the i^{th} element of the set d and q respectively.

2.1.2 Cosine Similarity

If $\mathbf{v}_i$ and $\mathbf{v}_j$ are two arbitrary vectors, we use cosine similarity to define the similarity between the two vectors. Empirically, cosine similarity can be expressed as follows.

$$sim(\mathbf{v}_i, \mathbf{v}_j) = \frac{\sum_{k=0}^{m} w_{ik}.w_{jk}}{\sqrt{\sum_{k=0}^{m} w_{ik}^2}.\sqrt{\sum_{k=0}^{m} w_{jk}^2}} \quad (1)$$

2.1.3 Kullback-Leibler Divergence (KLD)

Given two probability distributions p_i and p_j of a random variable, the distance between p_i and p_j can be defined by Kullback-Leibler divergence as follows.

$$KLD(p_i||p_j) = p_i.\log\left(\frac{p_i}{p_j}\right) \quad (2)$$

2.1.4 Density based Term Association

In the study (Ranbir et al., 2008), a *density based term association* (DBTA) is proposed to estimate the term associations. We also use the same estimator in this paper. If W denotes a collection of terms and $tf(x,W)$ denotes term frequency of a term x in W, then the density of word x in W is defined as

$$d(x,W) = \frac{tf(x,W)}{|W|} \quad (3)$$

Further, the combine density of two terms x and y occurring together in W is defined as follows.

$$d(\{x,y\},W) = \frac{\min(tf(x,W), tf(y,W))}{|W| - \min(tf(x,W), tf(y,W))} \quad (4)$$

Let $\lambda(x)$ represents the set of windows [1] containing the word x and $\lambda(x,y)$ represents the set of windows containing both the words x and y. Given a corpora of windows, the relative density score of x and y together in a window is defined as

$$rd(x,y) = \frac{C}{A+B-C} \quad (5)$$

where $C = \sum_{W_i \in \lambda(x,y)} d(\{x,y\},W_i)$, $A = \sum_{W_i \in \lambda(x)} d(x,W_i)$ and $B = \sum_{W_i \in \lambda(y)} d(y,W_i)$. The $rd(x,y)$ represents *how large is the amount of information shared between x and y relative to the space covered by x and y*. Further, the probability of a word y given a word x is defined as follows.

$$Pr(y|x) = \frac{\sum_{W_i \in \lambda(x,y)} d(\{x,y\},W_i)}{\sum_{W_i \in \lambda(x)} d(x,W_i)} \quad (6)$$

This probability represents *how confidently one word associates with another word*. Now, Equations (5) and (6) are combined to define DBTA between two words x and y as follows

$$DBTA(x,y) = Pr(x|y).rd(x,y).Pr(y|x) \quad (7)$$

2.1.5 Real Time Implicit Feedback (RTIF)

In this paper, we differentiate two types of implicit feedback; *history* and *active*. The active implicit feedback are the feedback provided by user at the time of

[1]A window refers to a document or a set of sentences.

search. We also refer to it by *real time implicit feedback* in this paper. A query session has been defined differently in different studies (Jansen et al., 2000; Jaime et al., 2007). This paper considers the definition discussed in (Jansen et al., 2000) and defines as a sequence of query events submitted by a user within a pre-defined time frame. Any feedback provided before the current query session is considered history.

2.2 Background on QE

Global analysis (Jones, 1971; Qiu and Frei, 1993) is one of the first QE techniques where a thesaurus is built by examining word occurrences and their relationships. It builds a set of statistical term relationships which are then used to select expansion terms. Although global analysis techniques are relatively robust, it consumes a considerable amount of computational resources to estimate corpus-wide statistics.

Local analysis techniques use only a subset of the document that is retrieved through an initial ranking by the original query. Thus, it focuses only on the given query and its relevant documents. A number of studies including the ones in (Xu and Croft, 1996; Xu and Croft, 2000; Attar and Fraenkel, 1977; Croft and Harper, 1979) indicate that local analysis is effective, and, in some cases, outperforms global analysis. However, local analysis based query expansion, even with the best of methods, has an inherent inefficiency for reformulating a query, that is, additional online processing for mining expansion terms (Billerbeck and Zobel, 2005).

The above studies focus on document side analysis and they do not take the query side analysis into account. Thus they, in fact, do not address the problem of poor coherence between expansion terms and user's search goal. This paper addresses this issue by exploring user's implicit feedback provided by the user's in real time.

3 FEW MOTIVATING OBSERVATIONS

3.1 Query Log Vs Academic Research

After the AOL incident in August 2006 [2], no query logs are available publicly (not even for academic researches). Obtaining query log from commercial search engines had always been a very difficult task to academic research communities. One alternative for research communities to obtain query log is to use organizational local proxy logs. From the proxy logs, we can extract *in-house* click-through information such as *user's id*, *time of search*, *query*, *click documents* and *the rank of the clicked documents*.

[2] http://www.nytimes.com/2006/08/09/technology/09aol.html

Table 1: Characteristics of the clicked-through log dataset.

Source	Proxy logs
Search Engine	Google
Observation Periods	3 months
# of users	3182
# of query instances	1,810,596
% of clicked queries	53.2%

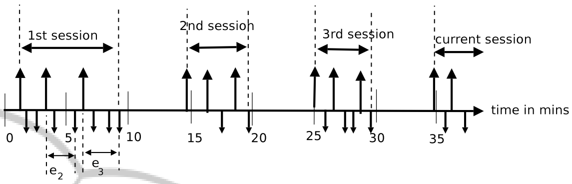


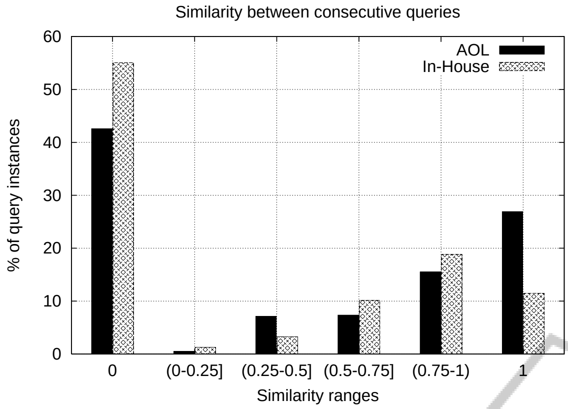
Figure 1: Pictorial representation of the query sessions.

In this study, we use a large proxy log of three months. We extract the queries submitted by the users to google search engine and users' clicked responses to the results. Table 1 shows the characteristics of the click-through query log extracted from the three-months long proxy logs. To prove that the In-House query log has similar characteristics with that of server side query log, we also analyse AOL query log. The analysis described in this paper is strictly anonymous; data was never used to identify any identity.

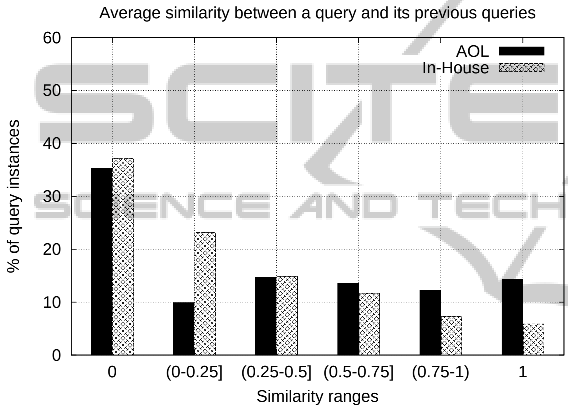
3.1.1 Constructing Query Session

For every user recorded in the query log, we extract sequence of queries submitted by the user. Figure 1 shows a pictorial representation of the procedure to construct query sessions. The upper arrows $\uparrow$ represent the arrival of the query events. Each session is defined by the tuple $\Gamma = < t_{e_{1f}}, uid, E, \delta >$. Just before the arrival of first query from the user u, the first query session has an empty record i.e., $\Gamma = < \phi, u, \phi, \delta >$. When user u submits his/her first query q, Γ is updated as $\Gamma = < t_{e_{1f}}, u, E, \delta >$, where $E = \{e_1\}$, $e_1 = < t_{e_{1f}}, t_{e_{1l}}, q_1, \phi >$, $q_1 = q$ and $t_{e_{1f}} = t_{e_{1l}}$. The down arrows $\downarrow$ in the Figure 1 represent the clicked events. As user clicks on the results for the query q_1, e_1 gets updated as $e_1 = < t_{e_{1f}}, t_{e_{1l}}, q_1, \mathcal{D}^{(q_1)} >$ where $\mathcal{D}^{(q_1)}$ is the set of clicked documents and $t_{e_{1l}}$ is the time of the last click.

When the second query q is submitted by the user u, it forms the second event $e_2 = < t_{e_{2f}}, t_{e_{2l}}, q_2, \phi >$,



(a)



(b)

Figure 2: Similarity between the queries in a query session.

where $q_2 = q$ and $t_{e_{2f}} = t_{e_{2l}}$. If $t_{e_{2f}} - t_{e_{1l}} \leq \delta$, then e_2 is inserted into Γ and E is updated as $E = \{e_1, e_2\}$. If $t_{e_{2f}} - t_{e_{1l}} > \delta$, then e_2 can not be fitted in current query session Γ. In such a case, e_2 generates a new query session with e_2 as its first event i.e., e_2 becomes e_1 and $E = \{e_1\}$ in the new query session. We, then, shift the current session Γ to the newly formed session. In this way, we scan the entire query sequence submitted by the user u and generate the query sessions.

3.2 Exploring Recent Queries

To form the basis of the proposed framework, we analyse two characteristics of user's search patterns during a short period of time defined by a query session: (a) *similarity between recently submitted queries* and (b) *user's topic dynamics*.

3.2.1 Similarity between Queries

In this section, we estimate average similarity between the queries submitted during a query session

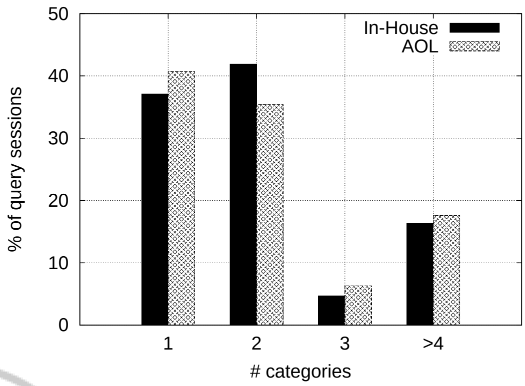


Figure 3: Distribution of the query session with the number of class labels of the visited documents.

(defined by $\delta = 30min$) using cosine similarity defined in Equation (1). Figure 2.(a) shows that almost 55% of the consecutive queries have non-zero similarity (58% for AOL).

Further in Figure 2.(b), we report the average similarity between a query and its previous queries in a session. Almost 65% of the queries have similarity larger than 0. It suggests that a significant number of queries in a session share common search context. Further, two queries with similar search context may have similarity 0. For example, the queries `madagascar` and `die hard 2`. Although, both the queries means movies, their similarity is 0. Therefore, the plots in Figure 2 represent the lower bound.

3.2.2 Topic Dynamics

We further study the distribution of the categories of the documents that user visits during a query session. To study topic dynamics of the user, we first need to assign a label to each of the visited documents. For this task, we have employed a seed-based classifier (the same classifier discussed in (Ranbir et al., 2010)) built over Open Directory Project [3] (ODP). We classify each visited document by the top 15 class labels of ODP.

Figure 3 shows the distribution of the topics that users explore in each query session. It clearly suggests that users visit documents belonging to one or two categories in majority of the query sessions. Only in around 21% of total query sessions for In-House query log (around 24% for AOL query log), users explore more than two categories.

`Remarks:` The above observations (similarity and topic dynamics) show that, in many instances, queries in a session often share common search context. This

[3] www.dmoz.org

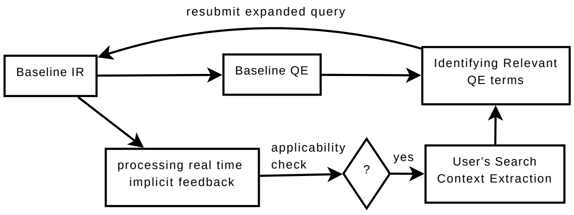


Figure 4: Proposed framework.

motivates us to explore user's real time implicit feedback to determine user's search context.

4 PROPOSED QE FRAMEWORK

To realise the effect of real time implicit feedback on query expansion, we systematically build a framework as shown in Figure 4. It has five major components.

1. `Baseline Retrieval Systems.` It retrieves a set of documents which are relevant with user's query and provides the top most R relevant documents to query expansion unit.
2. `Baseline Query Expansion.` Using the documents provided by the IR system, it determines a list of expansion terms which are related to the query submitted by the user.
3. `Processing real time implicit feedback`: It constructs query session using the procedure discussed in Section 3.1.1.
4. `Applicability Check.` Some query session may not have enough evidences of sharing common search context. This unit verifies whether the newly submitted query shares common search context with that of the other queries in the session.
5. `Determining Search Context.` It determines user's search context by exploiting the implicit feedback provided by the users in the current query session. It then identifies the relevant expansion terms.

4.1 Baseline Query Expansion

We first build a baseline query expansion system over the baseline retrieval system. In this study, we use a KLD (see Equation 2) based QE as discussed in (Billerbeck et al., 2003) as baseline QE. Algorithm 1 shows formal procedure of the baseline QE.

Algorithm 1: Conventional QE through local analysis.

1: run original query q and retrieve relevant documents
2: select top m documents as local set R
3: extracted all terms t from local set R
4: **for all** terms $t \in R$ **do**
5: calculate KLD
6: **end for**
7: rank terms t based on their KLD weight
8: add top $|E|$ terms to original query q
9: run expanded query q and rank documents using PL2

4.2 Determining User's Search Context

Let $\Gamma =< t_{e_{1f}}, u, E, \delta >$ be the current query session as defined in Session 3.1.1, where E is the sequence of n query events. Let $\mathcal{Q}^{(\Gamma)}$ and $\mathcal{D}^{(\Gamma)}$ be the set of queries and visited documents respectively present in E. Let q_{n+1} be a new query submitted by the user u and $\mathcal{E}^{(q_{n+1})} = \{f_{q_{n+1},1}, f_{q_{n+1},2}, f_{q_{n+1},3}, \dots\}$ be the set of expansion terms extracted using Algorithm 1 for the query q_{n+1}. Now the task is to identify relevant terms with that of user's search goal. Algorithm 2 summarises the procedure.

4.2.1 Common Query Terms

This section corresponds to Step 3 of Algorithm 2. It exploits the list of previous queries $\mathcal{Q}^{(\Gamma)}$ submitted by the user in the current query session Γ) and determines the popular query terms using a function $qf(f, \mathcal{Q}^{(\Gamma)})$ which is the number of queries in $\mathcal{Q}^{(\Gamma)}$ containing the term f. We consider a term f popular if its frequency is greater than a threshold i.e., $qf(f, \mathcal{Q}^{(\Gamma)}) \geq \Theta_{\mathcal{Q}}$. In this study, majority of the query sessions are short and the term frequencies are small. Therefore, we set threshold to $\Theta_{\mathcal{Q}} = 1$.

4.2.2 Common Document Terms

This section corresponds to Step 5 of Algorithm 2. Intuitively a popular term among the documents in $\mathcal{D}^{(\Gamma)}$ can also represent user's search context. However, such a term should not only be a good representative term of $\mathcal{D}^{(\Gamma)}$, but also be closely associated with the query. As done in local analysis based query expansion, KLD (as defined in Equation (**??**)) is a good measure to extract informative terms from $\mathcal{D}^{(\Gamma)}$. We estimate association between a query and a term using a density based score function $DBTA(q_{n+1}, f)$ defined in Equation (8). It defines association between two terms $DBTA(f_i, f_j)$. However, q_{n+1} may have more than one term. To estimate association between a query and a term, we use a simple average function as follows:

$$DBTA(q_{n+1}, f) = \frac{1}{|q_{n+1}|} \sum_{f_i \in q_{n+1}} DBTA(f_i, f) \quad (8)$$

where $|q_{n+1}|$ is the number of terms in q_{n+1}.

Algorithm 2: Identify relevant query expansion terms.

1: $\mathcal{E}_{rtif}^{(q_{n+1})} = \phi$
2: **for all** terms $f \in \mathcal{E}^{(q_{n+1})}$ **do**
3: **if** $qf(f, \mathcal{Q}^{(\Gamma)}) \geq \Theta_{\mathcal{Q}}$ {see Section 4.2.1} **then**
4: Insert f in $\mathcal{E}_{rtif}^{(q_{n+1})}$
5: **else if** $\text{score}^{\mathcal{P}^{(\mathcal{D})}}(f) \geq \Theta_{\mathcal{P}^{(\mathcal{D})}}$ {see Section 4.2.2} **then**
6: Insert f in $\mathcal{E}_{rtif}^{(q_{n+1})}$
7: **else if** $f \in \mathcal{P}^{(\mathcal{E})}$ {see Section 4.2.3} **then**
8: Insert f in $\mathcal{E}_{rtif}^{(q_{n+1})}$
9: **else if** $\text{score}^{\mathcal{P}^{(\mathcal{S})}}(f) \geq \Theta_{dbta}$ {see Section 4.2.4} **then**
10: Insert f in $\mathcal{E}_{rtif}^{(q_{n+1})}$
11: **else if** $\text{score}^{\mathcal{P}^{(\mathcal{C})}}(f) \geq \Theta_{\mathcal{P}^{(\mathcal{C})}}$ {see Section 4.2.5} **then**
12: Insert f in $\mathcal{E}_{rtif}^{(q_{n+1})}$
13: **end if**
14: **end for**
15: **for all** terms $f \in \mathcal{E}^{(q_{n+1})}$ and $f \notin \mathcal{E}_{rtif}^{(q_{n+1})}$ **do**
16: **if** $\exists f' \in \mathcal{E}_{rtif}^{(q_{n+1})}$ s.t. $DBTA(f, f') \geq \Theta_{dbta}$ **then**
17: Insert f in $\mathcal{E}_{rtif}^{(q_{n+1})}$
18: **end if**
19: **end for**
20: **return** $\mathcal{E}_{rtif}^{(q_{n+1})}$

Harmonic mean (Sebastiani, 2002) is a popular measure to merge the goodness of two estimators. Therefore, the values of KLD and DBTA are combined using the harmonic mean between the two. However, the two values are at different scales: KLD scales between $-\infty$ to $+\infty$ and DBTA scales between 0 to 1. To make the two estimators coherent to each other, the estimators are further normalized using min-max normalization (Lee, 1995) as follows.

$$\text{normalize}(g) = \frac{g - \min_g}{\max_g - \min_g} \quad (9)$$

where g is an arbitrary function. Now, the harmonic mean score between the two can be defined as follows:

$$\text{score}^{\mathcal{P}^{(\mathcal{D})}}(f) = \frac{2 \cdot KLD^{(\mathcal{D}^{(\Gamma)})}(f) \cdot DBTA(q_{n+1}, f)}{KLD^{(\mathcal{D}^{(\Gamma)})}(f) + DBTA(q_{n+1}, f)} \quad (10)$$

If an expansion terms $f \in \mathcal{E}^{(q_{n+1})}$ has a score greater than a threshold $\Theta_{\mathcal{P}^{(\mathcal{D})}}$ i.e., $\text{score}^{\mathcal{P}^{(\mathcal{D})}}(f) \geq \Theta_{\mathcal{P}^{(\mathcal{D})}}$, then the term f is selected. In this study, the threshold value is set to an arbitrary value 0.5. It is because intuitively the normalized average may cover the upper half of the term collections.

4.2.3 Expansion Terms of Previous Queries

This section corresponds to Step 7 of Algorithm 2. Let $e_i =< t_{e_{if}}, t_{e_{il}}, q_i, \mathcal{D}_c^{(q_i)} >$ be a query event in E, where $i \neq n+1$ and $\mathcal{E}^{(q_i)}$ be the expansion terms of the query q_i. If an expansion term $f \in \mathcal{E}^{(q_i)}$ is also present in any document $d \in \mathcal{D}_c^{(q_i)}$, then it is selected. The set of such terms is denoted by $\mathcal{P}_i^{(\mathcal{E})}$ and is formally define as follows:

$$\mathcal{P}_i^{(\mathcal{E})} = \{ f | f \in \mathcal{E}^{(q_i)} \text{ and } \exists d \in \mathcal{D}_c^{(q_i)} \text{ s.t. } f \in d \} \quad (11)$$

We assume that the visited documents against a query are relevant to user's information need of that query. Therefore, this set represents the set of expansion terms of previous queries in the same query session which are actually relevant to user's search goal. For all the queries in $\mathcal{Q}^{(\Gamma)}$, Equation (11) is repeated and all $\mathcal{P}_i^{(\mathcal{E})}$ are merged i.e., $\mathcal{P}^{(\mathcal{E})} = \cup \mathcal{P}_i^{(\mathcal{E})}$. An expansion term $f \in \mathcal{E}^{(q_{n+1})}$ is assumed to be relevant to user's current search context, if $f \in \mathcal{P}^{(\mathcal{E})}$.

4.2.4 Synonyms of Query Terms

This section corresponds to Step 9 of Algorithm 2. There are publicly available tools like `Wordnet` [4], `WordWeb` [5] which can provide synonyms of a given term. Such expert knowledge can be used effectively to select the expansion terms.

Let $P^{(\mathcal{S})}$ be the list of synonyms [6] for all the query terms in $\mathcal{Q}^{(\Gamma)}$ extracted using Wordnet. If an expansion terms $f \in \mathcal{E}^{(q_{n+1})}$ has an score greater than a threshold Θ_{dbta} i.e., $\text{score}^{\mathcal{P}^{(\mathcal{S})}}(f) \geq \Theta_{dbta}$, then the term f is considered to be relevant to user's search goal.

$$\text{score}^{\mathcal{P}^{(\mathcal{S})}}(f) = \begin{cases} DBTA(f, f'), & \text{if } f \in \mathcal{P}^{(\mathcal{S})} \text{ and} \\ & \exists f' \in \mathcal{P}^{(\mathcal{E})} \text{ s.t.} \\ & DBTA(f, f') \geq \Theta_{dbta} \\ 0, & \text{Otherwise} \end{cases} \quad (12)$$

[4]http://wordnet.princeton.edu

[5]http://wordweb.info/free/

[6]We apply the Wordnet command `wn auto synsn` to get list of synonyms. We pass the output of this command to a script. This script processes the output and returns the list of synonyms.

In this study, the threshold Θ_{dbta} is set to an arbitrary value i.e., the average value of $DBTA(f, f')$ over the corpus. However, more sophisticated procedure to set threshold value will be to study the distribution of positive and negative associations.

4.2.5 Category Specific Terms

This section corresponds to Step 11 of Algorithm 2. Another important information that can be extracted from implicit feedback is dominant class labels in $\mathcal{D}^{(\Gamma)}$. It is observed in Section 3.2.2 that users often confine their searches to a small number of class labels. We also expect that majority of the documents in $\mathcal{D}^{(\Gamma)}$ confine to few dominant class labels. The relevant expansion terms should have close association with the dominant class labels. In the study (Ranbir et al., 2010), the authors studied a new measure known as *within class popularity* and it is observes that WCP provides better association as compared to other estimators such as *mutual information*, *chi-square* (Yang and Pedersen, 1997). In this study, we use the same measure WCP to estimate association between a term and class.

If C be the set of global class labels and $C^{(\Gamma)}$ be the set of dominant class labels of the current query session Γ. We select a term $f \in \mathcal{E}^{(q_{n+1})}$ if $\exists c \in C^{(\Gamma)}$ such that

$$c = \max_{\forall c_i \in C} \{wcp(f, c_i)\} \tag{13}$$

where

$$wcp(f, c_i) = \frac{Pr(f|c_i)}{\sum_{k=1}^{|C|} Pr(f|c_k)} \tag{14}$$

4.2.6 Mining more Context Terms

This section corresponds to the steps 15 to 19 in Algorithm 2. Let $\mathcal{E}_{rtif}^{(q_{n+1})}$ be the set of relevant expansion terms thus obtained from the above sections. Still there may be terms in $\mathcal{E}^{(q_{n+1})}$ which are not included in $\mathcal{E}_{rtif}^{(q_{n+1})}$, but closely related to some terms in $\mathcal{E}_{rtif}^{(q_{n+1})}$. Intuitively, such missing terms are also related to the context of user's search goal. Therefore, we further determine missing terms as follows:

- for all terms $t \in \mathcal{E}^{(q_{n+1})}$ and $t \notin \mathcal{E}_{rtif}^{(q_{n+1})}$: if $\exists t' \in \mathcal{E}_{rtif}^{(q_{n+1})}$ s.t. $DBTA(t, t') > \Theta_{dbta}$, then insert the term t in $\mathcal{E}_{rtif}^{(q_{n+1})}$.

Now, we consider the terms in $\mathcal{E}_{rtif}^{(q_{n+1})}$ as the expansion terms related to the context of user's search goal.

4.3 Applicability Check

The above procedures to identify relevant expansion terms will return good results if the newly submitted query q_{n+1} indeed has the same search preference as that of other queries in E. But this condition is not always true. In some query sessions, there may not be enough evidences of having common search context.

Therefore, it is important to perform an applicability check before applying the above procedures. For every newly submitted query q_{n+1}, we perform an applicability check. We estimate average cosine similarity among the expanded terms of all queries in the session. If the average similarity of a current session is above a user-defined threshold Θ_{sim}, then it is assumed that the queries in the current query session share common search context.

5 EVALUATION METHODOLOGY

To evaluate the proposed framework we define three metrics – (i)*quality*: measure the quality of expansion terms, (ii) *precision@k*: measure retrieval effectiveness and (iii) *dynamics*: measure the capability of adapting to the changing needs of users.

The best evidence to verify the quality of the expanded terms or retrieval effectiveness of a system is to cross check with the documents actually visited by the user for the subjected query. Let q be an arbitrary query and $\mathcal{D}_c^{(q)}$ be the set of documents actually visited by the user for q. Now, given an IR system and a query expansion system, let $\mathcal{E}^{(q)}$ be the set of expansion terms for the query q. Then, the quality of the expansion terms is defined as follows:

$$quality = \frac{|\rho(\mathcal{E}^{(q)}, \mathcal{D}_c^{(q)})|}{|\mathcal{E}^{(q)}|} \tag{15}$$

where $\rho(\mathcal{E}^{(q)}, \mathcal{D}_c^{(q)})$ is the matching terms between $\mathcal{E}^{(q)}$ and $\mathcal{D}_c^{(q)}$ i.e.,

$$\rho(\mathcal{E}^{(q)}, \mathcal{D}_c^{(q)}) = \{f | f \in \mathcal{E}^{(q)}, \exists d \in \mathcal{D}_c^{(q)} \text{ s.t. } f \in d\}$$

Let $\mathcal{D}_n^{(q)}$ be the set of top n documents retrieved by the IR system. To define retrieval effectiveness, we determine the number of documents in $\mathcal{D}_n^{(q)}$ which are closely related to the documents in $\mathcal{D}_c^{(q)}$. We use cosine similarity (see Equation (1)) to define the closeness between two documents. Let $\mathcal{D}_r^{(q)}$ be a set of documents in $\mathcal{D}_n^{(q)}$ for which the cosine similarity

Table 2: List of the 35 queries. $\#\Gamma$ indicates number of query sessions and $\#\mathcal{Z}$ indicates the number different search context.

query	$\#\Gamma$	$\#\mathcal{Z}$	query	$\#\Gamma$	$\#\mathcal{Z}$	query	$\#\Gamma$	$\#\mathcal{Z}$	query	$\#\Gamma$	$\#\mathcal{Z}$
blast	15	1	books	18	4	chennai	18	3	coupling	10	2
crunchy munch	38	1	indian	14	2	games	59	1	jaguar	3	2
kate winslet	23	2	mallu	38	1	milk	15	2	namitha	22	1
nick	20	1	rahaman	2	1	passport	38	2	roadies	10	1
statics	36	4	times	5	2	science	16	2	scholar	16	3
simulation	3	1	smile pink	2	1	tutorial	11	6	reader	11	3
ticket	38	3	crank	10	1	engineering village	12	1	maps	15	4
nature	28	2	reshma	15	1	savita	2	1	dragger	11	2
sigma	11	2	spy cam	10	1	java	17	2			

with at least one of the document in $\mathcal{D}_c^{(q)}$ is above a threshold Θ_{sim} i.e.,

$$\mathcal{D}_r^{(q)} = \{d_i | d_i \in \mathcal{D}_n^{(q)}, \exists d_j \in \mathcal{D}_c^{(q)}\ s.t.\ sim(d_i, d_j) \geq \Theta_{sim}\}$$

In this study we define $\mathcal{D}_r^{(q)}$ with the threshold value $\Theta_{sim} = 0.375$. In our dataset, the majority of the co-click documents have cosine similarity in the range of [0.25,5). We have considered the middle point as the threshold value. Now we use the *precision@k* to measure the retrieval effectiveness and define it as follows:

$$\text{precision@}k = \frac{|\mathcal{D}_r^{(q)}|}{k} \tag{16}$$

Last we define the dynamics in query expansion. For a query, the system is expected to return different expansion terms for different search goals. Let $\mathcal{E}_i^{(q)}$ and $\mathcal{E}^{(q)})_j$ be the set of expansion terms for a query q at two different instances i and j. Then we define the dynamics between the two instances as follows:

$$\delta^{(q)}(i,j) = 1 - sim(\mathcal{E}_i^{(q)}, \mathcal{E}_j^{(q)}) \tag{17}$$

If there are n instances of the query q then we estimate the average dynamics as follows

$$E(\delta^{(q)}(i,j)) = \frac{n(n-1)}{2} \sum_{i \neq j} \delta^{(q)}(i,j) \tag{18}$$

6 PERFORMANCE OF THE PROPOSED FRAMEWORK

We build two baseline retrieval systems (i) an IR system which indexes around 1.6 million documents using PL2 normalization (He and Ounis, 2005), denoted by *LIR*, and (ii) a meta-search interface which receives queries from the users and submit it to Google search engine, denoted by *GIR*. On top of these systems, we have incorporated the proposed framework.

Table 3: Average quality of the top 20 expansion terms over 35 queries given in Table 2.

Baseline		Proposed	
LIR	GIR	LIR	GIR
0.287	0.329	0.536(+86.7%)	0.562(+70.8%)

6.1 Experimental Queries

A total of 35 queries are selected to conduct the experiments from the In-House query log discussed in Section 3.1. Top most popular *non-navigational* queries (Broder, 2002) of length 1 and 2 words are selected.

Table 2 shows the list of 35 selected queries. This table also shows the number of query sessions for each of the individual queries and denoted by ”#”. A total of 612 query sessions are found for these 35 queries. A query may have different search goals at different times. We manually verify and mark all these 612 instances. While verifying we broadly differentiate the goals (e.g. ”java programming” and ”java island” are two different goals, however ”java swing” and ”core jave” have same goal). Table 2 also shows the number of different search goals for individual query (denoted by ”$\#\mathcal{Z}$”). It shows that 20 out of 35 (i.e., 57.1%) queries have varying search preferences at different times.

6.2 Quality of Expansion Terms

Table 3 shows the average *quality* of the expansion terms over all 35 queries. There is a significant improvement in quality. On an average there is an improvement from 0.287 to 0.536 (86.7% improvement) on local IR system. For the Google meta search, there is an improvement of 70.8% from 0.329 to 0.562.

6.3 Retrieval Effectiveness

Now, we compare the retrieval effectiveness of the proposed expansion mechanism with the baseline expansion mechanism. We use the precision at k measure (defined in Equation (16)) to estimate retrieval effectiveness. In Table 4, we compare the retrieval performance of the baseline system and the proposed system in terms of the average of the precision at k for all 612 query instances. If a query has no visited documents, we simply ignore them. Note that, the set of visited documents $\mathcal{D}_c^{(q)}$ is obtained from the query log whereas the set $\mathcal{D}_n^{(q)}$ is obtained from the experimental retrieval system after simulating the query sequence. Table 4 clearly shows that our proposed framework outperforms the baseline systems for both the local IR system and Google results.

Table 4: Precision@k returned by different systems using top 20 expansion terms.

top k	Baseline		Proposed	
	LIR	GIR	LIR	GIR
10	0.221	0.462	0.749	0.763
20	0.157	0.373	0.679	0.710
30	0.113	0.210	0.592	0.652
40	0.082	0.153	0.472	0.594
50	0.052	0.127	0.407	0.551

6.4 Component Wise Effectiveness

In the section 4.2, we define different components that contribute to the expansion terms. In this section, we study the effect of each component separately. Table 5 shows the quality of the expansion terms returned by each component (considering the top 20 expansion terms). In the table, $P^{(\mathcal{Q})}$ denotes set of expansion terms based on query terms (Section 4.2.1), $P^{(\mathcal{D})}$ denotes the document terms (Section 4.2.2), $P^{(\mathcal{E})}$ denotes combine expansion terms of previously submitted queries (Section 4.2.3), $P^{(\mathcal{S}^{\mathcal{R}})}$ denotes word sense (Section 4.2.4)and $P^{(\mathcal{C})}$ denotes class specific terms (Section 4.2.5). We observe that expansion terms extracted using $P^{(\mathcal{D})}$ and $P^{(\mathcal{E})}$ contribute the most. This observation is true for both the local retrieval system and Google results. The summation of the percentages in each row is more than 100%. It is because, there are overlapping terms among the components.

6.5 Retrieval Efficiency

Though the proposed framework provides better retrieval effectiveness, it has an inherent efficiency problem. Apart from the time required for query

Table 5: Average quality of individual components over 35 queries given in Table 2.

	$P^{(\mathcal{Q})}$	$P^{(\mathcal{D})}$	$P^{(\mathcal{E})}$	$P^{(\mathcal{S}^{\mathcal{R}})}$	$P^{(\mathcal{C})}$
LIR	8.3%	39.8%	37.9%	4.6%	12.1%
GIR	8.8%	43.3%	39.2%	6.9%	8.4%

Table 6: Average retrieval efficiency of different expansion system in seconds.

Baseline IR		Baseline QE		Proposed QE	
LIR	GIR	LIR	GIR	LIR	GIR
1.028	0.731	3.961	3.205	14.518	14.149

expansion (Algorithm 1), the proposed framework needs computational time for determining context for user's search goal. Table 6 shows the efficiency of different retrieval systems. It clearly shows that the proposed framework has poor efficiency. It can be noted that the computational overhead is an order of magnitude higher than that of general expansion and without expansion.

The focus of this paper is to investigate feasibility of query expansion dynamically by exploiting real time implicit feedback provided by the users at the time of search. There will be additional computational overhead to process the expansion in real time. The implementation of the experimental systems are not optimal. Though the computational overhead reported in Table 6 is high, with efficient programming and hardware supports we believe that the overhead can be reduced to reasonable level.

6.6 Dynamics

Table 7 shows the average of the average dynamics of different systems over all experimental queries. It clearly shows that the baseline system has a dynamics of zero in all cases. It indicates that baseline systems always return the same expansion terms irrespective of user's search goal. Whereas the proposed framework has a small dynamics among the instances of the same query with same goal and high dynamics among the query instances of the same query with different goals. It indicates that the proposed framework is able to adapt to the changing needs of the users and generate expansion terms dynamically.

Table 7: Average of average dynamics over all queries.

	Baseline QE		Proposed QE	
Goal	LIR	GIR	LIR	GIR
Same	0	0	0.304	0.294
Different	0	0	0.752	0.749

7 CONCLUSIONS

In this paper, we explore user's real time implicit feedback to analyse user's search pattern during a short period of time. From the analysis of user's click-through query log, we observe two important search patterns – user's information need is often influence by his/her recent searches and user's searches over a short period of time often confine to 1 or 2 categories. In many cases, the implicit feedback provided by the user at the time of search have enough clues of what user wants. We explore query expansion to show that the information submitted at the time of search can be used effectively to enhance search retrieval performance. We proposed a query expansion framework, which explores recently submitted query space. From various experiments, we observed that the proposed framework provides better relevant terms compared to the baseline query expansion mechanisms. Most importantly, it can dynamically adapt to the changing needs of the user.

REFERENCES

Agichtein, E., Brill, E., Dumais, S., and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. In *SIGIR'06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM.

Attar, R. and Fraenkel, A. S. (1977). Local feedback in full-text retrieval systems. *Journal of ACM*, 24(3):397–417.

Billerbeck, B., Scholer, F., Williams, H. E., and Zobel, J. (2003). Query expansion using associated queries. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*. ACM.

Billerbeck, B. and Zobel, J. (2005). Document expansion versus query expansion for ad-hoc retrieval. In *ADCS '05: Proceedings of the tenth Australasian document computing symposium*.

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36(2):3–10.

Carroll, J. M. and Rosson, M. B. (1987). Paradox of the active user. In *Interfacing thought: cognitive aspects of human-computer interaction*, pages 80–111. MIT Press.

Craig, S., Monika, H., Hannes, M., Monika, H., and Michael, M. (1999). Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12.

Croft, W. B. and Harper, D. J. (1979). Using probabilistic model of document retrieval without relevance information. *Journal of Documentation*, 35:285–295.

He, B. and Ounis, I. (2005). Term frequency normalisation tuning for bm25 and dfr model. In *ECIR'05: Proceedings of the 27th European Conference on IR Research*, pages 200–214.

Jaime, T., Eytan, A., Rosie, J., and Michael, A. S. P. (2007). Information re-retrieval: Repeat queries in yahoo's logs. In *SIGIR07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 151–158. ACM.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *SIGKDD'02: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM.

Jones, S. (1971). *Automatic keyword classification for information retrieval*. Butterworths, London, UK.

Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 32(2):18–28.

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 180–188, New York, NY, USA. ACM.

Qiu, Y. and Frei, H. (1993). Concept based query expansion. In *SIGIR93: Proceeding of the 16th International ACM SIGIR Conference on Research and development in information retrieval*, pages 151–158. ACM.

Ranbir, S. S., Murthy, H. A., and Gonsalves, T. A. (2008). Effect of word density on measuring words association. In *ACM Compute*, pages 1–8.

Ranbir, S. S., Murthy, H. A., and Gonsalves, T. A. (2010). Feature selection for text classification based on gini coefficient of inequality. In *FSDM'10: Proceedings of the Fourth International Workshop on Feature Selection in Data Mining*, pages 76–85.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *ACM Communication*, 18(11):613–620.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Survey*, 34(1):1–47.

Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *SIGIR'96: Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11.

Xu, J. and Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transaction on Information System*, 18(1):79–112.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML'97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420. ACM.