Exploiting Real-Time Information Retrieval in the Microblogosphere

Feng Liang
liangfeng@pku.edu.cn

Runwei Qiang
qiangrw@gmail.com

Jianwu Yang*
yangjw@pku.edu.cn

Institute of Computer Science and Technology
Peking University,Beijing 100871,China

ABSTRACT

Information seeking behavior in microblogging environments such as Twitter differs from traditional web search. The best performing microblog retrieval techniques attempt to utilize both semantic and temporal aspects of documents. In this paper, we present an effective approach, including the query modeling, the document modeling and the temporal re-ranking, to discover the most recent but relevant information to the query. For the query modeling, we introduce a two-stage pseudo-relevance feedback query expansion to overcome the severe vocabulary-mismatch problem of short message retrieval in microblog. For the document modeling, we propose two ways to expand document with the help of the shortened URL. For the temporal re-ranking, we suggest several methods to evaluate the temporal aspects of documents. Experimental results demonstrate that our approach obtains significant improvements compared with baseline systems. Specifically, the proposed system gives 26.37% and 9.94% further increases in P@30 and MAP over the best performing result on *highrel* in the TREC'11 Real-Time Search Task.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

Algorithms, Experimentation, Performance

Keywords

Real-Time Search, Query Expansion, Language Model, Temporal Search

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
JCDL'12, June 10–14, 2012, Washington, DC, USA.
Copyright 2012 ACM 978-1-4503-1154-0/12/06 ...$10.00.

1. INTRODUCTION

Microblog is a light-weight, easy form of communication that enables users to broadcast and share information about their activities, opinions and statuses [10]. Microblog is provided by several internet services such as Twitter[1], Facebook[2] and Tumblr[3]. One of the most popular microblogging platforms is Twitter [18]. A microblog consists of individual posts and in Twitter, particularly, a user's post is called tweet. In Twitter, text is strictly under the length limitation of 140 characters. To enrich the information in Twitter, users usually post a shortened URL within a tweet. Besides, a tweet includes a group of well-defined symbols: RT denotes retweet, '@' followed by a user's screen name stands for mentioning or replying to the specific user, '#' symbol (i.e. hashtag) is used for organizing tweets into a particular topic.

User's information need for microblog is expressed in semantic and temporal aspects. For the semantic aspect, user specifies a small number of keywords as a query to search tweets, saying from this perspective, user wants to obtain a list of relevant tweets. However, for the temporal aspect,the same query with different timestamps may result in different retrieval results. For instance, when a user issued the query "Kubica crash" at the very beginning of a high-speed rally crash in Italy, he may just want to check whether this news is true. As time passes by, a burst of tweets were posted, and user entered the same query to keep informed of this accident, e.g. how badly Kubica was injured. Finally, as the number of tweets about this accident decreased, user typed the same query to review the whole story of this incident. This implies that temporal profile may be helpful for expressing user's information need in the microblogosphere. From this perspective, user would like to seek the most recent tweets.

To explore the information seeking behavior in microblogging environments, TREC introduced a novel pilot task named Real-Time Search Task (RTST) at the TREC'11 Microblog Track, aiming to "find the most recent but relevant information to the query" [17]. The real-time search task can be summarized as "*At time t, find tweets about topic X*". This task is akin to ad-hoc search on Twitter, where a user's information need is represented by a query at a specific time.

Some new characteristics keep RTST distinctive from previous ad-hoc tasks such as Web Ad-hoc Search. First, we

[1]http://twitter.com
[2]http://www.facebook.com
[3]http://www.tumblr.com/

are supposed not to use all the data in the corpus but only to take advantage of information available at the time the query is issued. Therefore, systems for RTST should build a dynamic dataset for each query. Second, RTST is confronted with the difficulties posed by the severe vocabulary-mismatch problem rooted in the sparsity of documents (i.e. tweets). For this reason, expansion techniques should be applied to enrich the representation of query and document. Third, participants are required to rank the returned tweets by timestamp instead of by relevance score. Hence, we should make a trade off between relevance and recentness, otherwise, re-ranking by time will sharply decrease the search effectiveness.

To solve these challenges,we present an effective approach that can improve the performance of the proposed system for RTST. The main contributions of this paper are: (1) we describe a two-stage pseudo-relevance feedback query expansion to estimate a query language model. (2) we propose two ways to expand document with the help of shortened URL to enrich the representation of document. (3) we suggest several temporal re-ranking functions and two representations of temporal profile to evaluate the temporal aspects of documents. In a set of experiments performed over TREC'11 Real-Time Search Task test collection, we compare the proposed system against both the baseline systems and the best performing systems in the TREC'11 RTST, and the experimental results demonstrate that the proposed approach delivers a significant retrieval performance.

The remainder of paper is organized as follows: In Section 2, we give an overview of related work. In Section 3, we introduce the framework of our system, and the proposed methods to search information in the microblogosphere in detail. In Section 4, we conduct a series of experiments and evaluate the performance of the proposed methods. Finally, in Section 5, we conclude the paper and outline our future work.

2. RELATED WORK

As a new pilot task, RTST aims to retrieve information in real-time environment, and to provide a ranked list of tweets from the latest timestamp to the earliest timestamp. There are few research findings for ad-hoc search in the microblogosphere. Three most relevant research topics to our work are pseudo-relevance feedback query expansion, sentence retrieval and temporal aspects analysis in IR and TDT.

Pseudo-relevance feedback technique has been widely used in various retrieval models [21, 19, 22, 2, 20, 11, 25]. In the past few years, several attempts have been made in developing pseudo-relevance feedback in language model framework [11, 25, 14, 23]. In the mixture-model feedback [25], words in feedback documents were supposed to be sampled from two models: (1) background language model and (2) to-be-discovered topic model. The mixture-model feedback revealed that background language model is reasonable of irrelevant information in feedback document while topic model is responsible for generating topically related words. The expanded query was then updated by incorporating original query model with topic model. Relevance model [11] improved query language model by first computing the joint probability of observing a word together with words in original query in a single feedback document and then summing the evidence over all the document set. Finally, documents were weighted by query likelihood $P(Q|D)$ and probability of word given by each document language model.

Both Sentence Retrieval (SR) and RTST suffer severely from the vocabulary-mismatch problem [8] because there is little overlap between the query and document terms. There was some previous work that exploits local context to enrich the representation of document (i.e. sentence) [8, 16, 13]. Losada and Fernández's work [13] informally introduced the local context of a sentence into the computation of sentence similarity. Essentially, terms that have high frequency in the associated documents were assigned an extra weight. Murdock [16] estimated sentence language model by utilizing some local context and combing them with evidence from the sentence and document level.

Previous work has shown that temporal evidence can be incorporated into IR techniques [5, 7], particularly in the language model framework [12]. Li and Croft proposed taking advantage of the publish date of news articles under an exponential distribution to estimate the document prior [12]. Dakka, Gravano and Iperirotis suggested that the standard query likelihood framework should take time into consideration [5]. They divided a given document into two components: (1) lexical terms in document and (2) timestamp of document. Moreover, temporal evidence was also used in New Event Detection Task in TDT [3, 4]. Chen et al proposed an aging theory to improve the performance for event detection [3]. Chen et al also used the aging theory combined with a sentence modeling to extract hot topics from news articles [4].

To our knowledge, (1) two-stage pseudo-relevance feedback query expansion using different estimations of query model in each stage has not been previously seen; (2) document expansion by refining the representation of tweet with the shortened URLs has not been described previously; (3) the two representations of temporal profile described in this paper is novel but effective to estimate the temporal aspects of documents in the microblogosphere.

3. REAL-TIME TWEET RANKING MODEL

To mine valuable information in the microblogosphere, we rank tweets by evaluating their importance to the given query at a specific timestamp. We conclude several criteria to estimate the importance of a tweet as follows:

- **Accuracy** Tweets that are accurate to feed user's information need are important.
- **Abundance** A high-quality URL within the tweet can enrich the representation of tweet, thus to improve the importance.
- **Timeliness** Tweets that were posted an hour ago are often more worthy than those that were updated a day before even though the former are not as much accurate as the latter.
- **Significance** A tweet that has been re-tweeted a lot of times or shared by some key users in the social network would spread fast in the microblogosphere, as a result, its importance would be increased.
- **Trend** People mentioned via '@' symbol and topics that tweets are organized into through '#' symbol also affect the importance of tweets.

Considering the lack of some crucial metadata such as user's profile in the HTML version of Tweet11 Corpus [17], we determine to adopt the first three criteria to measure the importance of a tweet. We assume that a tweet satisfies the criteria well when it occurs near the query timestamp and, at the same time, receives a high relevance score with respect to the query.

Motivated by our assumption, we propose our real-time tweet ranking model under the language model framework which is widely used in various IR tasks. Given the RTST problem, the proposed approach is to estimate the probability of generating a query Q given the content D and timestamp t of the tweet as follows:

$$P(Q|D,t) = \frac{P(t|Q,D) \cdot P(Q|D)}{P(t|D)} \quad (1)$$

Assuming that $P(Q|D) \propto Score(Q,D)$ which can be calculated using KL-divergence retrieval model [24], and that $P(t|D)$ can be assumed as a constant because it is query-independent, we rewrite the ranking formula as follows:

$$\begin{aligned} P(Q|D,t) &\propto P(t|Q,D) \cdot P(Q|D) \\ &\propto P(t|Q,D) \cdot Score(Q,D) \\ &= P(t|Q,D) \cdot \sum_{w \in V} P(w|\hat{\theta}_Q) \cdot \log P(w|\hat{\theta}_D) \quad (2) \end{aligned}$$

With the ranking formula, the retrieval task is reduced to three subtasks, i.e. the estimation of query model $\hat{\theta}_Q$, the estimation of document model $\hat{\theta}_D$ and the temporal re-ranking component $P(t|Q,D)$, respectively.

3.1 The Estimation of Query Model

Thanks to the flexibility that KL-divergence retrieval model offers, we can reformulate query model $\hat{\theta}_Q$ by taking pseudo-relevance feedback information into account. Much like most of pseudo-relevance feedback (PRF) approaches do, we consider the top-ranked document, which we call the support tweet, to expand the original query. However, the main difference between previous PRF methods and our approach derives that we employ a two-stage PRF query expansion with the help of the support tweet to estimate different feedback models in each stage. The motivations of our two-stage PRF query expansion can be concluded as follows:

- User tends to express a single topic in one document(i.e. tweet), due to the 140-character length limitation in Twitter. Hence, a single tweet can be utilized to generate accurate topical words to expand the original query.
- Top-ranked document set can enrich the feedback information, but may include noisy topics due to the severe vocabulary-mismatch problem in the initial retrieval. Therefore, relevant information should be discriminated from the irrelevant one in the pseudo-relevant document set.

Inspired by the motivations, we employ our two-stage PRF query expansion in the following steps: (1) In the first stage, we pick up a single support tweet to generate accurate topical words, helping to mitigate the vocabulary-mismatch problem for better initial retrieval results. (2) In the second stage, we obtain top-ranked documents as support tweet set and distill the relevant content of the support tweet set by implementing the model-based approach [25].

Technically, we employ the two-stage PRF query expansion under the language model framework. Specially, let $\hat{\theta}_Q$ be the original query model, $\hat{\theta}_{PRF_1}$ be the first-stage feedback model estimated based on the single support tweet. Then, we define our first-stage updated query model $\hat{\theta}_{Q'}$ after the first-stage query expansion as follows:

$$P(w|\hat{\theta}_{Q'}) = (1-\alpha) \cdot P(w|\hat{\theta}_Q) + \alpha \cdot P(w|\hat{\theta}_{PRF_1}) \quad (3)$$

where $\alpha \in [0,1]$ is a parameter to control the amount of first-stage feedback.

To select the single support tweet for the first-stage query expansion, we first obtain ζ top-ranked tweets (we set ζ as 10 in our experiments) retrieved by the original query, then pick the single support tweet from ζ tweets with the following three strategies: (1) the tweet with the earliest timestamp(**ET**). (2) the tweet with the latest timestamp(**LT**). (3) the tweet assigned with the highest score according to its similarity with respect to the query(**HS**).

After selecting the single support tweet T_s, we can estimate $\hat{\theta}_{Q'}$ with the combination of $\hat{\theta}_Q$ and $\hat{\theta}_{PRF_1}$ linearly. Adopting the unigram language model, we intuitively estimate the $\hat{\theta}_Q$ according to the maximum likelihood estimator. Recall that T_s is assumed to generate accurate topical words, and that first-stage feedback aims at improving the initial ranking quality thus to produce reliable support tweet set for second-stage feedback, we estimate the $\hat{\theta}_{PRF_1}$ using the same distribution that $\hat{\theta}_Q$ follows. Thus, we have:

$$P(w|\hat{\theta}_Q) = \frac{c(w,Q)}{\sum_{w' \in V} c(w',Q)} = \frac{c(w,Q)}{|Q|} \quad (4)$$

$$P(w|\hat{\theta}_{PRF_1}) = \frac{c(w,T_s)}{\sum_{w' \in V} c(w',T_s)} = \frac{c(w,T_s)}{|T_s|} \quad (5)$$

where $c(w,Q)$, $c(w,T_s)$ mean the count of term w in Q and T_s, respectively, $|Q|$ and $|T_s|$ is the length of query and the single support tweet.

Then, the modified query $\hat{\theta}_{Q'}$ can be further updated in the second-stage PRF query expansion. Let $F = \{t_1, t_2, ..., t_k\}$ be the pseudo-relevant document set, k is the size of the F which is set as 5 in our experiments, $\hat{\theta}_{PRF_{set}}$ be a feedback model based on F, and $P(\cdot|C)$ be the collection model. Finally, the query model $\hat{\theta}_{Q''}$ after the second-stage PRF query expansion is estimated as follows:

$$P(w|\hat{\theta}_{Q''}) = (1-\beta) \cdot P(w|\hat{\theta}_{Q'}) + \beta \cdot P(w|\hat{\theta}_{PRF_{set}}) \quad (6)$$

where $\beta \in [0,1]$ is a weighting parameter to control the amount of second-stage feedback.

To estimate $\hat{\theta}_{PRF_{set}}$, we implement the model-based feedback approach [25]. The model-based feedback approach helps to purify the document by eliminating some background noise and concentrates on words that are common in the support tweet set. In the approach, the log-likelihood function for the support tweet set is:

$$\begin{aligned} \log P(F|\hat{\theta}_{PRFset}) = \sum_i \sum_w c(w,t_i) \cdot \\ \log((1-\lambda) \cdot P(w|\hat{\theta}_{PRFset}) + \lambda \cdot P(w|C)) \quad (7) \end{aligned}$$

where $c(w,t_i)$ is the count of word w in tweet t_i. Then we implement the EM algorithm [6] with the fixed smoothing

parameter $\lambda = 0.5$ as follows:

$$P^{(n)}(z_w = 1) = \frac{(1-\lambda)P^{(n)}(w|\hat{\theta}_{PRF_{set}})}{(1-\lambda)P^{(n)}(w|\hat{\theta}_{PRF_{set}}) + \lambda p(w|C)} \quad (8)$$

$$P^{(n+1)}(w|\hat{\theta}_{PRF_{set}}) = \frac{\sum_{i=1}^{k} c(w,t_i)P^{(n)}(z_w = 1)}{\sum_u \sum_{i=1}^{k} c(u,t_i)P^{(n)}(z_u = 1)} \quad (9)$$

Overall, we have our query model with two-stage PRF query expansion as follows:

$$P(w|\hat{\theta}_{Q''}) = (1-\beta)\cdot\{(1-\alpha)\cdot P(w|\hat{\theta}_Q) + \alpha \cdot P(w|\hat{\theta}_{PRF_1})\} + \beta \cdot P(w|\hat{\theta}_{PRF_{set}}) \quad (10)$$

3.2 The Estimation of Document Model

The 140-character length limitation in Twitter makes shortened links popular when users update their statuses. URLs within tweets always aim at tracking breaking news stories, recommending interesting video clips and brand marketing [9]. Through clicking a high-quality external link, users can browse a topic-related web page to learn more details about the issues talked in the tweet. Thus, expanding tweets by leveraging external links can enrich the representation of the to-be-retrieved document.

Effective information of the web page that the external link directs to should be extracted to help expand the original tweet. Considering that web information extraction is a challenging task and it is usually domain-dependent, we employ a quick but effective method to extract the topic information from the given URL as follows: (1) we convert the shortened URL to the original long URL through the LongURL web service[4]. (2) we identify the substantive part of domain name in the original URL as the URL keyword (e.g. *cnn* in *www.cnn.com*). (3) we extract text embedded in the <*TITLE*> tag from the raw HTML markup. (4) we split the extracted text by common separator (e.g. '_', '|','-') and discard the substring that contains the URL keyword. For instance, we obtain "BBC News - Egypt blames Gaza group for Alexandria church bombing" from the URL *http://www.bbc.co.uk/news/world-middle-east-12261668* after step (3), then we divide it into two parts: "Egypt blames Gaza group for Alexandria church bombing" and "BBC News", finally we omit the second part which contains the URL keyword "bbc" and take "Egypt blames Gaza group for Alexandria church bombing" as the topic information of the web page that the corresponding URL directs to.

To estimate the document model, we propose two methods under the language model framework by incorporating topic information of the corresponding URL into the original tweet.

3.2.1 Topic Information as Local Context

In the first method, we consider the topic information as the local context of the tweet to estimate a new document model (i.e. LocCtxt document model). Specially, for a tweet T with a URL L, we posit that the topic information I of the corresponding L is supposed to be posted next to T but actually, it is not due to the length limitation in Twitter. Through "translating" L to I, we can include the context of T and refine the representation of T consequently. To summarize, we merge T and I to form a new document D and estimate the document language model using Dirichlet Smoothing [24] with collection language model $P(\cdot|C)$:

$$P(w|\hat{\theta}_D) = \frac{c(w,D) + \mu P(w|C)}{|D| + \mu} \quad (11)$$

where $|D| = |T \bigcup I|$ is the text length of the new document, $c(w,D)$ means the number of term w appearing in the new document D, μ is a smoothing parameter to control the amount of $P(\cdot|C)$.

3.2.2 Topic Information as External Resource

Except for considering the topic information I as the local context of the tweet T, we also utilize the topic information extracted from web pages as external resources. We first collect all URLs in the Tweet11 corpus and then obtain the corresponding topic information to generate a so-called **TopicInfo** corpus. Therefore, the original document model $\hat{\theta}_T$ is smoothed using linear incorporation with the topic information model $\hat{\theta}_I$ estimated based on TopicInfo corpus. The ExRes document model $\hat{\theta}_D$ is estimated as follows:

$$P(w|\hat{\theta}_D) = (1-\eta)\cdot P(w|\hat{\theta}_T) + \eta \cdot P(w|\hat{\theta}_I) \quad (12)$$

where $\eta \in [0,1]$ is the smoothing parameter. Note that only the tweet which has URL(s) can receive an additional smoothing value from the topic information model while the tweet with pure text is assigned with a zero probability in the topic information model.

Also, we estimate the original document model $\hat{\theta}_T$ and the topic information model $\hat{\theta}_I$ under the language model using Dirichlet Smoothing with collection language model $P(\cdot|C_T)$ and $P(\cdot|C_I)$, respectively.

$$P(w|\hat{\theta}_T) = \frac{c(w,T) + \mu_T P(w|C_T)}{|T| + \mu_T} \quad (13)$$

$$P(w|\hat{\theta}_I) = \frac{c(w,I) + \mu_I P(w|C_I)}{|I| + \mu_I} \quad (14)$$

where μ_T and μ_I is the Dirichlet smoothing parameter, $|T|$,$|I|$ is the length of the original tweet and topic information, $c(w,T), c(w,I)$ mean the count of term w in T and I, respectively.

3.3 Temporal Re-Ranking Component

In the microblogosphere, documents to be retrieved should not only be topically related with the given query, but also be posted in the recent past. Therefore, to some extent, the retrieval system is supposed to favor the newer document while penalizing the older one. In this section, we introduce three temporal re-ranking functions to estimate the importance of temporal evidence, and suggest two ways to represent temporal profile.

3.3.1 The Estimation of Temporal Importance

To evaluate the impact of temporal profile, we suggest several temporal re-ranking functions.

Exponential function

Taking advantage of an exponential distribution to combine temporal profile and language model has been shown effective in previous work [12], thus we employ our first approach by letting $P(t|Q,D)$ follow an exponential distribution as follows:

$$P(t|Q,D) = e^{-\frac{\aleph}{k}} \quad (15)$$

[4]http://longurl.org

where $\aleph$ denotes the temporal profile, and k is an exponential rate parameter.

Gaussian function

With the approximation method proposed in [15], the following estimation of $P(t|Q,D)$ is obtained by using Gaussian function:

$$P(t|Q,D) = e^{-\frac{\aleph^2}{2\sigma^2}} \tag{16}$$

where $\aleph$ denotes the temporal profile, and σ is a parameter to control the width of Gaussian function.

Cosine function

Following [15], we define our temporal weighting function using the Cosine function as follows:

$$P(t|Q,D) = \begin{cases} cos(\frac{\pi \cdot \aleph}{2a}) & \aleph < a \\ 0 & \aleph \geq a \end{cases} \tag{17}$$

where $\aleph$ denotes the temporal profile, and a is a parameter to control the width of the Cosine function.

3.3.2 Representation of Temporal Profile

We define temporal profile associated with a document in two different ways: (1) positional information in the recency ranking list and (2) time difference between query timestamp and document timestamp.

Ranking Position as Temporal Profile

Motivated by the positional information used in [15], we re-rank the initial relevance ranking list by document timestamp,from latest to earliest, to obtain a recency ranking list. For each document D, we have the ranking position P_r, and then set temporal profile $\aleph = P_r$. By taking ranking position into consideration, we assign the newer document with a higher probability while penalizing the older one with an appropriate discount.

Time Difference as Temporal Profile

Time difference between the query timestamp and the document timestamp can be used to measure the recentness of document with respect to the given query. We define t_D as the timestamp associated with a document D and let t^* denote the time when the query is issued. Thus we have the time difference as follows:

$$\triangle t_D = (t^* - t_D)/H \tag{18}$$

where H is an interval factor which helps normalize the time difference into a specific interval (we set H as 2 hour in our experiments).

After aggregating documents into intervals according to their time difference, we do an intuitive filtering work for every interval to omit some documents whose relevant score is lower than an interval-specific threshold $\tau_{interval}$ shown in Eq.19

$$\tau_{interval} = \varphi \cdot \frac{1}{N} \sum_{i=1}^{N} Score(Q, D_i) \tag{19}$$

where φ is a scale factor which we set as 0.2 in our experiments, N is the count of documents in the interval, $Score(Q, D_i)$ is the relevance score calculated by KL-divergence retrieval model.

Finally, for every document D, we set temporal profile $\aleph = \triangle t_D$ and then estimate $P(t|Q,D)$ according to Eq.15-17.

Table 1: Summary statistics of Tweet11 corpus

HTML Code	Status	# of Tweet
200	OK	13,839,083
302	Found	1,106,999
403	Forbidden	284,225
404	Not Found	844,494
Null	Null	67,011
Searchable		14,946,082

Table 2: Summary statistics of TopicInfo corpus

HTML Code	Status	# of URLs
200	OK	1,225,947
302	Found	688
403	Forbidden	5,050
404	Not Found	92,378
Other	Unavailable	265,468
Searchable		1,226,635

4. EXPERIMENTS

Several experiments are conducted to evaluate each component of our proposed system for RTST. We first evaluate the effect of the two-stage PRF query expansion. Second, we investigate the impact of the document expansion using external URLs. Third, we measure the performance of our temporal re-ranking component. Finally,we compare the integrated systems against the best performing systems in the TREC'11 RTST.

4.1 Setup

In this section, we describe the experimental dataset and evaluation method which are also adopted in TREC'11 Microblog Track [17]. In addition, baseline systems are set up to estimate the effect of the proposed methods.

4.1.1 Data set

Tweet11 corpus was obtained using a donation of the unique identifiers of a sample of tweets from Twitter [17]. We crawled the HTML version copy of the corpus with the provided tools[5]. The Tweet11 corpus was created by sampling 16 million tweets from January 24, 2011 to February 8, 2011, covering the time period of great events all around world, e.g. Egyptian revolution. Besides, different types of tweets are provided, including replies and retweets in the Tweet11 corpus. Table 1 shows basic statistics of our HTML version acquisition on June 23, 2011. In addition, we collect all the external URLs (i.e. TopicInfo corpus) contained in Tweet11 corpus and extract their topic information for our document expansion process in early December, 2011. Note that web pages might be deleted as time elapsed, we have only crawled a portion of the external URL set. Each piece of topic information in TopicInfo corpus is processed by Porter stemming algorithm and word elimination. Summary statistics of TopicInfo corpus is presented in Table 2.

For every query, we built a dynamic dataset consisting of a set of tweets whose timestamps are smaller than the query timestamp. We discarded the non-English tweets through analyzing the encoding of tweets and removed simple retweets beginning with 'RT'. Then, each tweet was processed by

[5]https://github.com/lintool/twitter-corpus-tools

stemming using the Porter stemmer and word elimination using the INQUERY words stoplist [1].

4.1.2 Evaluation Method

Tweets were judged on the basis of the defined information need using a three-point scale [17]:

- **Not relevant** The content of the tweet does not provide any useful information on the topic, or is either written in a language other than English, or it is a retweet.
- **Minimally Relevant** The tweet provides some information on the topic, but it is not sufficiently informative.
- **Highly Relevant** A highly relevant tweet will either contain highly informative content, or link to highly informative content.

The top-ranked documents which are strictly ranked by timestamp for all runs were evaluated in terms of their precision at rank 30 (P@30) and mean average precision (MAP).

We carried out all experiments based on two standard topic sets used in the TREC'11 RTST. The first topic set which we call **allrel** considers both minimally relevant and highly relevant tweets as relevant over 49 topics. The second topic set that considers only highly relevant tweets over 33 topics is called **highrel**.

4.1.3 Baseline System

In section 4.2, we discuss the performance of our proposed system. To contextualize our methods, we used the Lemur toolkit[6] (version 4.12) to implement two baseline systems for comparison.

The baseline retrieval model labeled as "KLNoFB" is the KL-divergence retrieval model [24] as follows:

$$Score(Q, D) = \sum_{w \in V} P(w|\hat{\theta}_Q) \cdot \log P(w|\hat{\theta}_D) \qquad (20)$$

where both $\hat{\theta}_Q$ and $\hat{\theta}_D$ are estimated using empirical word distribution, and we choose the Dirichlet smoothing method for document model estimation with the smoothing parameter μ set as 100. This baseline is set up for evaluation of the performance of the proposed query model and document model.

To measure the usefulness of temporal re-ranking component, we set up KL2SFBExR as another baseline based on KL-divergence retrieval model where we estimate query model using the proposed two-stage PRF query expansion (we set α=0.4, β=0.6), and utilize topic information as external resource to generate the document model (we set η=0.6).

Note that both in baseline systems and the proposed system, we obtain the top-ranked 30 tweets from the relevant ranking list achieved by the Real-Time Tweet Ranking Model described in Section 3, and then re-rank them according to their timestamps as an experimental run.

4.2 Experimental Results

In this section, we report the results of experiments carried out to evaluate the effectiveness of each component compared with the baseline systems, and the performance of the integrated systems compared with the best performing systems in the TREC'11 RTST. In addition, to explain the numerous acronyms for the various methods, we summarize the Table 8 which contains a brief description for each acronym.

4.2.1 The Evaluation of Query Model

We conducted several experiments to measure the effect of our two-stage PRF query expansion for RTST. These experiments aim at evaluating (1) how different support tweet selection strategies in the first-stage feedback affect the performance of the proposed query expansion; and (2) how each stage feedback boosts the effectiveness of retrieval results.

We first examined the three strategies we adopted to obtain the single support tweet. In this group of experiments, we only employed the first-stage feedback by setting the parameter β in Eq.10 as 0 to update the original query model, and we estimated the document model in a way as KLNoFB does. With the three selection methods described in Section 3.1, we labeled three variations of first-stage feedback approach as "KLFSFBET", "KLFSFBLT" and "KLFSFBHS", respectively. We compared the results of KLFSFBET, KLFSFBLT and KLFSFBHS in terms of P@30 in Table 3. From the table, it suggests that support tweet in the first-stage feedback can enhance the retrieval performance with varied improvements. Specifically, KLFSFBET hardly improves the search effectiveness while KLFSFBLT achieves a much better search performance, indicating that the latest tweet from the candidate feedback set is more effective than the earliest one to generate the feedback words; however, selecting the tweet according to the highest relevance score as the single support tweet can improve the system performance significantly, showing that semantic aspects of document outperform the temporal aspects in the first-stage feedback modeling.

We then evaluated the retrieval precision of the proposed PRF query expansion step by step. The results are summarized in Table 4. Note that all the methods listed in the table adopt the same document model as KLNoFB does. As we can see, by only employing the second stage of feedback, the method labeled as "KLSSFB" can improve both P@30 and MAP on two topic sets, though not significantly. However, the relative improvements of KLFSFBHS over KLNoFB on allrel and highrel are 12.92% and 18.52% in terms of P@30 respectively; while the corresponding MAP improvements of KLFSFBHS over KLNoFB on allrel and highrel are 27.18% and 50.21%, respectively, which may suggest that the first-stage feedback is more effective than the second-stage feedback. Moreover, the "KL2SFB" method that combines two stages of feedback can further enhance the effectiveness of retrieval results.

4.2.2 The Evaluation of Document Model

We investigated two methods to estimate the document model by incorporating topic information with the original tweet text. We experimentally evaluated the effectiveness of the proposed estimation of the document model. Let KLLocCtxt and KLExRes be the methods regarding topic information as local context and external resource, respectively. To compare with KLNoFB, we estimated the query model without further expanding the original query. Furthermore, we set the Dirichlet smoothing parameter μ as 100 in the KLLocCtxt while in the KLExRes, the Dirichlet

[6]http://www.lemurproject.org/lemur.php

Table 3: P@30 comparasion of three support tweet selection strategies using first-stage PRF query expansion

Topic Set	KLNoFB	KLFSFBET	KLFSFBLT	KLFSFBHS
allrel	0.3422	0.3442(+0.58%)	0.3748(+9.53%)	**0.3864(+12.92%)**
highrel	0.1253	0.1253(0%)	0.1445(+15.32%)	**0.1485(+18.52%)**

Table 4: The performance comparison of two-stage PRF query expansion, not using document expansion

Topic Set	Metric	KLNoFB	KLFSFBHS	Improv.	KLSSFB	Improv.	KL2SFB	Improv.
allrel	P@30	0.3422	0.3864	+12.92%	0.3653	+6.75%	**0.4082**	+19.29%
	MAP	0.1810	0.2302	+27.18%	0.1969	+8.78%	**0.2401**	+32.65%
highrel	P@30	0.1253	0.1485	+18.52%	0.1283	+2.39%	**0.1515**	+20.91%
	MAP	0.1657	0.2489	+50.21%	0.1749	+5.55%	**0.2514**	+51.72%

Table 5: The performance comparison of LocCtxt document model and ExRes document model, not using query expansion

Topic Set	Metric	KLNoFB	KLExRes	Improv.	KLLocCtxt	Improv.
allrel	P@30	0.3422	0.3422	-	**0.3973**	+16.10%
	MAP	0.1810	0.1830	+1.10%	**0.2139**	+18.18%
highrel	P@30	0.1253	**0.1626**	+29.77%	0.1596	+27.37%
	MAP	0.1657	0.1957	+18.11%	**0.2068**	+24.80%

Table 6: The performance comparison of three temporal re-ranking functions, and two temporal profile representations based on KL2SFBExR

Topic Set	Metric	KL2SFB ExR	KL2SFB ExRPE	KL2SFB ExRPG	KL2SFB ExRPC	KL2SFB ExTDE	KL2SFB ExTDG	KL2SFB ExTDC
allrel	P@30	0.4116	0.4177	**0.4177**	0.4170	0.4177	0.4156	0.4156
	MAP	**0.2380**	0.2378	0.2365	0.2359	0.2377	0.2374	0.2374
highrel	P@30	0.1889	0.1950	**0.1979**	0.1970	0.1960	0.1960	0.1960
	MAP	0.2680	0.2713	**0.2722**	0.2714	0.2665	0.2713	0.2713

smoothing parameters μ_T, μ_I are set as 100, 50 for original tweet language model and topic information language model, respectively.

Table 5 shows the performance of two document modeling approaches. Comparing the performance of proposed methods on two topic sets, we can conclude that KLLocCtxt outperforms KLNoFB significantly and consistently on both allrel and highrel; however, it is interesting to observe that KLExRes behaves quite differently on two topic sets: it improves P@30 and MAP on highrel strikingly compared with the KLNoFB baseline while it only slightly increase the MAP score over the baseline KLNoFB by 1.10% when running on allrel topic set. Besides, when comparing the performance of two methods running on a particular topic set in terms of P@30, we can figure out that KLLocCtxt outperforms KLExRes on allrel while KLExRes obtains better retrieval results than the corresponding one achieved by KLLocCtxt on highrel.

4.2.3 Temporal Re-Ranking Component

We suggest three temporal re-ranking functions to estimate the importance of the temporal aspects of documents and investigate two representations of temporal profile. To evaluate the usefulness of temporal re-ranking component, we combined three functions with two different types of temporal profile based on the KL2SFBExR baseline to form six methods: KL2SFBExRPE, KL2SFBExRPG, KL2SFBExRPC, KL2SFBExTDE, KL2SFBExTDG and KL2SFBExTDC. The retrieval results are summarized in Table 6, where we highlight the best result for each row. From Table 6, we observe that, compared with the KL2SFBExR baseline, all six temporal re-ranking functions achieve better search performances in terms of P@30. For a particular representation of temporal profile (e.g. ranking position), different temporal re-ranking functions can achieve the similar improvements when we tune their parameters to make the value of function at argument r_{max} (i.e. the max ranking position) is around 0.6. Besides, for a particular temporal re-ranking function, two different representations work similarly, which may indicate that any rational representation of temporal profile can obtain a reliable retrieval performance.

4.2.4 Integration of Query Model, Document Model and Temporal Re-Ranking Component

Finally, we measured the performance of the proposed system that integrates the query expansion component, document expansion component and temporal re-ranking component. For the query expansion component, we adopt two-stage PRF query expansion with HS selection strategy. For the document expansion component, we employ both LocCtxt document model and ExRes document model based on the observation that the two document models behave differently on different topic sets. For the temporal re-ranking component, we employ Gaussian function with ranking position as temporal profile. Overall, we have two runs labeled as KL2SFBLocRPG and KL2SFBExRPG .

Table 7 demonstrates the performance of our systems. In addition, we also report the best three performing results of the TREC'11 Real-Time Search Task [17] for comparison purpose. This result suggests that both of our systems outperform the best performance of TREC'11 RTST on the highrel topic set significantly, specifically, KL2SFBExRPG achieves 26.37% and 9.94% further increases in P@30 and MAP, respectively. On the other hand, the performances of

Table 7: The performances of systems integrating query expansion (we set $\alpha = 0.4$, $\beta = 0.6$), document expansion (we set η=0.6 when using ExRes document model) and temporal re-ranking components (we set σ=120 in Gaussian function). isi, FUB, CLARITY_DCU are the best three performances in TREC'11 Real-Time Search Task on highrel and allrel. The best performances are shown in bold. †, ‡, ¶ mean the corresponding improvements over isi, FUB and CLARITY_DCU are significant respectively.

Topic Set	Metric	isi	FUB	CLARITY_DCU	KL2SFBExRPG	KL2SFBLocRPG
allrel	P@30	**0.4551**	0.4401	0.4211	0.4177	0.4490
	MAP	0.1923	0.2348	0.2139	0.2365†¶	**0.2552**†‡¶
highrel	P@30	0.1566	0.1495	0.1434	**0.1979**†‡¶	0.1727†‡¶
	MAP	0.2476	0.2286	0.2065	**0.2722**†‡¶	0.2606‡¶

Table 8: Summary of mentioned acronyms

Acronyms	Description
KLNoFB	KL-divergence retrieval model without any expansion technique
KLFSFBET	KL-divergence retrieval model, only using first stage query expansion(support tweet with Earliest Timestamp)
KLFSFBLT	KL-divergence retrieval model, only using first stage query expansion(support tweet with Latest Timestamp)
KLFSFBHS	KL-divergence retrieval model, only using first stage query expansion(support tweet with Highest Score)
KLSSFB	KL-divergence retrieval model, only using second stage query expansion
KL2SFB	KL-divergence retrieval model, using two-stage query expansion(support tweet with Highest Score for first stage)
KLExRes	KL-divergence retrieval model, only using document expansion(topic infomation as external resource)
KLLocCtxt	KL-divergence retrieval model, only using document expansion(topic infomation as local context)
KL2SFBExR	KL-divergence retrieval model, using two-stage query expansion(support tweet with Highest Score for first stage) and document expansion(topic infomation as external resource)
KL2SFBLoc	KL-divergence retrieval model, using two-stage query extension(support tweet with Highest Score for first stage) and document expansion(topic infomation as local context)
KL2SFBExRP*	Using Temporal re-ranking (ranking position as temporal profile, * can be G(aussian),E(xponential),C(osine)) based on KL2SFBExR
KL2SFBExTD*	Using Temporal re-ranking (time difference as temporal profile, * can be G(aussian),E(xponential),C(osine)) based on KL2SFBExR
KLNoFBRPG, KL2SFBRPG, KL2SFBLocRPG	Using Temporal re-ranking (ranking position as temporal profile in gaussian function) based on KLNoFB,KL2SFB, KL2SFBLoc , respectively

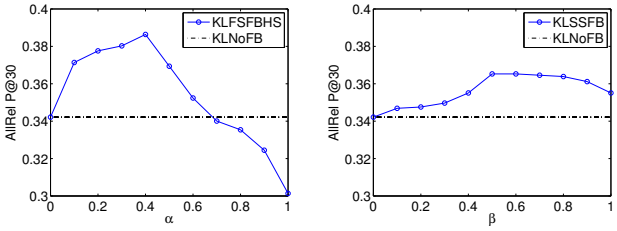

Figure 1: Sensitivity to the first-stage feedback interpolation coefficient α (left) and second-stage feedback interpolation coefficient β (right) of PRF query expansion.

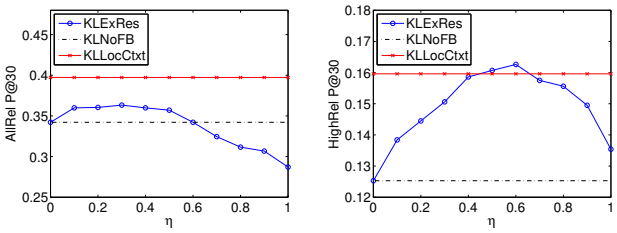

Figure 2: Sensitivity to the ExRes document model interpolation coefficient η on allrel (left) and highrel (right).

our systems on allrel are very competitive compared with the top 3 performances of TREC'11 RTST, though not better than the best one. Besides, it is not surprisingly to observe that KL2SFBLocRPG beats KL2SFBExRPG on the allrel topic set while losing to KL2SFBExRPG on the highrel topic set, because we have seen the different behaviors between ExRes document model and LocCtxt document model in section 4.2.2.

4.3 Discussion

Many parameters in our proposed approach can affect the behavior of the system performance. In this section, we analyze the robustness to the parameter setting in query expansion component, document expansion component and temporal re-ranking component.

In the two-stage PRF query expansion, there are parameters α and β to control the amount of first-stage feedback and second-stage feedback. Specifically, when $\alpha = 0$ (i.e. we ignore the effect of first-stage feedback), and let β range from 0 to 1.0, we can see how second-stage feedback affect the retrieval results; when $\beta = 0$, and let α vary from 0 to

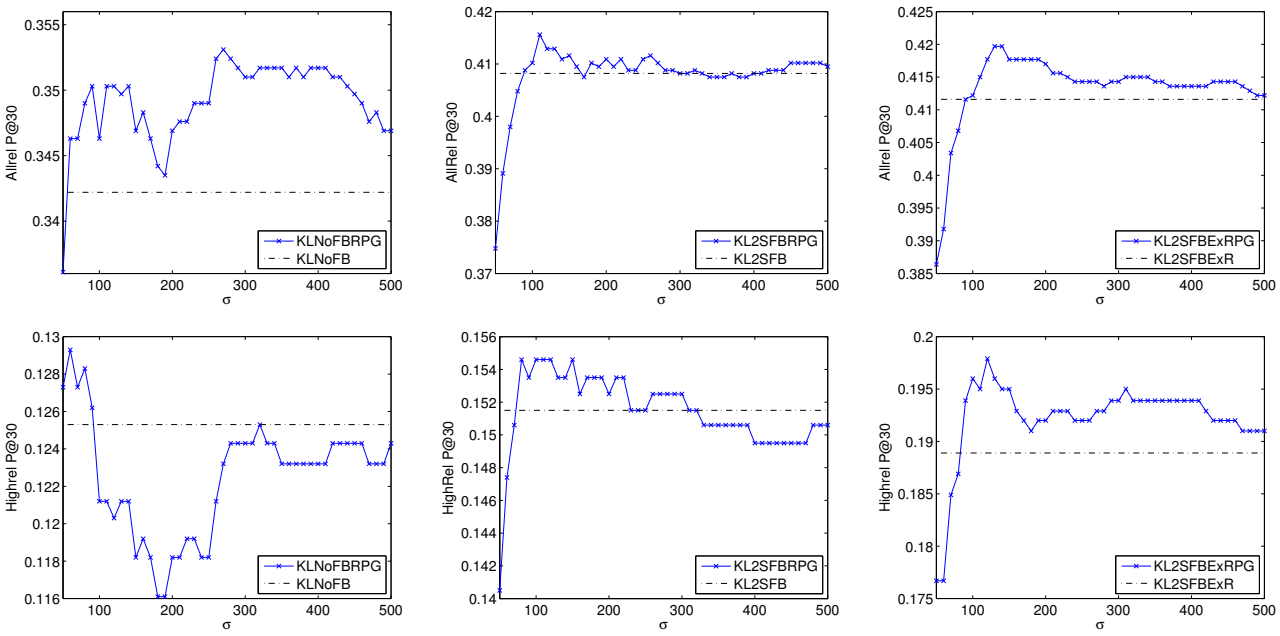

Figure 3: Sensitivity to the Gaussian function parametor σ over KLNoFB (left), KL2SFB (middle) and KL2SFBExR (right) on alrel(top) and highrel (down).

1.0, we can see the behavior of the first-stage feedback. The effects of first-stage feedback and second-stage feedback are shown in Figure 1. The experimental results indicate that when α is around 0.4 and β is around 0.6, each stage of query expansion can achieve their own best performances, which outperform the KLNoFB baseline. Besides, first-stage feedback can receive a better performance than second-stage does when they are assigned with their optimal parameters (i.e. $\alpha = 0.4, \beta = 0.6$). However, second-stage feedback seems to be more robust.

Recall that we build KLExRes document model by interpolating the topic information model with the original document model. The interpolation is controlled by a coefficient η. When $\eta = 0$, we only use the original document model (i.e. no topic information), while if $\eta = 1$, we completely ignore the original document model and only take advantage of topic information model. We demonstrate in Figure 2 how the score of P@30 changes according to the value of η along with the performance of KLNoFB and KLLocCtxt. From Figure 2, it is worthy to point out that KLExRes works differently on the two topic sets: when running on allrel topic set, topic information seems not to be effective, resulting in that KLExRes loses to KLLocCtxt consistently and only defeats KLNoFB when $\eta < 0.6$; however, when running on highrel topic set, the pure topic information model ($\eta = 1$) result is much better than the original document model ($\eta = 0$), which may indicate that topic information is very effective for retrieving highly relevant document in the microblogosphere, as a result, KLExRes consistently outperforms KLNoFB and can achieve a better result than KLLocCtxt when η is around 0.6. A rational reason for this interesting phenomenon may be that a highly relevant tweet tends to include a high-quality URL, thus topic information within the URL can boost the effectiveness of retrieval results.

According to the results shown in section 4.2, both query expansion component and document expansion component can boost the retrieval effectiveness, so we employ the temporal re-ranking component over KLNoFB, KL2SFB and KL2SFBExR, to further compare the robustness of temporal re-ranking component with respect to the different initial retrieval results. Note that all the temporal re-ranking functions and both representations of temporal profile perform similarly, we only take Gaussian function and ranking postion representation for example. From Figure 3 we can see that temporal re-ranking component improves the KL2SFBExR's performance in most cases when σ ranges from 50 to 500; however, it seems to be unstable when running over worse initial retrieval results (i.e. KLNoFB and KL2SFB) with parameter σ at the same range. Thus, based on the observation, we can conclude that temporal re-ranking component performs much better and more robust when running over better initial retrieval results. In addition, when tuning σ around 120, we obtain the best retrieval performance on allrel and highrel.

5. CONCLUSION AND FUTURE WORK

In this study, we propose an effective approach to improve the performance of our system to mine valuable information in the microblogosphere. Our system consists of three important components: query modeling, document modeling and temporal re-ranking. To estimate the query model, we introduce a two-stage pseudo-relevance feedback query expansion based on language model framework. For the document modeling, we present LocCtxt document model and ExRes document model which refine the representation of document with the topic information in different ways. For the temporal re-ranking component, we suggest several functions as well as different representations of temporal profile.

Experiments on Tweet11 corpus indicate the usefulness

of each component, as we obtain significant improvements over the baseline systems. Besides, the proposed system for RTST which integrates all the components increases the best performing system of TREC'11 RTST on the highrel topic set by 26.37% and 9.94% in P@30 and MAP, respectively.

Many studies remain for the future work. One of the most interesting directions is to explore the users' connection in social network for a further analysis on user's information need in the microblogosphere. Moreover, another interesting direction is to classify the type of URLs for more accurate document expansion (e.g. detect the news articles and extract the main content as topic information). Furthermore, we are also interested in how to incorporate the temporal profile into the language model to obtain better retrieval results.

6. ACKNOWLEDGMENTS

The work reported in this paper was supported by the National Natural science Foundation of China Grant 60875033.

7. REFERENCES

[1] J. Allan, M. E. Connell, W. B. Croft, F. Feng, D. Fisher, and X. Li. Inquery and trec-9. In *TREC*, 2000.

[2] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART: Trec 3. In D. K. Harman, editor, *Overview of the 3th Text REtrieval Conference* Trec-*3*, pages 69–80, Gaithersburg, 1995. NIST.

[3] C. C. Chen, Y.-T. Chen, Y. S. Sun, and M. C. Chen. Life cycle modeling of news events using aging theory. In N. Lavrac, D. Gamberger, L. Todorovski, and H. Blockeel, editors, *ECML*, volume 2837 of *Lecture Notes in Computer Science*, pages 47–59. Springer, 2003.

[4] K.-Y. Chen, L. Luesukprasert, and S. cho Timothy Chou. Hot topic extraction based on timeline analysis and multidimensional sentence modeling. *IEEE Trans. Knowl. Data Eng.*, 19(8):1016–1025, 2007.

[5] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time sensitive queries. In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, editors, *CIKM*, pages 1437–1438. ACM, 2008.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39, 1977.

[7] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *WWW*, pages 331–340. ACM, 2010.

[8] R. T. Fernández, D. E. Losada, and L. Azzopardi. Extending the language modeling framework for sentence retrieval to include local context. *Inf Retr.*, 14(4):355–389, 2011.

[9] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Micro-blogging as online word of mouth branding. In D. R. O. Jr., R. B. Arthur, K. Hinckley, M. R. Morris, S. E. Hudson, and S. Greenberg, editors, *CHI Extended Abstracts*, pages 3859–3864. ACM, 2009.

[10] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[11] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of SIGIR*, pages 120–127, 2001.

[12] X. Li and W. B. Croft. Time-based language models. In *CIKM*, pages 469–475. ACM, 2003.

[13] D. E. Losada and R. T. Fernández. Highly frequent terms and sentence retrieval. In N. Ziviani and R. A. Baeza-Yates, editors, *SPIRE*, volume 4726 of *Lecture Notes in Computer Science*, pages 217–228. Springer, 2007.

[14] Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, and J. J. Lin, editors, *CIKM*, pages 1895–1898. ACM, 2009.

[15] Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In F. Crestani, S. Marchand-Maillet, H.-H. Chen, E. N. Efthimiadis, and J. Savoy, editors, *SIGIR*, pages 579–586. ACM, 2010.

[16] V. Murdock. Aspects of sentence retrieval. *SIGIR Forum*, 41(2):127, 2007.

[17] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *Proceedings of TREC 2011*, 2012.

[18] J. Pontin. From many tweets, one loud voice on the Internet. *New York Times Online [web site]. Retrieved May*, 8:2006, 2007.

[19] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[20] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC'94*, pages 109–126, 1994.

[21] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: experiments in automatic document processing*, pages 313–323. Prentice Hall, 1971.

[22] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[23] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Jarvelin, editors, *SIGIR*, pages 162–169. ACM, 2006.

[24] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[25] C. Zhai and J. D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM*, pages 403–410. ACM, 2001.