Improving Microblog Retrieval with Feedback Entity Model

Feifan Fan Runwei Qiang * Chao Lv Jianwu Yang †
{fanff, qiangrw, lvchao, yangjw}@pku.edu.cn
Institute of Computer Science and Technology
Peking University, Beijing 100871,China

ABSTRACT

When searching over the microblogging, users prefer using queries including terms that represent some specific entities. Meanwhile, tweets, though limited within 140 characters, are often generated with one or more entities. Entities, as an important part of tweets, usually convey rich information for modeling relevance from new perspectives. In this paper, we propose a feedback entity model and integrate it into an adaptive language modeling framework in order to improve the retrieval performance. The feedback entity model is estimated with the latest entity-associated tweets based upon a regularized maximum likelihood criterion. More specifically, we assume that the entity-associated tweets are generated by a mixture model, which consists of the entity model, the domain-specific language model and the collection language model. Experimental results on two public Text Retrieval Conference (TREC) Twitter corpora demonstrate the significant superiority of our approach over the state-of-the-art baselines.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

Algorithms, Experimentation, Performance

Keywords

Microblog Retrieval; Language Modeling; Feedback Entity Model; Freebase

1. INTRODUCTION

With the rapid growth of microblogging, an increasing number of studies have paid much attention to the research on Information Retrieval (IR) in the context of microblogosphere, especially Twitter[1]. To explore the information seeking behavior in microblogosphere, TREC first introduced a temporally-anchored ad hoc search task in 2011 [18] to attract more studies on how to retrieve relevant tweets according to users' queries. In particular, the goal of the temporally-anchored ad hoc search task can be summarized as "*At time T, give me the most relevant tweets about topic Q*".

When searching over the microblogging, users tend to use entities in their queries to express the certain information need. Among the topics (i.e. queries) released by TREC[2] Microblog track [18, 23, 13], more than half of the topics include at least one named entities. Nevertheless, different people are likely to use alternative aliases to represent the same entity. Hence, the vocabulary gap between tweet-posters and the users who search tweets lies as a big challenge for effective IR over tweets. Moreover, as tweets are under the length limitation of 140 characters, the risk of mismatch between query terms and any word observed in relevant tweets is larger than that for traditional Web document retrieval.

Fortunately, we have observed that, while being not allowed to write long text in one tweet, users tend to leverage entities, usually regarding to specific real-world persons, organizations, and locations, to illustrate the topic of their tweets (See Figure 1 for example tweets about "Mila Kunis"). An analysis of a public Twitter corpus published by TREC has illustrated that more than 20% tweets contain at least one entity [18, 13].

Figure 1: Example tweets containing the entities.

Beyond the pure text, entities, as an important part of tweets, usually convey rich information for modeling relevance from new perspectives. Henceforth, to improve users' experience of surfing in microblogosphere, it becomes more necessary to explore how

*First two authors contribute to this work equally.
†Corresponding author.

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806461.

[1]http://www.twitter.com
[2]http://trec.nist.gov

to take advantage of entity information from tweets to facilitate ad hoc search task in the microblogging context.

In this paper, we propose a novel feedback entity model and integrate it into an adaptive language modeling framework. Our new framework takes advantage of the rich entity information in Twitter to address the challenges in microblog retrieval. We first estimate the entity model using the entity-associated feedback tweets based upon a regularized maximum likelihood criterion. In particular, we take advantage of the domain information in an open-domain ontology (i.e. Freebase[3]) to filter domain-specific background noise in the estimation of the entity model. Then, by leveraging a language modeling based IR framework, it is quite natural to integrate the entity model into the whole IR framework via query expansion. Entity model can help update the estimation of the query language model based on the extra evidence carried by the entity-associated feedback tweets. Furthermore, with using our new method, different entity aliases referring to the same entity are likely to lead to similar entity models as they share a similar word context. This feature can also mitigate the vocabulary gap between different users to some extent.

The main contributions of this paper include: (1) we propose a feedback entity model and integrate it into the adaptive language modeling framework in order for a more effective IR in microblogging; (2) we use a generative model to estimate the entity model with entity-associated feedback tweets; (3) we perform a set of experiments on two public twitter test collections published by TREC to compare our proposed method with the state-of-the-art baseline systems. And, the experimental results demonstrate that our proposed approach can give rise to significantly better retrieval performance.

The rest of the paper is organized as follows. Section 2 gives an overview of the related work. Then, we describe our adaptive language modeling framework in Section 3. In Section 4, we present our proposed feedback entity model in details. The experimental results as well as the comparisons with the-state-of-arts are shown in Section 5. Finally, we conclude the paper and outline our future work in Section 6.

2. RELATED WORK

2.1 PRF-based Query Expansion

Query expansion (QE) based on pseudo-relevance feedback (PRF) [12, 14, 15, 27, 8] is widely used in microblog search to improve the retrieval performance. However, traditional PRF-based query expansion may not work well as it relies on the assumption that most of the frequent terms in the pseudo-relevance documents are useful [16]. Cao *et al.* [2] re-examined this assumption and showed that it does not always hold in reality – many expansion terms identified in traditional approaches are indeed unrelated to the query and harmful for the retrieval performance. They then integrated a term classification process to predict the effectiveness of expanded terms. Liang *et al.* [12] proposed a Real Time Ranking Model (RTRM), which utilized a two-stage pseudo-relevance feedback query expansion to estimate the query language model and expand documents with shortened URLs in microblog. In addition, RTRM can evaluate the temporal aspects of documents with the temporal re-ranking components. Miyanishi *et al.* [17] proposed a first-stage manual tweet selection feedback to improve the retrieval performance. They further used a two-stage PRF based on similarity of temporal profiles of the query and top retrieved tweets. However, this method sometimes fails due to the redundancy of selected tweets, which usually contain a significant number of meaningless words that may degrade search results. Just like Liang and Miyanishi, we attempt to improve the effectiveness of PRF based QE by utilizing a first stage query expansion.

Other works attempted to acquire knowledge from external sources to obtain additional query terms [5]. Li *et al.* [11] explored the possibilities of using Wikipedia's articles as an external corpus to expand ad-hoc queries and demonstrated that Wikipedia especially useful for the case of weak queries that PRF fails to improve. In their methods, expansion terms were extracted from the top ranked Wikipedia articles. Pan *et al.* [19] proposed using Dempster-Shafer's Evidence Theory to measure the certainty of expansion terms from the Freebase structure. Dalton *et al.* [3] proposed a new technique, called entity query feature expansion (EQFE) which enriches the query with features from entities and their links to knowledge bases, including structured attributes and text. In our study, we seek help from Freebase to guide estimating the domain background language model in feedback entity model estimation.

2.2 Entity-based Query Expansion

Entities have been well exploited in the search scenario, especially in short text retrieval, where entities represent important semantics. Sokes *et al.* [24] employed ontology-based (MeSH and Entrez Gene) query expansion and entity-based relevance feedback for genomics search. Their approach works by identifying potentially relevant entity instances in an initial set of retrieved candidate paragraphs. These entities are then directly added to the initial query with the aim of boasting the rank of passages containing lists of these entities.

Entities, not only real-world ones, but also user-generated ones (e.g. hashtags, user mentions), play a more important role in Twitter than in traditional documents due to the length limitation of tweets. Efron *et al.* [6] described the problem of "hashtag retrieval", a type of entity search. They leveraged the retrieved query-related hashtags in query expansion and achieved better retrieval performance. They assumed that hashtags often reflect some important aspects of a tweet such as its topic or its intended audience. However, some hashtags are very informal, so using hashtags as direct feedback words may lead to risky situations such as topic drift. Further analysis of the entity might gain some retrieval performance. To the best of our knowledge, entity models which model the entities with entity-associated feedback tweets, have not been adapted or leveraged in temporally-anchored search task yet, which might help a lot in retrieving relevant tweets.

2.3 Entity Recognition

To estimate and utilize the entity topic model, one important issue is to recognize entities in tweets and query. Many works have been attempted to extract entities in the microblogosphere. Among them, Ritter *et al.* [21] proposed a distantly supervised entity recognition approach. They first constructed large amounts of unlabeled data collection including large dictionaries of entities gathered from Freebase and information about an entity's context across its mentions, and then applied LabeledLDA [20] to leverage the unlabeled data. Their method has been reported with a good performance in named entity segmentation for tweets. In this study, we apply Ritter's method to recognize entities in tweets

The detection of the named entity in queries is even harder as queries tend to be very short. Guo *et al.* [7] first addressed the problem of Named Entity Recognition in Query (NERQ) and proposed taking a probabilistic approach to the task using query log data and Latent Dirichlet Allocation. It was found that keyword queries

[3]http://www.freebase.com

that people issue to retrieve information from Twitter are, on average, significantly shorter than queries submitted to traditional Web search engines (1.64 words vs. 3.08 words) [25]. Henceforth, in this paper, we adopt a two-stage entity match method to recognize entities in queries. This method works quite well for the short Twitter queries, most of which are entities themselves.

3. ADAPTIVE LANGUAGE MODELING FRAMEWORK

To address the ad hoc search task in the microblogosphere, we employ language modeling approach. In particular, we assume that a query Q is generated by the query language model $\hat{\theta}_Q$, while a document D is generated by the document language model $\hat{\theta}_D$. After estimating $\hat{\theta}_Q$ and $\hat{\theta}_D$ according to [9], the relevance score of D with respect to Q can be computed by the following negative KL-divergence function:

$$S(Q,D) = -KL(\hat{\theta}_Q||\hat{\theta}_D) \propto \sum_{w\in V} p(w|\hat{\theta}_Q)\cdot \log p(w|\hat{\theta}_D) \quad (1)$$

where V is the set of words in our vocabulary. To obtain an effective ranking function for IR in microblogosphere, it is critical to accurately estimate $\hat{\theta}_Q$ and $\hat{\theta}_D$, respectively.

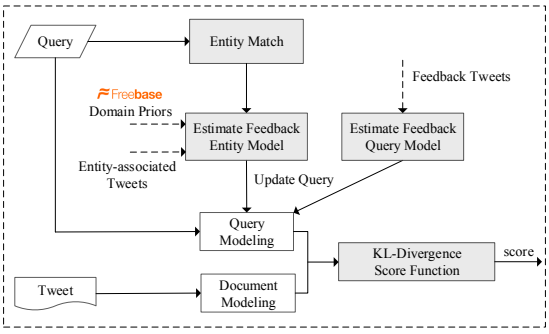

Figure 2: Overall architecture of the adaptive language modeling framework.

Figure 2 shows the overall architecture of our adaptive language modeling framework, which is based on the KL-divergence model and adopts a two-stage query expansion method.

When a new query comes, we first extract all the entities in the query. In our proposed system, we adopt a novel two-stage entity match method to recognize the named entities in the query:

- **Stage 1**: Tagging the entities in the query with an open source named entity recognition tool [4], which has been reported effective in twitter [21, 22].
- **Stage 2**: Taking another round of entity matching by adopting a maximum matching algorithm with a pre-generated entity dictionary. The entity dictionary is generated by combining all the entities extracted from the twitter corpus with the open source tool.

With the extracted entities, we estimate the feedback entity model with a generative model and update the original query model (i.e. first stage QE). Then, a traditional feedback query model is applied to further update the query model (i.e. second stage QE).

[4]https://github.com/aritter/twitter_nlp

3.1 Feedback Entity Model Based QE

To model the dynamics of microblogosphere, we propose to generate feedback entity model $\hat{\theta}_E$ for extracted entity in the query. Feedback entity model is a probabilistic distribution of words $\{p(w|\hat{\theta}_E)\}_{w\in V}$ and represents the common topics of a given entity. Section 4 will elaborated on the estimation of entity model. With the entity model $\hat{\theta}_E$, we are able to expand the original query language model, i.e. $\hat{\theta}_Q \rightarrow \hat{\theta}_{Q'}$ as query expansion. We take advantage of the linear interpolation for combining the original language model and the feedback entity model:

$$p(w|\hat{\theta}_{Q'}) = (1-\alpha)\cdot p(w|\hat{\theta}_Q) + \alpha \cdot p(w|\hat{\theta}_E) \quad (2)$$

where $\alpha \in [0,1]$ controls the influence of the entity model.

For the traditional feedback methods, the balance parameter α is usually set to a fixed value across all the queries. However, considering that entities in different queries may not have the same importance, an adaptive weighting parameter should be utilized to dynamically balance the original query and feedback entity information. In our method, our balance parameter takes the form of:

$$\alpha' = \alpha \times \frac{\sum_{w\in E} IDF(w)}{\sum_{w'\in Q} IDF(w')} \quad (3)$$

where E means the entities in the query Q, $\alpha \in [0,1]$ is a fixed parameter and the IDF of terms in the entities can affect the final balance coefficient.

Note that, when there are more than one entities recognized in the query, we employ the weighted average of all entity models as the unified entity model to expand the corresponding query language model:

$$p(w|\hat{\theta}_E) = \frac{\sum_{i=1}^{n} IDF(E_i)\times p(w|\hat{\theta}_{E_i})}{\sum_{i=1}^{n} IDF(E_i)} \quad (4)$$

where n is the number of entities in the given query. $\hat{\theta}_{E_i}$ is the entity model for entity E_i, and $IDF(E_i)$ is the inverse document frequency for entity E_i. We suppose that an entity with a higher inverse document frequency is more important than that with a lower inverse document frequency. Taking the query "Hu Jintao visit to the United States" as an example, the entity "Hu Jintao" ($IDF = 5.7$) will have a higher weight compared with the entity "United States" ($IDF = 3.8$).

3.2 Feedback Query Model Based QE

When the query is expanded with the feedback entity model, our approach can retrieve more relevant documents at the top, which contain more accurate word distribution than by initial search result. Based on this hypothesis, we further utilize a model-based feedback to update the query representation. Note that the feedback tweets for model estimation are top ranked results using the language modeling framework with first stage QE. More specifically, we update the $\hat{\theta}_{Q'}$ using the feedback query model $\hat{\theta}_F$ which is widely used in the microblogging retrieval [27, 12].

$$p(w|\hat{\theta}_{Q''}) = (1-\beta)\cdot p(w|\hat{\theta}_{Q'}) + \beta \cdot p(w|\hat{\theta}_F) \quad (5)$$

where $\beta \in [0,1]$ is a parameter to control the weight of the model-based feedback.

The model-based feedback model generates a feedback document by mixing the feedback query model $\hat{\theta}_F$ with the collection language model $\hat{\theta}_C$. Under this simple mixture model (SMM), the

log-likelihood of feedback documents F is:

$$\log p(F|\hat{\theta}_F) = \sum_w c(w, F) \cdot \log((1-\lambda) \cdot p(w|\hat{\theta}_F) + \lambda \cdot p(w|\hat{\theta}_C)) \quad (6)$$

where $c(w, F)$ denotes the count of word w occurred in the set of feedback documents F. Then, we follow the work of [12, 4] and implement the EM algorithm with the best tuned parameter λ.

4. FEEDBACK ENTITY MODEL

In this section, we describe the details of estimating the feedback entity model. We first propose a general method to estimate the feedback entity model with entity-associated feedback tweets (i.e. EF). Then, we discuss how to incorporate expert knowledge from Freebase into our model. Finally, we elaborate on two methods of fetching EF.

4.1 A Generative Entity Model of Feedback Tweets

A natural way to estimate a feedback entity model $\hat{\theta}_E$ is to assume that the EF is generated by a probabilistic model $p(EF \mid \theta)$. It is straightforward to leverage the unigram language model, which generates each word w in EF independently according to θ.

$$p(EF|\theta) = \prod_i \prod_w p(w|\theta)^{c(w,D_i)} \quad (7)$$

where $c(w, D_i)$ is the count of word w occurred in document D_i. This simple model would be reasonable if the feedback tweets only contain information relevant to the corresponding entity. However, the content in those feedback tweets is of high diversity, as it usually contains rich background information or even irrelevant topics. This problem is especially severe because of the redundancy of the tweet content. Therefore, it is extremely necessary to take advanced selection or filtration for these noisy feedbacks.

To address this problem, we propose a generative model to estimate the entity model using the observed EF based upon a regularized maximum likelihood criterion. The particular generative model we employ is a mixture of models, which incorporates not only the entity model, but also the collection language model and the domain-specific background language models. A graphical representation of the generative model is illustrated in Figure 3.

- Definition 1 (**Entity Model**): An entity model $\hat{\theta}_E$ in a twitter collection is a probabilistic distribution of words $\{p(w|\hat{\theta}_E)\}_{w \in V}$ and represents the common topics of the entity E. Clearly, we have $\sum_{w \in V} p(w|\hat{\theta}_E) = 1$.
- Definition 2 (**Collection Language Model**): A probabilistic distribution of words $\{p(w|\hat{\theta}_C)\}_{w \in V}$ is estimated based on the whole twitter collection. Also, we have $\sum_{w \in V} p(w|\hat{\theta}_C) = 1$.
- Definition 3 (**Domain-Specific Background Language Model**): A probabilistic distribution of words $\{p(w|\hat{\theta}_d)\}_{w \in V}$ is estimated based on the specific topic domain d, where $\sum_{w \in V} p(w|\hat{\theta}_d) = 1$. Here, you can think of domains as the sections in your favorite newspaper such as Business, Life Style, Arts and Economics, etc. And, the domain background language model is built for that specific domain. Note that we use the domain categories of Freebase in our approach.

By using the mixture model, the log-likelihood for the EF is:

$$\log p(EF|\hat{\theta}) = \sum_i \sum_w c(w, D_i) \cdot \log\{\lambda_E[(1-\lambda_C) \cdot p(w|\hat{\theta}_E) + \lambda_C \cdot p(w|\hat{\theta}_C)] + (1-\lambda_E) \sum_{d=1}^{k} \gamma_d p(w|\hat{\theta}_d)\} \quad (8)$$

where k is count of the corresponding domains for the entity E. Without prior expert knowledge of domains, we set k as the number of Freebase domain categories. $c(w, D_i)$ is the count of word w occurred in document D_i. Note that λ_C and λ_E are set empirically and represent the amounts of background noise and domain-specific background noise, respectively. Intuitively, when estimating the entity model, we try to "purify" the documents by eliminating some background noise along with the domain-specific background noise. The set of parameters includes:

- Entity model $\hat{\theta}_E$
- Domain-specific language models $\{\hat{\theta}_1, \ldots, \hat{\theta}_k\}$
- The domain mixture weights $\{\gamma_1, ..., \gamma_k\}$

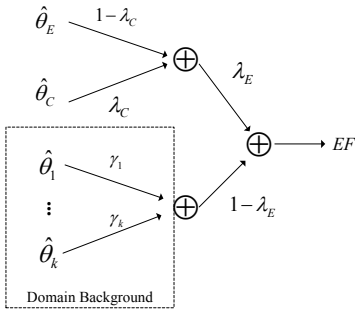

Figure 3: A graphical representation of the generative model.

We can apply EM algorithm [4] to compute a maximum likelihood estimate. The updating formulas are Eq.9-14.

4.2 Domain Priors Incorporation

Our framework aims at solving the problem with minimum supervision; however, if there exists prior expert knowledge on the structured semantics of entities, we also want to incorporate it into the model. Such prior knowledge, for example, can be obtained based on pseudo feedbacks [1] or click-through data [26] in the context of Web search. In our study, as an illustration, we seek help from Freebase, which provides a gold mine of knowledge for entities.

In Freebase, human knowledge is described by structured categories, which are also known as *types* and each type has a number of defined *properties*. Just as properties are grouped into types, types themselves are grouped into *domains*. Each entity in Freebase has been assigned into several specific domains according to expert's judgement. For instance, "Mila Kunis" is involved in seven specific Freebase domains (i.e. Film, TV, People etc.). Hence, we can set the domain number k in the mixture model as 7. In addition, we can use the type and property information of given domains to guide the discovering of domains by adding conjugate priors.

Specifically, we build a unigram language model $\{p(w|d)\}_{w \in V}$ for each pre-defined common domain d based on the information

$$t_d^{(n)}(w) = \frac{(1-\lambda_E)\gamma_d^{(n)} p^{(n)}(w|\hat{\theta}_d)}{\lambda_E[(1-\lambda_C)p^{(n)}(w|\hat{\theta}_E) + \lambda_C p(w|\hat{\theta}_C)] + (1-\lambda_E)\sum_{d'=1}^{k}\gamma_{d'}^{(n)} p^{(n)}(w|\hat{\theta}_{d'})} \tag{9}$$

$$s^{(n)}(w) = \frac{\lambda_E[(1-\lambda_C)p^{(n)}(w|\hat{\theta}_E) + \lambda_C p(w|\hat{\theta}_C)]}{\lambda_E[(1-\lambda_C)p^{(n)}(w|\hat{\theta}_E) + \lambda_C p(w|\hat{\theta}_C)] + (1-\lambda_E)\sum_{d=1}^{k}\gamma_{d}^{(n)} p^{(n)}(w|\hat{\theta}_{d})} \tag{10}$$

$$r^{(n)}(w) = \frac{(1-\lambda_C)p^{(n)}(w|\hat{\theta}_E)}{(1-\lambda_C)p^{(n)}(w|\hat{\theta}_E) + \lambda_C p(w|\hat{\theta}_C)} \tag{11}$$

$$p^{(n+1)}(w|\hat{\theta}_d) = \frac{\sum_i c(w, D_i)\cdot t_d^{(n)}(w)}{\sum_{w'}\sum_i \sum_{d'=1}^{k} c(w', D_i)\cdot t_{d'}^{(n)}(w')} \tag{12}$$

$$p^{(n+1)}(w|\hat{\theta}_E) = \frac{\sum_i c(w, D_i)\cdot r^{(n)}(w)\cdot s^{(n)}(w)}{\sum_{w'}\sum_i c(w', D_i)\cdot r^{(n)}(w')\cdot s^{(n)}(w')} \tag{13}$$

$$\gamma_d^{(n+1)} = \frac{\sum_w\sum_i c(w, D_i)\cdot t_d^{(n)}(w)}{\sum_w\sum_i \sum_{d'=1}^{k} c(w, D_i)\cdot t_{d'}^{(n)}(w)} \tag{14}$$

of Freebase entites. We first recognize all the entities in the twitter collection with the entity recognition tool described in Section 3, and then link them to Freebase with the search API [5]. After that, we collect all the text from type names and the corresponding properties of these entities, and group the text according to their related domains. In this way, we could create a virtual document for each domain. A unigram domain language model can be built in each virtual document as:

$$p(w|d) = \frac{c(w, d)}{\sum_{w'\in V} c(w', d)} \tag{15}$$

where $c(w, d)$ is the count of word w occurred in the domain virtual document d. Table 1 shows the top occurred words for several typical domains in Freebase.

Table 1: Top words in several Freebase domains.

Domain	Top Words
TV	tv, program, episode, appear, series, contribute
Organization	organization, found, origin, leader, headquarter
Chemistry	chemical, compound, measure, element, identify
Astronomy	orbit, measure, relationship, star, object
Book	author, work, written, create, book
Location	locate, part, hud, area, place

Then, we could define a conjugate prior (i.e. Dirichlet prior) on each unigram language model, parameterized as:

$$Dir(\{\sigma_d \cdot p(w \mid d) + 1\}_{w\in V}) \tag{16}$$

where σ_d is a confidence parameter for the prior. Since we use a conjugate prior, σ_d can be interpreted as the *equivalent sample size* because the effect of adding the prior would be equivalent to adding $\sigma_d \cdot p(w \mid d)$ pseudo counts for word w when we estimate the specific domain language model. Basically, the prior serves as some *training data* to intentionally bias the domain language model estimation.

[5]https://developers.google.com/freebase/v1/search-overview

To this end, the prior for all the parameters is given by:

$$p(\theta) \propto \prod_w \prod_d p(w|\theta_d)^{\sigma_d \cdot p(w|d)} \tag{17}$$

where $\sigma_i = 0$ if we do not have prior knowledge for some domain d. Then, we can use the Maximum A Posteriori (MAP) estimator to estimate all the parameters as follows:

$$\hat{\theta} = \operatorname*{argmax}_{\theta} p(E|\theta)p(\theta) \tag{18}$$

The MAP estimate can be computed using the essentially same EM algorithm as presented above with slightly different updating formula for the component language models (Eq. 19).

4.3 Entity Feedback Acquisition

According to our proposed approach, EF yields a considerable influence on the feedback entity model. Two methods (i.e. *entity match* method and *entity query* method) are therefore proposed to collect the top M feedback tweets for each entity in a given query.

4.3.1 Entity Match Method

The entity match method first identifies all the entities from each tweet with the entity recognition tool described in Section 3. Then, we can obtain the feedback tweets by collecting all the tweets containing the very entity. To maintain the temporal characters of the feedback entity model, we can use the most recent M tweets containing that entity before query issue time T_Q.

4.3.2 Entity Query Method

In this method, we regard each entity as a new query and search related tweets with a temporal based language model [10]. To this end, we enhance the selection of feedback tweets by taking into account the features related to both temporal information and relevance. In contrast to the previous method, the feedback tweets may not have an exact match for a given entity. However, this method can give rise to flexibility of balancing the tradeoff between tweet recency and relevance. To build the time based language model, we leverage Eq.1 to incorporate a time prior for each document D:

$$p(D|T_D) = r \cdot e^{-r(T_Q - T_D)} \tag{20}$$

$$p^{(n+1)}(w|\hat{\theta}_d) = \frac{\sigma_d \cdot p(w|d) + \sum_i c(w, D_i) \cdot t_d^{(n)}(w)}{\sigma_d + \sum_{w'} \sum_i \sum_{d'=1}^{k} c(w', D_i) \cdot t_{d'}^{(n)}(w')} \quad (19)$$

where r is the exponential parameter that controls the temporal influence. T_Q is the query issue time and T_D is time when the tweet was posted. Both T_Q and T_D are measured in the granularity of days. Note that T_D is constantly less than T_Q as we cannot use the future evidence. Finally, we use the top ranked M tweets as the feedback set.

5. EXPERIMENTS

In this section, we conduct several experiments to evaluate the effectiveness of our proposed adaptive language modeling method enhanced by the feedback entity model. In these experiments, we also conduct corresponding analysis to investigate: (1) the influence of the various feedback entity model parameter settings on retrieval performance; (2) the effects of feedback tweets number; (3) the influence of the interpolation coefficient in query expansion and (4) a comparison of two different entity feedback acquisition methods.

5.1 Experimental Setup

In this section, we describe the experimental setup, including the dataset and evaluation methods which are adopted in TREC Microblog track [18, 23, 13].

5.1.1 Dataset

Two corpora (i.e. Tweets11 and Tweets13 collection) are used in our experiments. Instead of distributing the microblog corpus via physical or direct downloading, TREC organizers release a streaming API to participants [13]. Using the official API [6], we crawled a set of local copies of the canonical corpora. Our local Tweets11 collection has a sample of about 16 million tweets, ranging from January 24, 2011 to February 8, 2011 while Tweets13 collection contains about 259 million tweets, ranging from February 1, 2013 to March 31, 2013 (inclusive). In addition, we also crawled all the shortened URLs contained in Tweets11 and Tweets13 Corpora, and inferred their topic information to enrich the original tweets. In particular, we consider the topic information as the local context of the original tweets and combine it with the original tweets to form the tweet language model [12]. Tweets11 is used for evaluating the effectiveness of the proposed Twitter search systems over 50 official topics in the TREC 2011 Microblog track as well as 60 official topics in the TREC 2012 Microblog track, respectively. And, Tweets13 is used in evaluating the proposed Twitter search systems over 60 official topics in the TREC 2013 Microblog track. In our experiments, we only make use of those topics containing entities. The topics in TREC 2011 are used for tuning the parameters and then we use the best parameter setting to evaluate our methods with topics for TREC 2012 and TREC 2013. Table 2 summarizes basic statistics of the three years' topics. From the table, we can observe that half of the topics include entities, which is quite consistent with reality.

In our experiments, the tweets and their corresponding topic information would be preprocessed in several steps. First, we discarded those non-English tweets using a language detector with infinity-gram, named ldig[7]. After that, according to the Microblog track's guidelines, all simple retweets, i.e. tweets beginning with the string 'RT', were removed. Moreover, each tweet was stemmed using the Porter algorithm and stopwords were removed using the InQuery stopwords list. Table 3 summarizes basic statistics of the two corpora after all the steps of preprocessing. In TREC's temporally-anchored ad hoc search task, track organizers created several test topics, each of which contains query text Q and a timestamp T_Q. Only tweets posted prior to T_Q were assessed for relevance [18]. Thus for each query Q, we built a dynamic dataset consisting of tweets whose timestamps are prior to T_Q.

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

[7]http://github.com/shuyo/ldig

Table 2: Summary statistics of topics in TREC Microblog track.

Year	2011	2012	2013
Total number of topics	50	60	60
Topics including entities	36	36	34

Table 3: Summary statistics of Tweets11 and Tweets13 corpora.

Corpus	Tweets11	Tweets13
Tweets after preprocessing	4,948,137	68,682,325
Tweets containing entities	1,422,292 (29%)	13,480,936 (20%)

5.1.2 Evaluation Metric

In TREC Microblog track, tweets are judged on the basis of the defined information using a three-point scale [18]: irrelevant (labeled as 0), minimally relevant (labeled as 1), and highly relevant (labeled as 2). In our experiments, we mainly leverage two widely-used evaluation metrics in IR, including Mean Average Precision (MAP) and Precision at N (P@N). Specifically, MAP for top 1000 ranked documents and P@30 with respect to *allrel* (i.e. tweet set labeled as 1 or 2) are the official main metrics for the temporally-anchored ad hoc search task in TREC, which are also used in this paper. Furthermore, we also do a query-by-query analysis and conduct t-test to determine whether the improvements on MAP and P@N are statistically significant.

5.2 Baselines

To demonstrate the superiority of our feedback entity model in query expansion for microblog retrieval, we compare our adaptive language model with several baseline methods.

1) The simple KL-divergence (denoted as **SimpleKL**) [28] is used as our first baseline. **SimpleKL** estimates both $\hat{\theta}_Q$ and $\hat{\theta}_D$ with empirical word distribution, in which we choose Dirichlet smoothing method for document model estimation. Throughout this paper, we set the Dirichlet smoothing parameter $\mu = 100$.

2) In addition, we implement Efron's hashtag-based relevance feedback method HFB1 [6] as an entity-based query expansion baseline (labeled as **QEHashtag**). The model parameter k is set as 25 and λ is set as 0.2.

3) To compare with the external query expansion method, we employ a Wikipedia-based query expansion method **QEWiki**, which is similar with the work of [11]. We downloaded a local copy of Wikipedia data for faster access and indexed the articles

using Lemur toolkit [8] (version 4.12). The expansion terms are derived from top ranked Wikipedia articles. In our experiments, we rank Wikipedia articles using language model (i.e. SimpleKL), and 10 terms with highest TFIDF scores are picked from the top 5 articles. Then we treat the terms as a new query and interpolate it with the original query with a weight of 0.5.

4) Simple Mixture Model [27] (denoted as **SimpleKL + SMM**) is used as another baseline. **SimpleKL + SMM** is reported with a relatively better retrieval performance among the state-of-the-art PRF-based query expansion methods [14]. The number of feedback documents is set as 5 and the number of terms in the feedback model is set as 7. The interpolation parameter β is set as 0.6. Note that the feedback tweets used for **SimpleKL + SMM** are derived from the initial search results, which often contain many irrelevant documents. Hence, we replace the feedback tweets with the top retrieval results using **QEHashtag** and **QEWiki**. In this way, we generate two additional baselines, i.e. **QEHashtag + SMM** and **QEWiki + SMM**.

5) Finally, we compare our approach with the state-of-the-art real-time ranking model (denoted as **RTRM**), proposed in [12]. **RTRM** also adopts a two-stage query expansion method like our proposed adaptive language model does. The first stage query expansion weight is set as 0.4. For the second stage query expansion, we extract the top 7 feedback terms from the top 5 tweets and set the interpolation weight as 0.6. In addition, it utilizes ranking position as temporal profile, and adopts the Gaussian temporal re-ranking function with $\sigma = 120$.

All the parameters of these baseline methods are tuned using TREC 2011 topics.

5.3 Experimental Results

In this section, we report the experimental results to demonstrate the effectiveness of our proposed feedback entity model based query expansion methods. In the following, we denote the KL-divergence retrieval model with feedback entity model based query expansion for query model $\hat{\theta}_{Q'}$ as **QEFEM**, and that for $\hat{\theta}_{Q''}$ as **QEFEM + SMM**. When estimating the entity model, we set feedback document count M as 50 with entity match method. The model parameter λ_E is set as 0.7 and λ_C is set as 0.5. The first interpolation coefficient α is set as 0.6 in Eq.3 for both **QEFEM** and **QEFEM + SMM**. For the simple mixture model, we use top 5 feedback documents, while setting β to 0.6 in Eq.5. For the adaptive language model with query model $\hat{\theta}_{Q'}$ and fixed interpolation coefficient α, we label it as **QEStaticFEM** and set $\alpha = 0.4$ for all queries in Eq.3. All these parameters are tuned using topics in TREC 2011.

5.3.1 Overall Results

Table 4 shows the MAP and P@30 performances of eight methods with statistical significance test results for *allrel* tweets. The best performances are marked in bold typeface. Note that all the methods listed in the table estimate the document model as **SimpleKL**.

We first examine the effectiveness of our adaptive language model using first stage query expansion (i.e. **QEFEM**). †, ‡ and ¶ indicate that the corresponding improvements over **SimpleKL**, **QEHashtag** and **QEWiki** are statistically significant ($p < 0.05$), respectively. It can be clearly observed from Table 4 that all these query expansion methods can result in improvements in terms of MAP compared with the **SimpleKL** method for both TREC 2012 and 2013 topics, which indicates the importance of query expansion in microblog search. Besides, **QEFEM** performs better than hashtag-based query expansion method **QEHashtag**, which indicates the importance of analyzing entities in twitter rather than simply using the entity terms. Moreover, **QEFEM** is better than the external expansion method **QEWiki** in terms of both MAP and P@30. This indicates that **QEFEM** can get purer entity-associated topic terms and thus leads to higher precision in top retrieved results. For TREC 2012 topics, **QEFEM** improves the P@30 and MAP scores significantly compared with the three baselines.

When the query is expanded with the feedback entity model, our approach can retrieve more relevant documents at the top, which contain more accurate word distribution than by initial search (i.e. search results by **SimpleKL**). Thus, we can further improve the retrieval performance by combining the model-based feedback method as second stage query expansion. Table 4 also shows all the performances using two-stage query expansion methods. §, △ and ▲ indicate that the corresponding improvements over **SimpleKL + SMM**, **QEHashtag + SMM** and **QEWiki + SMM** are statistically significant ($p < 0.05$), respectively. Note that our **QEFEM + SMM** method outperforms the baseline **SimpleKL + SMM** significantly. More specifically, for TREC 2012 topics, the **QEFEM + SMM** improves the MAP and P@30 over those of **SimpleKL + SMM** by 21.8% and 13.2%, respectively; while the corresponding increments for TREC 2013 topics are 9.5% and 6.5%, respectively. Besides, **QEFEM + SMM** also performs betther than the other two-stage query expansion method **QEHashtag + SMM** and **QEWiki + SMM**. This proves the superiority of our feedback entity model based query expansion method.

Table 5: Performance comparison of the adaptive language model with RTRM.

Topics	TREC 2012		TREC 2013	
Method	**MAP**	**P@30**	**MAP**	**P@30**
RTRM	0.3397	0.4356	0.3382	0.5207
QEFEM + SMM	**0.3471**	**0.4567**	**0.3519**	**0.5253**

Finally, we compare our adaptive language model with the state-of-the-art baseline **RTRM** in Table 5. RTRM is reported to acheive the best MAP score in TREC 2011 Microblog Track [12]. It can be clearly observed that **QEFEM + SMM** even performs better than **RTRM**, which demonstrates the effectiveness of our approach.

5.3.2 Effects of the Adaptive Weighting Parameter

In Table 6, we list the performances of our adaptive language model with both static weighting parameter (i.e. **QEStaticFEM**) and adaptive weighting parameter (i.e. **QEFEM**). From Table 6, we can observe that both of the methods perform well with respect to MAP and P@30. Besides, we can obtain additional performance improvement when using the adaptive weighting parameter α. This proves our assumption that entities in different queries have different influence, and thus we should adopt an adaptive parameter to balance the entity feedback and the original query.

Table 6: Performance comparison of the adaptive language model with both static weighting parameter and adaptive weighting parameter.

Topics	TREC 2012		TREC 2013	
Method	**MAP**	**P@30**	**MAP**	**P@30**
SimpleKL	0.2713	0.4067	0.3070	0.4931
QEStaticFEM	0.3053	0.4578	0.3130	0.4954
QEFEM	**0.3180**	**0.4611**	**0.3222**	**0.5057**

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

Table 4: Performance comparison of the proposed methods and baselines for allrel documents.

Topics	TREC 2012		TREC 2013	
Method	**MAP**	**P@30**	**MAP**	**P@30**
SimpleKL	0.2713	0.4067	0.3070	0.4931
QEHashtag	0.2717	0.4122	0.3162	0.5034
QEWiki	0.2901	0.4244	0.3180	0.5023
QEFEM	0.3180 †‡¶	**0.4611** †‡¶	0.3222 †	0.5057 †
SimpleKL + SMM	0.2849	0.4033	0.3214	0.4931
QEHashtag + SMM	0.2831	0.4144	0.3293	0.5126
QEWiki + SMM	0.3267	0.4311	0.3331	0.5149
QEFEM + SMM	**0.3471** §△▲	0.4567 §△▲	**0.3519** §△▲	**0.5253** §

To further study what percentage of entity queries are enhanced by the adaptive weighting parameter, we conduct a query by query performance analysis using TREC 2012-2013 topics involving entities. Figure 4 shows the performance difference between **QEFEM** and **QEStaticFEM** in terms of MAP. It is apparent that using adaptive weighting parameter is effective for improving most queries containing entities.

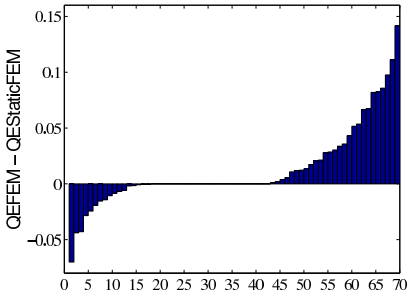

Figure 4: Difference in MAP between QEFEM and QEStaticFEM using the TREC 2012-2013 Microblog track topics.

5.4 Discussion

Many parameters in our proposed approach can affect the system performance. In this section, we analyze the robustness of the parameter setting in our adaptive language model. For the **QEFEM + SMM**, we expand the original query with feedback entity model as **QEFEM** does, and it also applies the traditional model-based feedback to further update the query. We set the second-stage query expansion parameter β as 0.6 and feedback document number as 5, which are reported with good retrieval performance in microblog search [12]. All these experiments in this section are run on TREC 2011 topics, which are used for parameter selection. Note that we first conduct a grid search on λ_C, λ_E, M and α in order to find the optimal parameters, then we check the sensitivity of each parameter when setting others as optimal values.

5.4.1 Effects of Feedback Entity Model Parameters

There are many parameters that may influence the effectiveness of feedback entity model in query expansion. To estimate the entity model, the parameter λ_E controls the amount of "domain background noise" while the parameter λ_C controls the amount of "collection background noise". The number of feedback documents for each entity is also very critical since it directly affects the available terms for the entity. In this section, for each query, we collect the latest 50 entity-associated feedback tweets, using entity match method for model estimation. When incorporating the domain prior information in Freebase, we simply set all the Dirichlet parameters $\sigma_i (i = 1, ..., k)$ as 1000.

To evaluate the sensitivity of retrieval performance to the two feedback model parameters λ_C and λ_E, we fix the interpolation coefficient α as 0.6. In the actual experiments, we truncated the estimated entity model by ignoring all terms whose probability is lower than 0.001, and renormalized it before interpolating. Then we conducted a grid search on λ_C and λ_E. Figure 5 shows the performance changes of the **QEFEM** and **QEFEM + SMM** against different λ_E while setting λ_C to a fixed value 0.5. A smaller λ_E can filter more domain background noise when estimating the entity model. However, from the figure, we can find that it might also decrease the retrieval performance when ignoring too many domain-specific words. In general, the performance change of the **QEFEM** with different λ_E is slight. However, if we set λ_E as 1 and totally ignore the domain-specific language models, the performance of **QEFEM** declines a lot in terms of MAP, which proves the importance of domain background noise filtration.

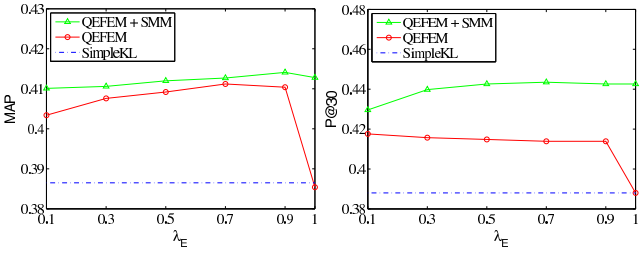

Figure 5: Sensitivity of retrieval performance to λ_E.

We also conduct an experiment to analyze the performance change of that **QEFEM** against different values of λ_C. Experimental results show the retrieval performance is not very sensitive to λ_C, while totally ignoring the collection background model would lead to a performance drop.

To further demonstrate the effectiveness of our mixture model, we compare the best tuned **QEFEM** (i.e. $\lambda_E = 0.7, \lambda_C = 0.5$, denoted as **QEFEM-Filtering**) with **QEFEM** without any noise filtration (i.e. $\lambda_E = 1, \lambda_C = 0$, denoted as **QEFEM-ML**). From Figure 6, we can observe that, after filtering the domain and background noise, we can increase both P@10 and P@30 scores, compared with the baseline method and the entity model without noise filtration.

5.4.2 Effects of Feedback Tweets Number

Another important parameter is the number of entity-associated feedback tweets for model estimation. Too few feedback tweets may not be adequate to summarize the entity, while too many entity related tweets may give rise to much noise for the entity model and

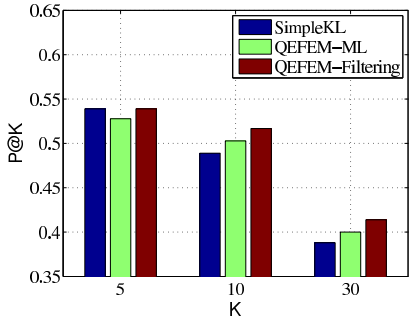

Figure 6: Precision at K for the methods of QEFEM-ML and QEFEM-Filtering.

it will decrease the efficiency if applying the EM algorithm. Figure 7 shows the performance change of **QEFEM** and **QEFEM + SMM** against different settings of the number of feedback tweets, i.e. M. From the figure, we can observe that either too small or too large M will lead to a performance drop of **QEFEM** in terms of MAP. This indicates the importance of choosing an appropriate M.

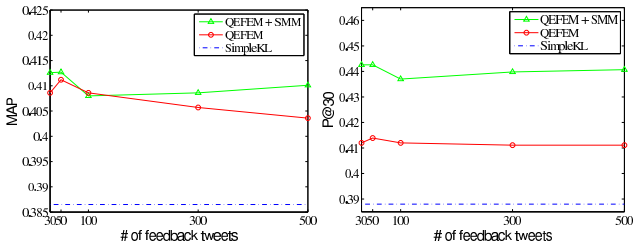

Figure 7: Sensitivity of retrieval performance to # of feedback tweets M.

5.4.3 *Effects of the Interpolation Coefficient*

Recall that we interpolate the estimated feedback entity model with the original maximum likelihood model estimated on the plain query text. The interpolation is controlled by an adaptive coefficient. When $\alpha = 0$, we only use the original query model (i.e. $\hat{\theta}_Q$); when $\alpha = 1$, our approach emphasizes the feedback entity model most. Actually, for queries which are entities themselves, our method of $\alpha = 1$ simply uses the entity model while ignoring the original query model (i.e. $\alpha' = 1$). For the feedback entity model estimation, we set λ_E as 0.7, λ_C as 0.5 and use the top 50 entity-associated tweets to estimate the model. Then, we evaluate the performance with different α varying from 0 to 1.

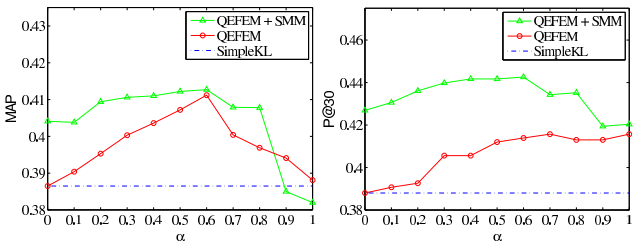

Figure 8: Sensitivity of retrieval performance to α.

Figure 8 shows the performance changes of **QEFEM** and **QEFEM + SMM** against different values of α. Note that, when $\alpha = 0$, **QEFEM** degenerates to the baseline method **SimpleKL**, while **QEFEM + SMM** degenerates to **SimpleKL + SMM**. From the figure, we can observe that the value of α yields a significant effect on the retrieval performance. It is important to choose an appropriate α. Since the entity model can only represent the entity part of the original query, setting α too large will result in a performance drop as it leads to much information loss about the original query. When setting α around 0.6, **QEFEM** and **QEFEM + SMM** can reach their optimal MAP scores. Besides, with the feedback entity model, the search engines can get more high-quality feedback tweets for the query. This could benefit the traditional PRF-based query expansion methods to gain additional performance improvements.

5.4.4 *Comparison of Entity Feedback Acquisition Methods*

Note that we propose two methods to fetch the feedback tweets, i.e. entity match method and entity query method. In the previous sections, we mainly use the entity match method. In the following, we take further discussion on the main differences between these two methods. First, when using the entity match method, those tweets which have an exact entity match can be fetched. When using the entity query method, we can get much more tweets since the partial match can also fetch back the related tweets. However, this could lead to a high probability of introducing irrelevant entities. Second, to reflect the temporal aspect of the entity model, the entity match method simply sorts the tweets by chronological order while the query match method incorporates the recency information into a temporal prior in the language modeling framework.

We evaluate the performance of **QEFEM** using entity match and entity query methods. Both of them use top 50 feedback tweets. The temporal prior r is set as 0.01 for the entity query method. The average and minimum count of feedback tweets fetched with different methods ($M = 50$) are summarized in Table 7. For the model estimation, we set parameter $\lambda_E = 0.7$ and $\lambda_C = 0.5$. The interpolation coefficient α is set as 0.6.

Table 7: Feedback tweets count analysis using different entity feedback acquisition methods.

Method	Average Count	Minimum Count
entity match	34	2
entity query	46	17

Table 8 shows the performance comparison of **QEFEM** using the two feedback acquisition methods. From the table, we can observe that both of the methods can improve the retrieval performance significantly compared with the baseline method. This indicates the effectiveness of both methods. Besides, the retrieval performances of the two methods are comparative across all the evaluation metrics.

Table 8: Performance comparison of different entity feedback acquisition methods.

Method	MAP	P@30	P@10
SimpleKL	0.3865	0.3880	0.4889
QEFEM with entity match	0.4112	0.4139	0.5167
QEFEM with entity query	0.4056	0.4111	0.5111

6. CONCLUSION AND FUTURE WORK

In this study, we propose to use feedback entity model to utilize the rich entity information in Twitter and solve the challenges such as the vocabulary mismatch problem in microblog search. By incorporating the feedback entity model into language modeling framework, the queries containing entities can be more comprehensible and thus more relevant documents can be retrieved. As a result, combining simple mixture model can gain further performance improvement as the feedback tweets contain more accurate word distribution than by initial search. Moreover, as we track the latest tweets containing the entity to build a feedback entity model, our method could also satisfy the real-time information need in microblog retrieval. Our thorough evaluation, using two standard TREC collections, demonstrates the effectiveness of the proposed method.

Many studies remain for the future work. (1) One of the most interesting directions is to generalize our method for topics not containing any entity by searching for top related entities (i.e. pseudo named entities of topics). (2) In this paper, we simply use the general entity model to update the query. In fact, entities can be ambiguous in different domains (e.g. apple in food and technology domains) and can have quite different word distribution in each domain. Thus, if we can recognize the concerning domain of the user query, we can use the domain-specific entity model to update the query. (3) Besides, when using entity model to update the query, we use an adaptive parameter α which only takes account of the IDF information of entity and query terms. However, due to the difference in the entities and entity-associated tweets, this balance parameter can be optimized for each entity and feedback tweets with a supervised method.

7. ACKNOWLEDGMENTS

The work reported in this paper is supported by the National Natural Science Foundation of China Grant 61370116. We thank anonymous reviewers for their beneficial comments. We also thank Jiang Bian, Lili Yao and Yue Fei for valuable suggestions related to this paper.

8. REFERENCES

[1] A. Z. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. Robust classification of rare queries using web knowledge. In *SIGIR*, pages 231–238, New York, NY, USA, 2007. ACM.

[2] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR*, pages 243–250, 2008.

[3] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. 2014.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistcal Society*, 39(B):1–38, 1977.

[5] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR*, pages 154–161. ACM, 2006.

[6] M. Efron. Hashtag retrieval in a microblogging environment. In *SIGIR*, pages 787–788, 2010.

[7] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, and J. Zobel, editors, *SIGIR*, pages 267–274. ACM, 2009.

[8] M. Keikha, S. Gerani, and F. Crestani. Time-based relevance models. In *SIGIR*, pages 1087–1088. ACM, 2011.

[9] J. D. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*, pages 111–119, 2001.

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

[11] Y. Li, W. P. R. Luk, K. S. E. Ho, and F. L. K. Chung. Improving weak ad-hoc queries using wikipedia as external corpus. In *SIGIR*, pages 797–798, 2007.

[12] F. Liang, R. Qiang, and J. Yang. Exploiting real-time information retrieval in the microblogosphere. In *JCDL*, pages 267–276, 2012.

[13] J. Lin and M. Efron. Overview of the TREC-2013 Microblog Track. In *TREC'13*, 2014.

[14] Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *CIKM*, pages 1895–1898, 2009.

[15] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *ECIR*, volume 6611 of *Lecture Notes in Computer Science*, pages 362–367. Springer, 2011.

[16] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *SIGIR*, pages 206–214. ACM, 1998.

[17] T. Miyanishi, K. Seki, and K. Uehara. Improving pseudo-relevance feedback via tweet selection. In *CIKM*, pages 439–448, 2013.

[18] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *TREC'11*, 2012.

[19] D. Pan, P. Zhang, J. Li, D. Song, J.-R. Wen, Y. Hou, B. Hu, Y. Jia, and A. N. D. Roeck. Using dempster-shafer's evidence theory for query expansion based on freebase knowledge. In *AIRS*, volume 8281 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2013.

[20] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256. Association for Computational Linguistics, 2009.

[21] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.

[22] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *KDD*, 2012.

[23] I. Soboroff, I. Ounis, and J. Lin. Overview of the TREC-2012 Microblog Track. In *TREC'12*, 2013.

[24] N. Stokes, Y. Li, L. Cavedon, E. Huang, J. Rong, and J. Zobel. Entity-based relevance feedback for genomic list answer retrieval. In *TREC*, 2007.

[25] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *WSDM*, pages 35–44, 2011.

[26] J.-R. Wen, H. Zhang, and J.-Y. Nie. Query clustering using content words and user feedback. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *SIGIR*, pages 442–443. ACM, 2001.

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

[28] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342, 2001.