Temporal Models for Microblogs

Jaeho Choi
NHN Corporation
NHN Green Factory, 178-1
Seongnam, Korea
jaehos@nhn.com

W. Bruce Croft
Dept. of Computer Science
Univ. of Massachusetts Amherst
Amherst, MA
croft@cs.umass.edu

ABSTRACT
Time information impacts relevance in retrieval for the queries that are sensitive to trends and events. Microblog services particularly focused on recent news and events so dealing with the temporal aspects of microblogs is essential for providing effective retrieval. Recent work on time-based retrieval has shown that selecting the relevant time period for query expansion is promising. In this paper, we suggest a method for selecting the time period for query expansion based on a user behavior (i.e., retweets) that can be collected easily. We then use these time periods for query expansion in a pseudo-relevance feedback setting. More specifically, we use the difference in the temporal distribution between the top retrieved documents and retweets. The experimental results based on the TREC Microblog collection show that our method for selecting periods for query expansion improves retrieval performance compared to another approach.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval

General Terms
Algorithms, Experimentation

Keywords
Time-based model, microblogs, query expansion

1. INTRODUCTION
Time information impacts relevance in retrieval for queries that are sensitive to trends and events. A microblog (e.g., Twitter) is a medium where users post short messages to broadcast current events or their personal opinions. In particular, microblog services focus on recent issues, since users in a microblog community can express opinions and discuss social issues with other users immediately. Due to this highly temporal nature, incorporating time information into ranking is crucial in microblog retrieval. Microblog ad-hoc retrieval, in general, aims to find relevant and the most recent content for the queries that are related to social issues [18]. The TREC 2011 microblog track pursued a similar goal where TREC provided topics with each query's timestamp that indicates when this query was issued. According to the guidelines, no documents newer than a given query's timestamp should be retrieved and the final ranking should be sorted in descending chronological order (from the latest to the oldest). Given these assumptions, modeling the temporal aspects of microblog queries is a significant issue for this task.

In previous work, researchers focused on identifying recency-sensitive (or temporal) queries and incorporating time into the retrieval model. For example, Li and Croft [10] defined two types of time-based queries in TREC volumes that contain many news documents and proposed a time-based language model. Diaz and Jones [3] proposed a temporal profile of the query to predict retrieval performance in a TREC news collection. They then used the temporal profile to classify the query into three temporal query classes [6]. The advent of social media (e.g., blogs, microblogs) has increased the interest in time-based retrieval models. Recent studies on time-based models in IR have focused on using temporal distributions of retrieved documents in the pseudo-relevance feedback setting [2,7,12,15]. This work has shown that selecting a *relevant time period* for a specific query, and then extracting expanded terms by using weights derived from the relevant time can improve retrieval performance.

Here, we hypothesize that a user behavior (i.e., retweeting) can indicate the relevant time period for a query. As we mentioned, microblog services mostly involve inter-communication between users. In particular, information propagation through forwarding other users' content is well-known as a prominent characteristic of microblog services. Indeed, we found that retweeting can be used for identifying the relevant time period for temporal queries.

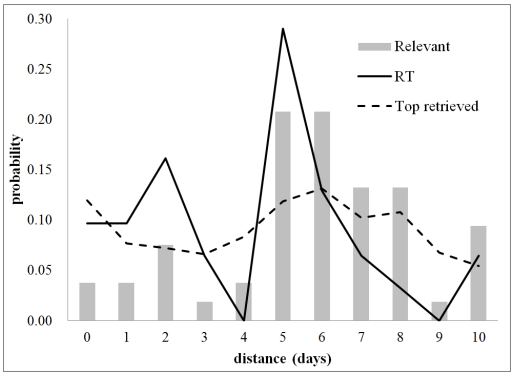

Figure 1: The temporal distribution ("Mexico drug war")

For example, in Figure 1, we display the temporal distribution of relevant documents, top retrieved documents, and retweets (RT) for a TREC query "Mexico drug war". This graph shows us that RT (bold line) tends more closely follow the occurrence of the relevant documents. More specifically, we use the difference in

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

the temporal distribution between the top retrieved documents and retweets to find the relevant time period. We then do query expansion from the top retrieved documents in that time period. We evaluated with TREC Microblog track data, and the results show that our approach improves the retrieval performance against strong baselines. Further analysis shows that our method extracts more relevant terms for the query by selecting relevant time periods and using only the documents in those time periods.

2. RELATED WORK

Previous work has studied time-based models and our approach is related to some of this work. Li and Croft [10] defined two types of time-based queries in TREC collections that contain news archives: one always favors the most recent documents and the other has relevant documents within a specific period in the past. To incorporate time information into retrieval models, they proposed a time-based language model using a prior based on an exponential or a normal distribution depending on the types of recency queries. Efron and Golovchinsky [4] proposed an estimator for the rate parameter of an exponential distribution that incorporates query-specific information. They also suggested a time-smoothing language model that uses a time factor to estimate the mixing parameter for language model smoothing.

Diaz and Jones [3] proposed a temporal query model, denoted *P(t|Q)*, which is defined as the normalized sum of the relevance scores of retrieved documents that are published at time *t* for query *Q*. They used temporal features for query performance prediction [3] and temporal query classification [6] tasks. Keikha et al. proposed a time-based relevance model [7] for blog feed retrieval, which uses the *P(t|Q)* introduced in [3] as a weight of the terms in the pseudo-relevance feedback setting. In this work, we extend the framework of the time-based relevance model to incorporate the temporal factor into ranking. That is, we estimate *P(t|Q)* by using the temporal distribution of retweets instead of using the normalized sum of the relevance scores.

Dakka et al. [2] suggested a general framework to estimate *P(t|Q)*. They arranged the top retrieved documents into bins and assigned estimated relevance value to these bins. Peetz et al. presented an adaptive temporal query modeling [15] for blog feed retrieval, in that they analyzed the top retrieved documents in terms of temporal histogram to find the bursts. They used documents with the highest scores from the bursts for query expansion and weighted each feedback document with the distance from the peak that contains most documents.

Massoudi et al. [11] proposed a query expansion model for microblogs, which selects terms temporally closer to the query submission time. Their model is supposed to work well for finding documents related to events currently happening but, not as well for past events. We found that many topic queries were related to events occurring in the past rather than the query time. Metzler et al. [12] proposed a temporal query expansion method for microblogs based on the temporal co-occurrence of terms in a timespan. They first performed pseudo-relevant timespan retrieval for an event query (e.g., earthquake) and used those timespans for query expansion. Although their goal was retrieving a ranked list of historical event summaries, the temporal query expansion method showed that selecting relevant timespan is crucial for query expansion for microblog documents. If the temporal query expansion works for an event query, it might be useful for ad-hoc search queries.

3. TEMPORAL MODELS FOR MICROBLOG

Our temporal model for microblogs builds upon a time-based relevance model [7] that incorporates time factors into the language model framework. In this section, we first introduce the time-based relevance model in detail and then, suggest a method for selecting the relevant time using retweets for the query.

3.1 Time-based Relevance Model

Microblog documents contain many cases of word variations, hashtags, and internet slangs. This increases the vocabulary mismatch problem [12] in microblog retrieval. In our preliminary experiments, we found that relevance modeling [9] is helpful for this problem, since it can potentially address issues related to synonymy and polysemy. The pseudo-relevance model generates expansion terms based on the top *k* retrieved documents (denote *R*) as Eq. (1)

$$P(w|R) \approx P(w|Q) \propto \sum_{d \in R} P(w|d)P(d)\prod_{k} P(q_k|d) \qquad (1)$$

Keikha et al. [7] proposed a time-based relevance model which incorporates time factor into relevance model framework. They introduced a generative model of the query that first selects a time and then selects a term based on the time and query as Eq. (2)

$$P(w|Q) = \sum_{t} P(w|t,Q)P(t|Q) \qquad (2)$$

P(w|t,Q) can be computed by product sum of *P(w|d)* and *P(d|t,Q)* over all relevant documents published in time *t* (denoted R_t). Unlike *P(d|t,Q)* was set to be uniform [7], in this work, we extend their framework. Based on a simplifying assumption, *P(d|t,Q)* can be equal to *P(d|Q)* since the time *t* is already encoded in choosing *d*. Therefore, we get the Eq. (3) by using Bayes rule and independence assumption between the query terms, where *P(Q)* is eliminated based on the rank equivalence.

$$\begin{aligned} & P(w|t,Q) \\ &= \sum_{d \in R_t} P(w|d)P(d|t,Q) \\ &= \sum_{d \in R_t} P(w|d)P(d|Q) \\ &\propto \sum_{d \in R_t} P(w|d)P(d)\prod_{k} P(q_k|d) \end{aligned} \qquad (3)$$

When we substitute the Eq. (3) into Eq. (2), we get the following final equation, Eq (4).

$$P(w|Q) \propto \sum_{t} P(t|Q) \sum_{d \in R_t} P(w|d)P(d)\prod_{k} P(q_k|d) \qquad (4)$$

This formulation allow us to extract the expanded terms from the relevant documents published in time *t*, weighted by *P(t|Q)*, an arbitrary temporal model of the query. In next section, we suggest a novel method for estimating *P(t|Q)*.

3.2 Temporal Models for Microblogs

Microblog users often quote or forward other users' content (e.g., retweeting). We might think of some influences on retweeting such as content, network, and temporal influence [16]. For example, a famous celebrity's tweet can be retweeted often due to the popularity of the user (network influence). People also tend to broadcast newsworthy tweets (content influence). A recent study [8] reported that 75% of retweets occur within a day (temporal influence) after posting. In other words, old content is unlikely to

be retweeted. One possible reason for this behavior is because microblog documents can be written in a very short time and also tend to be ephemeral, in contrast, other user-generated content (e.g., blog) or news needs enough time to be published. Indeed, retweeting can be done by a simple click. Due to these properties of retweets, we can estimate the time period when an event happened and people discussed that issue heavily in the past, and we can then consider this as a relevant time period for the query.

To identify the relevant time period for the query, we need the top *N* retrieved documents returned by an initial retrieval model. We found that our method performs well when *N*=500. We compute *P(t|RT,Q)* and *P(t|D,Q)* as described in Eq. (5) and Eq. (6), respectively, where *#docs(t,RT,Q)* is the number of retweets posted at time *t* in top *N* retrieved documents returned by query *Q* and *#docs(t,D,Q)* is the number of documents posted at time *t* in top *N* retrieved documents returned by query *Q*. In this work, the unit of temporal granularity is a day.

$$P(t|RT,Q) = \frac{\#docs(t,RT,Q)}{\sum_{t'} \#docs(t',RT,Q)} \quad (5)$$

$$P(t|D,Q) = \frac{\#docs(t,D,Q)}{\sum_{t'} \#docs(t',D,Q)} \quad (6)$$

Given these probabilities, we define an indicator function ϕ that has the value 1 if *P(t|RT,Q)* is higher than *P(t|D,Q)*, and 0 otherwise. We normalize it by sum of ϕ for all time *t* and get the *P(t|Q)* as Eq. (7).

$$\phi(t,Q) = \begin{cases} 1 & ,if\ P(t|RT,Q) > P(t|D,Q) \\ 0 & ,otherwise \end{cases}$$

$$P(t|Q) = \frac{\phi(t,Q)}{\sum_{t'} \phi(t',Q)} \quad (7)$$

Note that substituting Eq. (7) into Eq. (4) indicates that only the document posted in that time *t* will be used for query expansion. In other words, we construct the pseudo-relevance feedback set with the documents occurred in the most active days in terms of retweeting. We exclude the retweets from the pseudo-relevance feedback set on purpose, since TREC assessors explicitly judged all retweets as non-relevant. The number of the documents considered in the expansion (i.e., fbDocs) and the number of expanded terms (i.e., fbTerm) are tuned in 5-fold cross-validation in our experiments. If there is no relevant time period for a query, that is, all *P(t|Q)* equals to zero for all time *t*, we use all feedback documents for query expansion, just as the original pseudo-relevance model (back-off model).

4. EVALUATION

4.1 Experimental Setup

We used the TREC 2011 Microblog collection for evaluation. This collection consists of approximately 16 million tweets and 49 topics (MB050 topic omitted due to the absence of relevant tweet). Since TREC judged non-English tweets as non-relevant, we filtered out non-English tweets by using both the language property (i.e., lang=en) and characters-set (i.e., ASCII). We also constructed an English word dictionary from the WSJ 87-92 newswire collections, and eliminated documents whose fraction of words that exist in the dictionary is less than 0.5. As a result, 46.1% tweets were removed in total. To satisfy the constraint in terms of final ranking order, we first rank the top k results based on the relevance score and re-sort them in descending chronological order based on tweet id. We index all tweets using the Galago retrieval system, and stem with the Porter2 stemmer. We use a stopword list which is constructed from web corpus [1]. To tune the language model parameters (e.g., μ), we used a training corpus with 59 topics and approximately 5,900 relevance scores that were manually judged by twelve computer science students. The training corpus consists of 17M tweets that had been crawled using the Twitter API. We found that there were no common tweets between the training and the evaluation data. We annotate retweets in two ways. First, we match *RT signatures*, (i.e., *RT* @username) on the tweet content [5,14]. Second, we use the retweet count in the metadata provided in JSON format. This retweet count is counted only when the retweet button in the Twitter service is clicked. We do not consider the number of times a document is retweeted in this work. We exclude the retweets that contain few query terms, for example, the retweet should contain all query terms if the number of query terms is less than two. If the number of query term is more than two, the retweet should not omit more than one query term.

We use six variants of language models as our baselines: a query likelihood language model with Dirichlet smoothing (**QL**), a sequential dependence model (**SDM**) [13], a full dependence model (**FDM**) [13], a relevance model (**RM**), a relevance model based on SDM (**SDRM**), and a relevance model based on FDM (**FDRM**). We denote the performance of our approach based on a relevance model as **RM-T**. Accordingly, **SDRM-T** and **FDRM-T** stand for the performance of our approach based on SDRM and FDRM respectively. In addition, we add a time-based relevance model (**TBRM**) [7] as our strong baseline. Similarly, **TBRM-SD** and **TBRM-FD** stand for the performance of the time-based relevance model based on SDRM and FDRM respectively. To evaluate the performance, we used two measures, MAP and precision at 30 (P@30). P@30 was used as the official measurement in the TREC 2011 Microblog track ad-hoc task.

4.2 Experimental Results

We display the performance of the baselines and our approach in Table 1. The results show that pseudo-relevance feedback models (e.g., RM, SDRM, and FDRM) perform better than other baseline language models (e.g., QL, SDM, and FDM). This supports our hypothesis that a relevance model can potentially address the vocabulary mismatch problem.

Table 1: The performance results

Model	#Rel@30	MAP	P30
QL	588	0.2326	0.4000
SDM	611	0.2304	0.4156
FDM	630	0.2436	0.4286
RM	712	0.2677	0.4844
TBRM	738	0.2651	**0.5020***
RM-T	756	0.3076	**0.5143***
SDRM	723	0.2604	0.4918
TBRM-SD	746	0.2712	**0.5075***
SDRM-T	767	0.2870	**0.5218***
FDRM	749	0.2838	0.5095
TBRM-FD	776	0.2864	**0.5279***
FDRM-T	798	**0.3226*†**	**0.5429***

The results also show that our approach improves retrieval performance in all cases. An asterisk denotes significant difference compared to pseudo-relevance model baselines and a plus denotes significant difference compared to time-based relevance model baselines by two sided paired randomization test [17] (p-value<0.05). Significant differences in precision at 30

were observed in RM-T, SDRM-T, and FDRM-T compared to RM, SDRM, and FDRM respectively. We note that our approach outperforms time-based relevance models (i.e., TBRM, TBRM-SD, and TBRM-FD). Significant differences in MAP were observed in FDRM-T compared to FDRM and TBRM-FD.

4.3 Query Analysis

In Table 2, we display sample terms of query expansion for the query "Keith Olbermann new job" in RM (left) and in RM-T (right). As we can see, more relevant terms such as "join", "liberal", and "controversial" appear in our approach.

Table 2: Expanded terms for "Keith Olbermann new job"

RM		RM-T	
P(w|Q)	w	P(w|Q)	w
0.116	tv	0.037	current
0.069	current	0.034	tv
0.069	home	0.020	former
0.062	file	0.017	join
0.062	rich	0.017	liber
0.062	richer	0.017	countdown
0.053	former	0.015	controversi
0.034	loser	0.015	pundit
0.034	becom	0.015	report
0.034	countdown	0.012	ap

We also display the query-wise performance comparison of the all queries that are affected (46.9%) by our approach compared to RM baseline in Figure 2. This shows that our approach improves the performance by a large margin for some topic queries; whereas it decreases the performance by a small margin for a smaller number of topic queries.

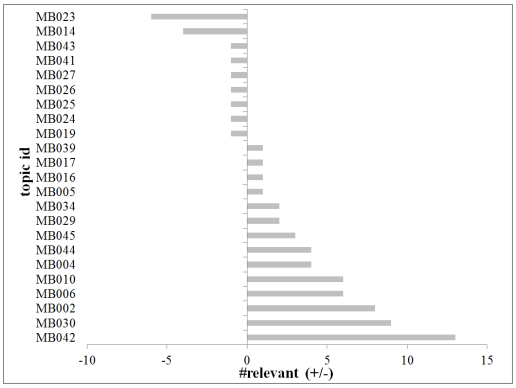

Figure 2: Query-wise performance comparison – the increase (+) / decrease (-) of the number of relevant document at 30 of RM-T compared to RM

5. CONCLUSION

In this work, we suggested a method for selecting the time period based on a user behavior (i.e., retweets) that can be collected easily. We incorporated these time periods for query expansion in a pseudo-relevance feedback setting. To that end, we extended the previous work on a time-based relevance model. More specifically, we used the difference in the temporal information from the top retrieved documents and retweets to capture time periods with many relevant documents. The experimental results based on the TREC Microblog track collection and query analysis showed that our approach for query expansion improves retrieval performance compared to the language model baselines and another approach incorporating time into a ranking criteria.

6. ACKNOWLEDGMENTS

This work was supported in part by NHN Corp. and in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors' and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] T. Brants and A. Franz. Web 1T 5-gram Version 1, 2006.

[2] W. Dakka, L. Gravano, and P. Ipeirotis. Answering general time-sensitive queries. In CIKM'08, 2008.

[3] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In SIGIR'04, 2004.

[4] M. Efron and G. Golovchinsky. Estimation Methods for Ranking Recent Information. In SIGIR'11, 2011.

[5] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In WWW'11, 2011.

[6] R. Jones and F. Diaz. Temporal profiles of queries. ACM Transactions on Information Systems, 25(3):14, 2007.

[7] M. Keikha, S. Gerani, F. Crestani, Time-based Relevance Models. In SIGIR'11, 2011.

[8] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media?. In WWW'10, 2010.

[9] V. Lavrenko, W. B. Croft. Relevance-based language models. In SIGIR'01, 2001.

[10] X. Li and W. B. Croft. Time-based language models. In CIKM'03, 2003.

[11] K. Massoudi, E. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In ECIR'11, 2011.

[12] D. Metzler, C. Cai, E. Hovy, Structured Event Retrieval over Microblog Archives, NAACL-HLT '12, 2012.

[13] D. Metzler, W. B. Croft. A Markov random field model for term dependencies. In SIGIR'05, 2005.

[14] N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Searching microblogs: Coping with sparsity and document quality. In CIKM'11, 2011.

[15] M-H. Peetz, E. Meij, M. de Rijke, W. Weerkamp, Adaptive Temporal Query Modeling, In ECIR'12, 2012.

[16] H.K. Peng, J. Zhu, D. Piao, R. Yan and Y. Zhang. Retweet Modeling Using Conditional Random Fields. ICDM Workshops'11, 2011.

[17] M. D. Smucker, J. Allan, B. Carterette, A Comparison of Statistical Significance Tests for Information Retrieval Evaluation, In CIKM'07, 2007.

[18] J. Teevan, D. Ramage, and M. Morris. #Twittersearch: A comparison of microblog search and web search. In WSDM'11, 2011.