TREC 2012 Microblog Track Experiments at Kobe University

Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara

Graduate School of System Informatics Kobe University,
1-1 Rokkodai, Nada-ku, Kobe, Hyogo, 657-8501 Japan.
{miyanishi,seki,uehara}@ai.cs.kobe-u.ac.jp

Abstract. This paper describes our approach to real-time ad hoc task processing in the TREC 2012 Microblog track. The approach uses two-stage relevance feedback to model search interests and temporal dynamics on a microblog. The first relevance feedback uses a single user-selected tweet as feedback. The second approach uses time-based query expansion method leveraging the temporal property derived from the real-time feature on microblogging services. The experimentally obtained results demonstrate that our two-stage relevance feedback approaches improve search result relevance considerably.

1 Introduction

Microblogging is a powerful online medium enabling people to understand what is happening around the world today. By retrieving microblog messages, we can quickly find interesting information related to social events [18]. Among the different microblogging platforms, Twitter [1] is a well-known service on which users post 340 million tweets (messages issued by Twitter users) per day to communicate with other users, to report on their own daily lives, and to seek/share information[2]. One interesting property of Twitter is that many tweets are posted by crowds of people when a notable event occurs. Consequently, a set of tweets constitutes an important clue about what topics are being described actively at a particular time. We leverage this temporal variation of the topic for implicit feedback similar to the past research [4, 6] after the simple tweet selection feedback in which the user manually selects a single relevant tweet from the initial search results for enriching an original query.

2 Our Approach

In our work, we propose two-stage relevance feedback methods: Tweet selection and time-based query expansion. Our tweet selection based relevance feedback presents interactive relevance feedback. Classical interactive relevance feedback

[1] https://twitter.com/
[2] http://blog.twitter.com/2012/03/twitter-turns-six.html

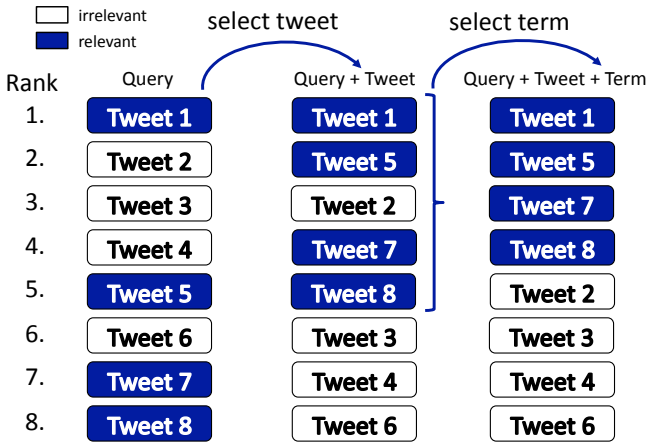

Fig. 1. Overview of two-stage relevance feedback methods.

such as the Rocchio algorithm [15] can improve IR performance. However, it requires sufficient judged documents. Moreover, the common relevance judgment is a difficult job. Query expansion method based on pseudo-relevance feedback does not require judged documents [9]. This pseudo-relevance feedback assumes that the top-ranked documents in the initial search results include good terms for feedback. However, that assumption is often invalid because only a small share of the suggested expansion terms are useful. Many others are either harmful or useless [1]. To overcome these problems, we investigate the effect of manual relevance judgment with the least effort for selecting a single relevant document and re-retrieve documents using the selected document as a new query, which often includes topic-related terms. Moreover, we apply time-based query expansion method to the re-retrieved documents, which almost all include relevant tweets at the top. Figure 1 presents an overview of our approach. The next section explains details of respective methods.

2.1 Tweet Selection as Feedback

The first relevance feedback is a tweet selection from the initial search results. We assume that the relevant tweet selected by users is a good indicator to retrieve relevant tweets to a given query because the relevant tweet generally includes good topic-related terms. Moreover, we observed that the tweets retrieved at the top using a standard search engine with default settings are almost relevant. For example, the precision at 30 of the search results obtained using Indri search engine [17] with default setting of about 0.38 on TREC 2012 Microblog track queries were 51–110: users can detect about 11 relevant tweets in search results within the top 30 on average. In addition, users can read many tweets quickly because the tweet length limit is 140 characters. Consequently, users can readily

Fig. 2. New query = original query + relevant tweet by user feedback.

detect a relevant tweet without much effort. Finally, we combine the selected tweet with the original query to create a new query for retrieving a new set of tweets. Figure 2 presents an example of the original query (*Taco Bell filling lawsuit*), its relevant tweet, and a new query.

2.2 Time-Based Query Expansion

In this section, we show the time-based implicit feedback inspired by Efron's model [6], which incorporates temporal properties such as recency and the smoothed temporal variation of a topic into a microblog search. His method used a temporal profile [8] represented as a timeline for a set of documents returned using a search engine and assumed that the density of a relevant document's temporal profile (relevant profile) has a smaller Kullback–Leibler (KL) divergence from the temporal profile for a seed query (query profile) than the non-relevant document's profile (irrelevant profile). We also use this temporal property to select topic-related terms that have similar temporal dynamics to an original query.

Temporal modeling To model the temporal properties of a candidate term combined with a seed query, we borrow Jones and Diaz's idea [8]. First, the distribution in a particular day t is defined as $P(t|Q)$, where Q is a query. This probability is defined as

$$P'(t|Q) = \sum_{D \in R} P(t|D) \frac{P(Q|D)}{\sum_{D' \in R} P(Q|D')}, \tag{1}$$

where R represents the set of top M documents returned by a search engine for Q, D denotes a document, and $P(t|D) = 1$ if the dates of t and D are the same; otherwise, $P(t|D) = 0$. Here, $P(Q|D)$ is the relevance score of a document D for Q.

To handle possible irregularity in the collection distribution over time, background smoothing is applied as

$$P(t|Q) = \lambda P'(t|Q) + (1 - \lambda) P(t|C), \tag{2}$$

where the temporal model of this collection C (collection temporal model) is defined as $P(t|C) = \frac{1}{|C|} \sum_{D \in C} P(t|D)$. Here, C is the set of all documents in a

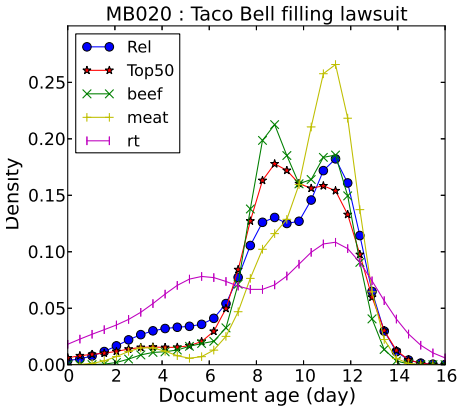

Fig. 3. Kernel density estimate obtained using topic MB020 (Taco Bell filling lawsuit). Green, yellow, and purple lines respectively show temporal profiles for *beef*, *meat* (topic-related terms), and *rt* (general term). *Top50* and *Rel* are temporal profiles created from the top 50 documents and relevant documents for the topic.

corpus. We set λ to 0.9 following previous work [8] and use this $P(t|Q)$ as the query temporal model. Although the existing method applies smoothing across adjacent days for the query temporal model, we do not do so in our microblog search settings because the daily frequency of a term is important for microblog.

Temporal variation. By measuring the difference between the query profile and the expanded query profile (temporal profile created from an expanded query), we devised a new query expansion method (TSQE) for selecting temporally related terms. This model is based on the insight derived from Figure 3, where the temporal profile created from the combination of a seed query and a related term resembles the relevant profile and conversely that the temporal profile of a non-related term is dissimilar from the relevant profile. The candidate terms are selected by the KL-divergence between two temporal models defined as

$$S_{TSQE}(w,Q) = -D_{KL}(P(t|w\cap^{+}Q), P(t|Q)) = -\sum_{t=1}^{T} P(t|w\cap^{+}Q)\log\frac{P(t|w\cap^{+}Q)}{P(t|Q)}, \tag{3}$$

where $w \cap^{+} Q$ is the expanded query that includes *both* at least one seed query term and a candidate term. We assume that a term with low KL-divergence for a seed query has the capability of retrieving relevant documents as effectively as a seed query because low KL-divergence indicates that a candidate term has been used along with at least one seed query term over time. Moreover, our model can capture the daily document frequency. For that reason, that it is applicable to any temporal variations. However, it unfortunately ignores the recency factor. The detail of this model is described in our work [13].

2.3 Additional Query Expansion

Temporal recency. To assess the recency of a microblog, we use another temporal recency-based query expansion method (TRQE), which is a modification of Efron's model [6] defined as follows.

$$S_{TRQE}(w, Q) = \phi(T_Q, T_{Q'}) = \log\big(\frac{m_{T_Q}}{m_{T_{Q'}}}\big) \tag{4}$$

Therein, m_{T_Q} and $m_{T_{Q'}}$ is the sample mean of the time stamps (average document age) obtained from the top L documents retrieved by a search engine with an original query Q and the expanded query $Q' = Q \cap w$ that includes a term w and at least one original query term. This model can suggest the candidate term related to a given query, which favors more recent documents than a seed query.

Atemporal measure. To handle a topic for which the temporal variation is difficult to predict, we use an atemporal QE method that can discover the terms related to a given topic without temporal properties. To calculate relatedness between a query and a candidate term, we use a modified Normalized Google Distance (NGD) [3] defined as

$$S_{NGD}(w, Q) = \frac{\max\{\log H(w), \log H(Q)\} - \log H(w \cap^+ Q)}{\log N - \min\{\log H(w), \log H(Q)\}}, \tag{5}$$

where $H(w)$, $H(Q)$, and $H(w \cap^+ Q)$ are the page counts of documents retrieved by the word w, the original query Q, and the expanded query $w \cap^+ Q$. We set all web similarity measures to zero if the page count $H(w \cap^+ Q)$ is less than a threshold c (set to four in our experiment). Thereby, we avoid the case in which w and Q accidentally appear on some tweets.

2.4 Learning to Rank

For integrating various features, we use the learning to rank (L2R) approach for tweet ranking, similarly to past works [5, 11, 12]. To learn a ranking function (ranker) that provides tweet ranking in relevance order, we prepared training data based on the features as described in Table 1. We use SVMrank as a ranker, an implementation of Ranking SVM [7]. No kernel was used to speed up the learning process or to reduce the number of parameters to be optimized. After learning, we performed leave-one-out cross-validation on the 49 topics (MB1-49 used in TREC 2011 Microblog track) to determine the optimum parameter C of Ranking SVM, which controls the tradeoff between empirical loss and regularization. We tested 0.001, 0.003, 0.005, 0.008, and 0.01. With the optimum parameters (C = 0.001), we reranked both search results for the 60 test queries (MB51-110 used in TREC 2012 Microblog track). As the output, we considered only the top 30 tweets after filtering and reranking.

Table 1. Features representing a tweet.

Abbr.	Feature description
LM	Query likelihood with Dirichlet smoothing ($\mu = 2500$)
TSQE	Temporal variation similarity score in Equation 3
TRQE	Temporal recency score in Equation 4
NGD	Modified normalized google distance in Equation 5
Voca	Number of vocabularies in a tweet
Hashtag	Contains a hashtag in a tweet
URL	Contains a uniform resource identifier (URL) in a tweet
Mention	Contains a mention (e.g. @trecmicroblog) in a tweet

Table 2. Number of fetched tweets.

Status code	# of tweets (%)
200 (OK)	14,230,073 (88.1)
302 (Found)	1,131,329 (7.00)
403 (Forbidden)	207,373 (1.28)
404 (Not Found)	573,034 (3.54)

3 Evaluation

3.1 Experimental Settings

We built the Tweets2011 corpus using the twitter-corpus-tools[3] based on the 16,141,812 tweet seeds provided by the track organizers. Table 2 shows the distribution of the HTTP status codes of the results. These tweets were posted by 5,356,842 distinct users in total, of whom 5,194,623 (97.0%) corresponded to the status codes 200 (OK) or 302 (Found). Then, we indexed tweets posted before the specific time associated with each topic by the Indri search engine. This index was created to simulate a realistic real-time search setting, where no future information is available when a query is issued. We built an index for each query. To retrieve tweets, we use the query likelihood model with Dirichlet smoothing [19] (we set smoothing parameter $\mu = 2500$) implemented by the Indri search engine. All queries and tweets are stemmed using the Krovetz stemmer without stop-word removal. They are case-sensitive. Moreover, we use the full dependence variant of the Markov random field (MRF) model [10] for a microblog search similar to Metzler's work [11]. In our experiments, we used the titles of TREC Microblog track topics numbered 51–110 as test queries, which are the official queries in the TREC 2012 Microblog track. We removed tweets with http status codes of 301, 302, 403, and 404 at May first, 2012 and also filtered all non-English retrieved tweets using a language detector with an infinity-gram, called ldig[4] from the final tweet ranking.

For TSQE, TRQE, and NGD, we selected candidate terms among the top 30 tweets retrieved by the original query after removing the uniform resource

[3] https://github.com/lintool/twitter-corpus-tools
[4] https://github.com/shuyo/ldig

locators (URLs), and user names starting with '@' or special characters (!, @, #, ', ", etc.). All query terms, candidate terms, and tweets were decapitalized. The candidate terms included no stop-words prepared in Indri. We removed candidate terms that did not appear more than five times along with a query term. For TSQE and TRQE, we used the temporal profile consisting of the top 10 retrieved tweets ($L = 10$) and selected 10 terms among candidate terms in descending order of S_{TSQE} and S_{TRQE} scores, respectively. NGD also selected 10 terms among candidate terms in ascending order of the S_{NGD} score. We used retweets for each query expansion method because retweets are a good source for improving twitter search performance [2]. The selected terms contained no original query term. We used the combination of the original query and the selected terms as an expanded query; they were weighted by the Indri query language [17] with 6 : 4.

For tweet selection as feedback (TSF) described in Section 2.1, one author manually selected relevant tweets from search results at the top 30 by topic, which were retrieved by the Indri search engine with default settings. All selected tweets were stopped using Indri's stop words list with URL and mention (e.g. @trecmicroblog) removal. Table 3 presents a summary of our runs, where *tsqe* and *kobeL2R*, which are close to our run in TREC 2011 Microblog track, are strong baselines. In the new query, the selected tweet and the original query were weighted by the Indri query language with 1 : 1 for *kobeMHC2*. For *kobeMHC*, we only use the selected tweet as the new query.

3.2 Evaluation Measure

The goal of our system is to return a ranked list of tweets using relevance feedback of two types. The evaluation measures that we used are precision at rank 30 (P@30) and mean average precision (MAP) with respect to allrel and highrel ("allrel" denotes a tweet judged relevant and "highrel" denotes a tweet judged as highly relevant). P@30 was the official Microblog track metric in 2011 [14]. These measures provided a succinct summary of the quality of the retrieved tweets. "highly relevant" is the required level of relevance in this year [16]. To test for statistical significance, we used a two-tailed paired t-test. The best performing run is shown in bold. Significant improvements are denoted, respectively, with † and ‡ for the significance probability $p < 0.05$ against *tsqe* and *kobeL2R*.

3.3 Experimental Results

Feature analysis. To elucidate the importance of the various features used in our ranking model, we examined the weight of each feature used by the Ranking SVM model on our training data as −0.404 (LM), 0.607 (NGD), 8.244 (TRQE), 0.650 (TSQE), 0.209 (Voca), 0.0832 (Hashtag), 0.930 (URL), −0.520 (Mention). TRQE is the most important, followed by URL, TSQE, and NGD. However, the smallest important feature is Mention with −0.520 weight. Based on these results, we found that the existence of URL in a tweet and the usage of query expansion are important for improving microblog search performance.

Table 3. Summary of our approaches. *baseline* is an unofficial run: *tsqe*, *kobeL2R*, *kobeMHC*, and *kobeMHC2* are official runs.

Run ID	Run type	Approaches
baseline	Automatic	MRF
tsqe		MRF + TSQE
kobeL2R		MRF + learning-to-rank
kobeMHC	Manual	MRF + TSQE + TSF (selected tweet)
kobeMHC2		MRF + TSQE + TSF (original query + selected tweet)

Table 4. Results of the real-time ad hoc task in the TREC 2012 Microblog track.

Criteria	*baseline*	*tsqe*	*kobeL2R*	*kobeMHC*	*kobeMHC2*
P@30 (allrel)	0.3921	0.4339	0.4429	0.4616	**0.4689**†
MAP (allrel)	0.2413	0.2843	0.2767	0.2986†	**0.3070**†‡
P@30 (highrel)	0.1983	0.2311	**0.2384**	0.2339	0.2356
MAP (highrel)	0.1695	0.2093	0.2081	0.2115	**0.2137**

Overall results. We summarized the results of our experiments in Table 4, which shows that the *tsqe* run that uses time-based query expansion performs better than *baseline* does, suggesting that temporal dynamics of a query and topic-related terms seems to play an important role for microblog search. The method *kobeL2R* outperformed other methods in P@30 on highly relevant tweets which should contain either highly informative content, or link to highly informative content [16], because L2R approach can leverage a URL feature. In addition, *kobeMHC*, which uses relevance feedback based on the tweet selection and time-based expansion without an original query, highly ranked relevant tweets, thereby displaying a striking boost in performance. Moreover, *kobeMHC2*, which uses time-based query expansion after re-ranking with a given query and a selected tweet as relevance feedback, showed some improvement compared to *kobeMHC*. The results demonstrate that relevance feedback created from manually selecting a relevant tweet dramatically improves microblog search performance in spite of its simplicity. This finding implies that microblog search leveraging minimum user interaction with search engine is a promising approach.

4 Conclusion

Through the real-time ad hoc task of TREC 2012 Microblog track, we developed a two-stage relevance feedback approach: tweet selection and time-based query expansion. In the tweet selection step, we manually select a relevant tweet from initial search results and combine the tweet with an original query to create a subsequent query. The next step re-retrieves tweets with the new query using time-based query expansion method, which leverages temporal variation of a given topic and its topic-related terms. The query expansion step further improved the search performance, achieving the best overall result after the tweet selection step. For future work, we plan to analyze the characteristics of a rel-

evant tweet that improves information retrieval performance by tweet selection and plan to refine our time-based query expansion method.

References

1. Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting good expansion terms for pseudo-relevance feedback. In: SIGIR. (2008) 243–250
2. Choi, J., Croft, W.B.: Temporal models for microblogs. In: CIKM. (2012) 2491–2494
3. Cilibrasi, R.L., Vitanyi, P.M.B.: The google similarity distance. TKDE **19**(3) (2007) 370–383
4. Dakka, W., Gravano, L., Ipeirotis, P.G.: Answering general time-sensitive queries. TKDE **24**(2) (2012) 220–235
5. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.: An empirical study on learning to rank of tweets. In: COLING. (2010) 295–303
6. Efron, M.: The university of illinois' graduate school of library and information science at TREC 2011. In: TREC. (2011)
7. Joachims, T.: Optimizing search engines using clickthrough data. In: Proc. KDD. (2002) 133–142
8. Jones, R., Diaz, F.: Temporal profiles of queries. TOIS **25**(3) (2007)
9. Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR. (2001) 120–127
10. Metzler, D., Croft, W.: A markov random field model for term dependencies. In: SIGIR. (2005) 472–479
11. Metzler, D., Cai, C.: Usc/isi at trec 2011: Microblog track. In: TREC. (2011)
12. Miyanishi, T., Okamura, N., Liu, X., Seki, K., Uehara, K.: TREC 2011 microblog track experiments at Kobe university. In: TREC. (2011)
13. Miyanishi, T., Seki, K., Uehara, K.: Combining recency and topic-dependent temporal variation for microblog search. In: ECIR. (2013)
14. Ounis, I., Macdonald, C., Lin, J., Soboroff, I.: Overview of the TREC-2011 microblog track. In: TREC. (2011)
15. Rocchio, J.J.: Relevance feedback in information retrieval. In Salton, G., ed.: The SMART Retrieval System – Experiments in Automatic Document Processing. Prentice Hall, Englewood Cliffs, NJ (1971) 313–323
16. Soboroff, I., Ounis, I., Lin, J.: Overview of the TREC-2012 microblog track. In: TREC. (2012)
17. Strohman, T., Metzler, D., Turtle, H., Croft, W.: Indri: a language model-based search engine for complex queries. In: ICIA. (2005)
18. Teevan, J., Ramage, D., Morris, M.: #TwitterSearch: a comparison of microblog search and web search. In: WSDM. (2011) 35–44
19. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. TOIS **22**(2) (2004) 179–214