QU at TREC-2013: Expansion Experiments for Microblog Ad hoc Search

Maram Hasanain, Latifa Al-Marri, and Tamer Elsayed
Computer Science and Engineering Department
Qatar University
Doha, Qatar
{maram.hasanain,latifa.almarri,telsayed}@qu.edu.qa

ABSTRACT

In the first appearance of Qatar University (QU) at Text REtrieval Conference (TREC), our submitted microblog runs explored different ways of expanding the context of both queries and tweets to overcome the sparsity and lack of context problems. Since the task is real-time, we have also considered the temporal aspect, once combined with tweet expansion technique, and another separately as a scoring factor. Our explored ideas were all unsupervised and only used internal resources (i.e., the provided API service with only access to the tweets). For query expansion, we have used pseudo relevance feedback to include terms from the top-ranked retrieved tweets. Based on experiments on previous TREC collections, an aggressive expansion with 30 terms or more provided the best improvement. For tweet expansion, a 2-step relevance modeling approach was leveraged to temporally and lexically expand a tweet. To further explore the effect of considering the time dimension in scoring tweets, we also developed a temporal re-scoring function used to favor tweets that are closer in time to the query over tweets that might be more lexically relevant but are posted further apart in time from the query. We also conducted post-TREC experiments in which we worked on enhancing the query expansion and temporal re-scoring approaches. Resuls released by TREC have shown that the temporal re-scoring run was the most effective run among all of our submitted ones. As for the post-TREC experiments, our results have shown that the enhanced query expansion and temporal re-scoring approaches had notable improvements on retrieval effectiveness.

1. INTRODUCTION

Microblogging online services have been very popular and widely used in the recent few years. Twitter in particular is one of the most rapidly growing microblogging platforms that is used to share information, communicate with friends, and follow up on ongoing events. Every day, millions of users are posting millions of posts (called "tweets") that can be viewed as "the heart beat of the world" at the moment since it fairly represents news, events, actions, reflections, comments, ideas, conversations, and more from all over the world, all in real-time.

As searching the Web became a daily habit by many users in the last decade, searching Twitter is becoming more needed recently to get updated on real-time incidents or topics over tweets that are posted every second. However, searching tweets is far different from searching the Web due to the different nature of the sought content. One of the most notable and distinguishing features of Twitter is the limited allowed length of tweets that may not exceed 140 characters. This limitation forces the users to post tweets that might lack context when viewed in isolation. Twitter users have partially overcome this problem by using special sybmols in their tweets such as '@' (to mention or reply to other users), 'RT' (to quote or re-post/re-tweet other tweets), and '#' (to tag the tweet by a following label, called a hashtag, that generally indicates the topic of the tweet). Hyperlinks to web pages have also been embeded (in a shortened form) to extend relations to the external Web content. Although these user-generated features indirectly enrich the content of the tweet beyond the exact words appearing in the text, it is tricky sometimes for an automated system to correctly interpret and thus recounstruct. The informal and conversational nature of the tweets make it even harder to process, and the temporal aspect adds another dimension that is hardly considered in Web search. All of these special characterisitcs make the task of searching in microblogs an interesting but challenging research problem.

Text REtrieval Conference (TREC) has recently introduced a microblog track for the first time in 2011. The main task in the track (which continues for the third time this year) is concerned with real-time ad-hoc search that aims to retrieve timely relevant tweets given a free-text query issued at a given time. The primary difference this year lies in the newly crawled tweets collection and the way that participants interact with it. This year, we, at Qatar University (QU), have participated in TREC microblog track for the first time, with two basic objectives. The first is to form a basis for an IR research team at QU, and the second is to build a strong baseline for that task that can be compared with other teams and easily extended. In our first appearence at TREC, we were interested in exploring ideas eveloving around context expansion of both queries and tweets to tackle the problem of vocabulary mismatch between both. For query expansion, we tried the classical blind relevance feeback to add new topically-similar terms

TREC '13 Gaithersburg, Maryland USA

to the query. For tweet expansion, we used relevance modelling based approach to expand tweets by topically and temporally similar tweets. Finally, we experimented with one way of temporally scoring tweets to favor the ones that are temporally-close to the query.

The rest of the paper is organized as follows. Section 2 introduces our proposed expansion approach in the different dimentions we explored. Section 3 discusses some issues in implementing our system that are related to the new track-as-a-service design of the ad hoc search task this year. Section 4 presents our experimenatal evaluation results. Section 5 concludes the paper and outlines some of the future directions we are interested in exploring.

2. APPROACH

In this work, we adopted a general strategy that tries to tackle the vocabulary mismatch problem by expanding the context of both queries and tweets, either topically, temporally, or both. In this section, we discuss the adopted approach in detail.

2.1 Query Expansion

Queries usually consist of few terms which can barely describe the user information need. Such limitation can negatively affect the effectiveness of a retrieval system. The problem in an ad-hoc search system over Twitter might be more prevalent as tweets are also very short which increases the possibility of terms mismatch between a query and a tweet. To overcome such limitation, query expansion is usually applied to enrich a query Q with a set of terms. In our system, we utilized a pseudo relevance feedback [5] approach to expand a query Q using m terms extracted from the top k assumed-relevant tweets to Q. Expansion is applied as follows:

1. Q is issued to the retrieval system to retrieve an initial ranked list, R_0.
2. All terms that appear in the top k tweets of R_0 are scored using a scoring function.
3. Top m terms (called expansion terms) are selected from the scored terms and appended to Q to produce an expanded query.
4. The expanded query is eventually used to retrieve the final ranked list of tweets.

Terms are scored using a variant of $tf - idf$ scoring function [5] that replaces tf with the number of documents in R_0 in which the term appeared. Such scoring was used to favor expanding a query using terms that frequently appeared in top retrieved tweets. Such terms are thus viewed to be the most relevant to the original query.

2.2 Document Expansion

Lexical Expansion: Since tweets are very short, words posted in tweets often lack context. Moreover, the small number of terms that appear in a tweet increases the risk of terms mismatch between query and tweets. Thus, enrichment of tweets via document expansion techniques can possibly enhance retrieval effectiveness. We have explored aggressive expansion of tweets lexically and temporally as proposed by Efron et al. [1]. The approach relies on language modeling where tweets are represented by language models over terms in the vocabulary. The relevance between a query Q and a tweet D can be estimated by the likelihood that the language model of D has generated Q.

$$P(Q|D) = \prod_{q=1}^{|Q|} P(w_q|D) \tag{1}$$

Using multinominal language models, $P(w|D)$ can be estimated using the maximum-likelihood estimator $P_{ml} = \frac{f_{w,D}}{|D|}$. Dirichlet smoothing [5] is also applied to enhance the model of the document. As a result, the language model of the tweet can be estimated as follows:

$$P(w|D) = \frac{|D|}{|D| + \mu} P_{ml}(w|D) + \frac{\mu}{|D| + \mu} P(w|C) \tag{2}$$

where C denotes the tweets collection composed of N documents and $|D|$ represents the length of the tweet (in terms). $P(w|C)$ is the relative collection frequency of w. Expansion is then applied to enhance the language model of the tweet. To achieve that, a tweet D is issued as a pseudo-query Q_D, consisting of $|D|$ terms d_1,...,$d_{|D|}$, to a retrieval system. The retrieved ranked list of tweets, denoted as R_D, can be used to induce a lexically expanded tweet D'. The language model of D' is computed as follows:

$$P(w|D') = \frac{P(w, d_1, ..., d_{|D|})}{P(d_1, ..., d_{|D|})} \tag{3}$$

The denominator $P(d_1, ..., d_{|D|})$ does not depend on w, which makes the numerator $P(w, d_1, ..., d_{|D|})$ the quantity of interest in this equation. As shown by Efron et al. [1], $P(w, d_1, ..., d_{|D|})$ can be estimated as follows:

$$P(w, d_1, ..., d_{|D|}) = \sum_{D_i \in C} P(D_i) P(w|D_i) \prod_{j=1}^{|D|} P(d_j|D_i) \tag{4}$$

In this context, we consider estimating the model of D' over the tweets set R_D only rather than considering all the documents in the collection C. $P(w|D_i)$ can be estimated using the maximum likelihood estimation. $\prod_{j=1}^{|D|} P(d_j|D_i)$ can be viewed as the likelihood that the language model of tweet D_i has generated D. Usually, smoothing is performed to provide an estimate of a document based on its original form and expanded form. Thus, smoothing is applied in this case to generate the final tweet D' model as follows:

$$P_\lambda(w|D') = (1 - \lambda) P_{ml}(w|D) + \lambda P(w|D') \tag{5}$$

Following the lexical expansion of D, a lexical relevance score of tweet D to query Q can be estimated as follows:

$$P(Q|D) = \prod_{i=1}^{|Q|} \frac{|D|}{|D| + \mu} P_\lambda(w_i|D') + \frac{\mu}{|D| + \mu} P(w_i|C) \tag{6}$$

Temporal Expansion: temporal expansion was also preformed on tweet D. Similar to lexical expansion, D can be viewed as a pseudo-query for which we retrieve a set of tweets R_D to enrich the representation of D. In the context of temporal expansion, tweets in R_D are used to construct a *temporal profile* [2] of D. Assuming that any tweet D_i is associated with a timestamp t_i denoting the time at which D_i was posted, the temporal profile of D, $P(t|D)$, is defined as a probability distribution over time which is used to describe how relevant D is to events happening at different

points in time.
To estimate $P(t|D)$, Efron et al. [1] proposed the following estimate:

$$\hat{P}(t|D) = \sum_{t_i \in R_D} s_i.r.e^{-r.|t_i - t|} \tag{7}$$

where r is an exponential rate parameter that controls the temporal influence on this estimate, and s_i represents the likelihood of generating D_i given D. s_i can be estimated as follows:

$$s_i = P(D_i|D) = \frac{P(D|D_i)P(D_i)}{\sum_{j \in R_D} P(D|D_j)P(D_j)} \tag{8}$$

Using equation 7, tweet D can be temporally scored by having a temporal profile for both the query and the tweet D using the following equation:

$$P(T_Q|T_D) = \prod_{i=1}^{k_Q} P(t_{Qi}|D) \tag{9}$$

where k_Q represents the number of timestamps in Q's profile. The query temporal profile can be constructed based on the timestamps of the top k_Q tweets initially retrieved by Q. The time in this equation is in days.
Lexical and Temporal Scoring: to benefit from both the temporal and lexical evidence in a tweet D to estimate its relevancy to a query Q, scores estimated in equations 6 and 9 can be combined to generate the following scoring function:

$$P(Q, D, T_Q, T_D) = P(Q|D).P(T_Q|T_D) \tag{10}$$

Where $P(Q|D)$ is estimated using equation 6 and $P(T_Q|T_D)$ is estimated by equation 9.

2.3 Temporal Re-scoring

To further explore the effect of considering the time dimension in scoring tweets, we have also developed a temporal re-scoring function used to favor the tweets that are closer in time to the query over tweets that might be more lexically relevant but are posted further apart in time from the query. The motivation behind considering this approach is based on an assumption that tweets are usually posted on Twitter in bursts related to certain events. Capturing the recency factor when scoring a tweet can possibly enhance the effectiveness of ad-hoc search. In our temporal re-scoring approach we have again considered the temporal factor discussed in [1]. Given a query Q and a retrieved tweet D, we can construct a temporal profile for both the query and the tweet. The simplest way to view these temporal profiles is to consider a temporal profile to be represented by a single timestamp. Here we select the tweet posting time, t_D, to construct the temporal profile of D, and the query time t_Q to construct the profile of Q. To estimate the similarity between D and Q while capturing the time difference between D and Q, we have used the following estimate to temporally re-score D:

$$Score^*_{Q,D} = Score_{Q,D}.r.e^{-r.|t_D - t_Q|} \tag{11}$$

$Score_{Q,D}$ represents the original score given to D by the retrieval system. In this work, retrieval was based on a query-likelihood model with Dirichlet smoothing. r is a parameter that controls the temporal influence on the original score. The difference in time in this equation is represented in fractions of day.

3. IMPLEMENTATION ISSUES

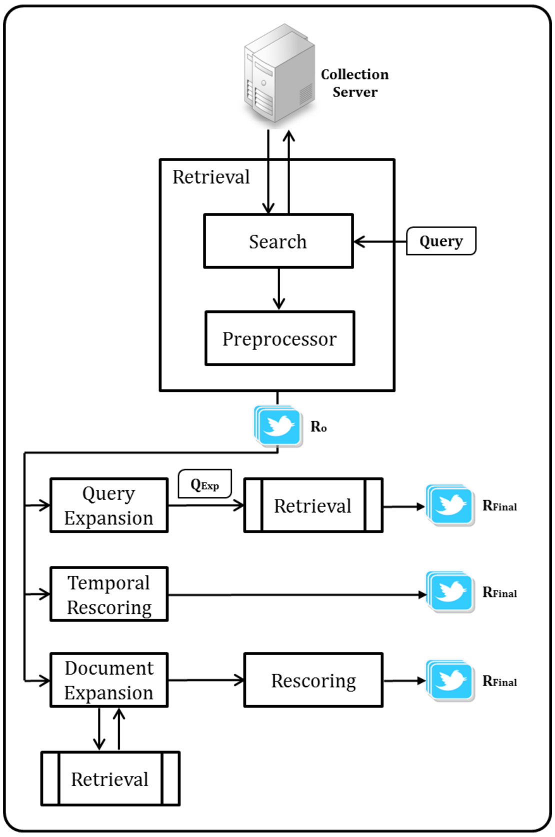

Figure 1: An overview of the ad-hoc search system

In this section, we discuss some issues related to the implementation of our ad-hoc search system. We particularly focus on the impact of the new "Track-as-a-Service" [3] design of the microblog track on our implementation. An overview of the ad-hoc search system we have developed is depicted in Figure 1. The system consisted of four main modules:

- Retrieval: Used to establish communication with the search API (that is discussed in the following section), retrieves a raw ranked list of tweets, and filters out retweets and non-English tweets (through a preprocessor) to finally produce an initial ranked list R_0 of retrieved tweets to a given query.

- Query Expansion: Expands the query as discussed earlier and then issues the expanded query, Q_{Exp}, to the retrieval module to retrieve the final ranked list of tweets R_{Final}.

- Temporal Re-scoring: Implements the approach discussed in section 2.3 to produce R_{Final}.

- Document Expansion and Re-scoring: Composed of two main stages: at the document expansion stage, the tweets of R_0 are lexically and temporally expanded as discussed in section 2.2 and then re-scored using equation 10. The initial ranked list R_0 is then sorted based on the updated scores of tweets to produce R_{Final}.

3.1 Collection API and Access

In TREC-2011 and TREC-2012, participants in the microblog track were provided with tools and unique identifiers of tweets allowing them to crawl about 16 million tweets, spanning over 16 days, through Twitter public stream. However, this approach had prohibited increasing the collection size while adhering to Twitter's terms of service. In addition, allowing participants to crawl their own copies of the collection resulted in a lack of consistency among copies acquired by different teams. To overcome these limitations, TREC-2013 organizers were motivated to provide participants in the microblog track with a centralized, remotely-stored large collection of tweets. Details on the collection can be found in section 4. Participants in the track were also provided with collection-level statistics in addition to a search API through which they can interact with the collection. The API offers three main services [4]:

- Retrieval based on query-likelihood model with Dirichlet smoothing [5]
- Access to text and metadata of retrieved tweets
- Access to both Tweets2011 and Tweets2013 collections hosted on two separate servers

In order to retrieve a ranked list of tweets given a query Q, the search API should be called with a set of parameters: query text, number of results to return, and the query time. Each tweet in the ranked list is represented as a structure with the following major attributes [3]:

- Text: the text of the tweet
- Tweet ID: the unique identifier given to the tweet by Twitter
- Metadata of the tweet such as: the language of the tweet and the number of times it has been retweeted
- Some statistics about the tweet's author such as the author's followers count and number of tweets posted by this author, etc.

3.2 Preprocessing

The tweets retrieved through the search API are not filtered to follow TREC-Microblog track rules of relevant tweets. There are two main policies that were highlighted in the TREC guidelines of the track regarding relevant tweets:

- non-English tweets are considered non-relevant: We needed to filter out non-English tweets. We used a language detection tool developed by Cybozu Labs [1]. The tool is a java library that uses Bayesian filtering to detect a text language. As reported, it gives 99% precision for 53 languages.
- Retweets are not relevant: We filtered out the tweets that start with "RT" since those are considered pure retweets. According to the track policy, partial retweets would be judged based on the non-retweeted text.

3.3 Design Issues

The new design of the track has imposed some constraints that we have considered in the design and implementation of our system. Due to the fact that the system should process tweets remotely indexed and stored and can only be acquired through the search API in a restricted rate of retrieval, it was designed to completely function at query time, even for methods that naturally do most of the computations at indexing time. Such behavior is evident in the document expansion approach adopted in our system. As discussed in section 2.2, given a query Q, each relevant tweet to Q can be expanded by representing it as a pseudo-query to the retrieval system to retrieve the expansion tweets, which is a process that should be naturally performed at indexing time. Having the search API as the sole service, such expansion behavior was performed at query time by sequentially expanding initially-retrieved tweets. This work-around resulted in a significant time overhead in this approach.

4. EXPERIMENTAL EVALUATION

In this section, we discuss our experimental setup followed by the evaluation results of our official runs. We also discuss some of our post-TREC experiments and findings.

4.1 Experimental Setup

TREC-2013 microblog track provided participants with a common API to access and retrieve tweets from a collection of approximately 240 million tweets. The API allows users to submit a query and provides back a list of tweets from the tweets collection (Tweets13). A list of 60 new topics were also provided for evaluation. In this section we report our system evaluation over this collection. For training, we have also utilized the 2011 tweets collection (Tweets11, which is also accessible via the search API) associated with the 2011 and 2012 ad hoc search topics developed by TREC for the microblog track in 2011 and 2012 respectively.

As discussed in the Microblog track overview papers [6, 7], the primary evaluation measures used to evaluate ad hoc search in Twitter are precision at 30 (P@30), mean average precision (MAP), and R-precision (R-Prec).

Baselines: Table 1 below briefly describes the three baselines we used in our experiments. Preceding TREC runs submission, we measured the effectiveness of our system in comparison to two baselines: *Baseline11* and *Baseline12*. Following the release of the relevance judgments of this year's track, we carried out post-submission experiments on the 2013 collection with an additional baseline, *Baseline13*. In all of the baselines, the retrieval model was the one underlying the search API, i.e., query likelihood model with Dirichlet priors. Following the track guidelines, retweets removal (denoted by **RTR**) was applied in all of the three baselines. Non-English tweets were also removed using the language detection tool described in section 3.2. We refer to the process of non-English tweets removal by **LD**.

Baseline	Collection	Topics	LD?	RTR?
Baseline11	Tweets11	2011	✓	✓
Baseline12	Tweets11	2012	✓	✓
Baseline13	Tweets13	2013	✓	✓

Table 1: Baselines we used in our experiments

[1] https://code.google.com/p/language-detection/

4.2 TREC Official Runs

The table below summarizes our official runs submitted to TREC-2013 microblog track. It indicates for each run, which approach was followed and whether or not an external evidence [2] is used. In our case, we used one external resource, which is the third-party language detection tool.

Run ID	Approach	LD?	RTR?
QUBaseline	Query Expansion	✗	✓
QUDocExp	Document Expansion	✓	✓
QUQueryExp	Query Expansion	✓	✓
QUTemporal	Temporal Rescoring	✓	✓

Table 2: Submitted runs to TREC-2013

Our four runs submitted to TREC are discussed below with more details.

- **QUBaseline:** The run utilizes query expansion, discussed in section 2.1, to expand each query Q and provides search results based on the expanded Q. Additionally, retweets removal is applied to eliminate retweets from the results of this run.
- **QUDocExp:** The run utilizes document expansion, explained in section 2.2, to expand each tweet in the list of tweets retrieved given a query Q and re-score the expanded tweets. Additionally, retweets removal is applied and a language detection tool is used to eliminate non-English tweets.
- **QUQueryExp:** This is the same as the QUBaseline run but with non-English tweets eliminated.
- **QUTemporal:** This run follows the approach discussed in section 2.3. For each query, the tweets retrieved through the search API are temporally re-scored and the ranked list of results is sorted based on this temporal score. Retweets and non-English tweets removal is also applied.

Parameter Tuning: In the three main approaches we have utilized to develop our system, we had several parameters that we needed to tune. Table 3 presents these parameters categorized based on the approach in which they were used. It also indcates the final tuned value set for each parameter. We followed two main strategies to set or tune the parameters. In the first strategy, we used parameters values reported in the studies from which we adopted some of the approaches. Under this strategy, parameters: λ, μ, and r used in document expansion were adopted as reported in [1]. As for the second strategy, we ran experiments for each of the approaches using 2011 and 2012 topics. In these experiments, we focused on selecting parameters values that maximize P@30 for each approach. Parameters tuned by this strategy are: k in both query expansion and document expansion approaches, m, k_Q, k_t, and r used in temporal re-scoring. Table 4 shows the P@30 values we got using the corresponding tuned parameters. The baselines in the table are Baseline11 and Baseline12 described in section 4.1. For setting parameters for the query expansion approach, we focused on QUQueryExp run rather than QUBaseline run that did not include non-English tweets removal. Having maximized P@30 values using 2011 and 2012 topics ran on Tweets11, we also worked on comparing these results to the corresponding baselines to gain some insights on the effectiveness of the three approaches. In all runs, the significance of difference between P@30 for the each approach and the cprrespnding baseline is measured using a one-tailed paired t-test with p= 0.05. As the table shows, temporal re-scoring had a significant improvement on P@30 in the case of 2011 topics, and had a negligible improvement in 2012 topics. For the query expansion approach, P@30 was significantly improved in the case of 2012 topics. The improvement was not significant in the 2011 topics, but measuring the difference between the 2011 baseline and P@30 resulted in a probability of 0.056 which is very close to the significance level 0.05.

Run ID	Parameters	Values
QUBaseline QUQueryExp	*k:* Number of expansion tweets	k= 40
	m: Number of expansion terms	m= 20
QUDocExp	*k:* Number of expansion tweets	k= 5
	λ & μ*:* Factors used in smoothing in the lexical expansion	λ= 0.5, μ= 2500
	k_Q*:* Number of timestamps in query profile	k_Q= 5
	k_t*:* Number of timestamps in tweet profile	k_t= 5
	r: Factor to control temporal information influence on expansion	r= 0.01
QUTemporal	*r:* Factor to control temporal influence on re-scoring a tweet	r=0.0003

Table 3: Parameters used in our four official runs

Approach	Topics11	Topics12
Baseline	0.4259	0.3537
Temporal Re-scoring	**0.4422***	0.3559
Query Expansion	0.4605	**0.4085****
Document Expansion	0.4061	0.3537

Table 4: Maximized avg. P@30 for each approach given the final parameters setting. Values marked with a * symbol are significantly different from the corresponding baseline (*$p<0.05$*), and a ** symbol indicates a *$p<0.01$*

Referring back to table 4, it should be noted that P@30 values reported for the document expansion runs of 2011 and 2012 topics were produced by running the experiments to produce 250 final tweets for each query rather than 1000 tweets as in the case of the 2013 official run. To maintain a fair comparison between these special document expansion runs and their corresponding baselines, we have also ran the system to produce two baselines for 2011 and 2012 topics with only 250 tweets per query produced in each of the two baselines. The 2011 baseline in this case achieved an P@30 of 0.4238 and the 2012 baseline has a P@30 of 0.3542.

[2] That is any evidence obtained from sources other than the official API

We eventually utilized the tuned parameter values presented in Table 3 to produce the official TREC runs. The evaluation of these runs is discussed in the following section.

4.3 TREC Results

Table 5 shows the official results for all of our official submitted runs. It also includes Baseline13 and the average of medians (AvgOfMedians) across all queries over all submitted automatic runs that was provided by TREC for comparison purposes.

Run	MAP	P@30	R-Prec
Baseline13	0.2724	0.4722	0.3194
QUBaseline	0.2555	0.4294	0.2936
QUDocExp	0.2311	0.4478	0.2809
QUQueryExp	0.2710	0.4433	0.3128
QUTemporal	**0.2748**	**0.4739**	**0.3245**
AvgOfMedians	0.2126	0.4217	0.2721

Table 5: Average MAP, P@30, and R-Prec of each run including Baseline13 and the avg. of medians over auto runs

As Table 5 shows, our QUTemporal run exhibited better average results compared to the remaining runs we submitted. The temporal re-scoring run also had a higher average score in all of the given evaluation measures compared to the average of medians across all queries. The run had a minimal improvement over the Baseline13 in both P@30 and MAP, and a significant improvement in R-Prec. QUQueryExp and QUDocExp had better P@30 scores compared to the average of medians, however it produced worse results than Baseline13.

Following results announcements by TREC, we conducted further experiments using our different approaches. Some of these experiments are discussed in the following section.

4.4 Post-TREC Results

In post-TREC experiments, we worked on enhancing the query expansion and temporal re-scoring approaches.

Query Expansion: The microblog track organizers provided participants with the terms statistics for Tweets13 collection. The terms appearing in the statistics were not stemmed and thus we have used the unstemmed terms when we selected the expansion terms in our official TREC runs. Post-TREC, we changed our implementation to use stemmed terms statistics to compute the weights of the expansion terms before selecting the top m terms to expand the query with. Following this approach, the results in all evaluation measures were generally improved. The optimal values of the two query expansion paramaters differed from the ones used in the official QUQueryExp run. Those optimal values are reported in Table 6. The enhancement introduced to the query expansion approach also resulted in an improved P@30 compared to Baseline13 and the improvement was found statistically significant. We also noticed an interesting observation in query expansion for 2013 topics; results with a low number of expansion tweets were the best, while increasing the number of expansion tweets resulted in a decrease in P@30 as represented in Figure 2. That was in contrary to the results we got using query expansion over 2011 and 2012 topics.

Temporal Re-scoring: Since the parameter r was set based on our experiments on 2011 and 2012 topics, we experimented with different values of r on 2013 topics. Table 6 also shows the value of r resulting in a maximized P@30 for the temporal re-scoring approach. It is clear from the table that the temporal run is better (non-statistically significant though) than Baseline13 but yet it had a worse P@30 compared to the new QueryExp run.

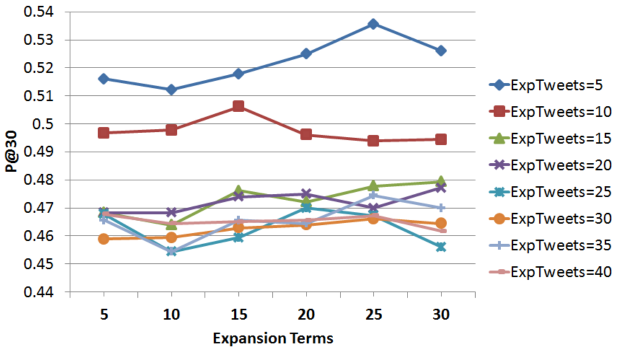

Figure 2: The average P@30 values for different combinations of expansion tweets and expansion terms in post-TREC experiments

Run	P@30	Parameters Values
Baseline13	0.4722	-
QueryExp	**0.5356***	m= 25 expansion terms
		k= 5 expansion tweets
Temporal	0.4867	r= 0.001

Table 6: Post-TREC runs compared to Baseline13. Values marked with a * symbol are significantly different from the baseline (p<*0.05*)

It should be noted that P@30 values reported in the table are based on the parameters' values set to maximize P@30.

5. CONCLUSION AND FUTURE WORK

In our approach to develop a microblog ad hoc search system, we focused on expansion methods to enrich the representation of both the query and the tweets. We also experimented with the time information available along with the tweets to temporally expand them. The time dimension of tweets was further explored in developing a temporal re-scoring model that re-ranks retrieved tweets considering the posting time of tweets relative to the query. The run based on the temporal re-scoring approach was the most effective run among our four submitted TREC official runs, which motivates us to further explore methods that consider temporal information in ranking tweets. More investigation is needed to explain the poor performance of the document expansion approach. Our experiments following TREC helped us enhance both the query expansion and temporal re-scoring methods resulting in an improved retrieval effectiveness.

An obvious extension to our work is to combine two or more of the approaches we used for ranking tweets. Another idea to explore is to develop a selective temporal scoring method in which temporally-informed scoring is only applied for event-based queries.

6. ACKNOWLEDGMENTS

This work has been supported in part by College of Engineering at Qatar University under grants QUSG-CENG-CSE-12/13-1 and QUST-CENG-SPR-12/13-7. Any opinions, findings, conclusions, or recommendations expressed are the authors' and do not necessarily reflect those of the sponsors.

7. REFERENCES

[1] M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 911–920, New York, NY, USA, 2012. ACM.

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

[3] J. Lin and M. Efron. Trec 2013 api specifications. `https://github.com/lintool/twitter-tools/wiki/TREC-2013-API-Specifications`, 2013. [Online; accessed 27-September-2013].

[4] J. Lin and M. Efron. Trec 2013 track guidelines. `https://github.com/lintool/twitter-tools/wiki/TREC-2013-Track-Guidelines`, 2013. [Online; accessed 27-September-2013].

[5] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, Cambridge, United Kingdom, 2008.

[6] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. *Proceedings of the Twentieth Text REtrieval Conference (TREC 2011)*, (May):1–4, 2011.

[7] I. Soboroff, I. Ounis, C. Macdonald, and J. Lin. Overview of the TREC-2012 Microblog Track. *Proceedings of the Twenty-First Text REtrieval Conference (TREC 2012)*, 2012.