

QU at TREC-2014: Online Clustering with Temporal and Topical Expansion for Tweet Timeline Generation

Maram Hasanain and Tamer Elsayed

Computer Science and Engineering Department
Qatar University
Doha, Qatar
{maram.hasanain,telsayed}@qu.edu.qa

ABSTRACT

In this work, we present our participation in the microblog track in TREC-2014, building upon our first participation last year. We present our approaches for the two tasks of this year: temporally-anchored ad-hoc search and tweet timeline generation. For the ad-hoc search task, we used topical expansion in addition to temporal models to perform retrieval. Our results show that our run based on the typical pseudo relevance feedback query expansion outperformed all of our other runs with a relatively high mean average precision (MAP). As for the timeline generation task, we approached this problem using online incremental clustering of tweets retrieved for a given query. Our approach allows the dynamic creation of “semantic” clusters while providing a framework for detecting redundant tweets and selecting representative ones to be added to the final timeline. The results demonstrate that using incremental clustering of tweets retrieved through a *temporal* retrieval model produced the best effectiveness among the submitted runs.

1. INTRODUCTION

Microblogging services such as Twitter are attracting users looking to engage in vibrant and influential hubs for information sharing and finding. With hundreds of millions of tweets posted daily, a large number of queries are issued seeking information. Recent studies on Twitter data have emphasized the high temporality of information published through Twitter, mostly covering breaking news and events [8, 23]. Such temporality of the data is also reflected in searching behavior over tweets [23], making it essential for a microblog search system to consider such characteristic of the data and the task. In addition, the very short length of queries (e.g., average of 3.76 words in this year’s microblog ad-hoc search task at TREC-2014) and tweets (with 140-character of maximum length) makes searching for tweets a challenging task.

Due to these factors, a microblog search system should consider temporal signals in tweets and queries in addition to augmenting their context to improve retrieval. In this work, we aim at studying the effectiveness of retrieval given these two main factors: temporality and context. We specifically study ad-hoc search given three types of retrieval models: (1) a purely temporal model, (2) a query expansion model,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
TREC’14 Gaithersburg, Maryland USA

and (3) a model that combines both temporal and query expansion factors to perform search.

Given the huge number of tweets that can be retrieved using a query, presenting a long list of tweets to a user on a given information need might not be plausible anymore; the amount of tweets the user has to go through about a topic can be overwhelming [20]. Minimizing tweet redundancy and irrelevancy can help provide a user with more informative and compact list of tweets on a topic of interest. Continuous clustering algorithms are among the most commonly-used methods to bring summarized tweet timelines to a user [15, 20]. In such approaches, online clustering, usually supported by near-duplicate detection, is used to extract representative tweets of a large stream of tweets on an ongoing topic. We employ these ideas to design a tweet timeline generation system that accepts a temporally-anchored query and provides the user with a timeline of non-redundant, chronologically-ordered tweets posted before the query time.

The remainder of this paper is organized as follows. We discuss our approach to the temporally-anchored ad-hoc search task in addition to the evaluation results in Section 2. Section 3 describes how we tackled the problem of tweet timeline generation (TTG) along with the evaluation results. We conclude this paper with Section 4.

2. AD-HOC SEARCH

The temporally-anchored ad-hoc search task is one of the microblog track tasks at TREC that continued since 2011 [18, 21, 11]. Given a free-text query issued at a given time, this task aims to retrieve timely relevant tweets for that query. To perform this task, we leverage retrieval models based on two main intuitions. First, due to the temporality of the task and the data, temporal retrieval models might be effective in this task as demonstrated in previous studies [3, 5, 12, 4]. Second, the very short length of tweets and queries can impede effective retrieval which motivated utilizing context expansion methods in microblog ad-hoc search [2, 5, 24, 16]. In total, we work with three retrieval models described next.

2.1 Retrieval Models

2.1.1 Query Likelihood (QL)

All of the models we use in this work benefit from the Query Likelihood (QL) model [19] in retrieval. This model ranks documents by the likelihood that their language mod-

els generated the query as follows:

$$P(D|Q) \propto P(Q|D)P(D) \quad (1)$$

where D is a document and Q is the query. Assuming a uniform document prior $P(D)$ and terms independence, documents can be ranked by

$$P(D|Q) \propto P(Q|D) = \prod_{w \in Q} P(w|D) \quad (2)$$

more specifically, we use the log-likelihoods to rank documents by

$$\sum_{w \in Q} \log P(w|D) \quad (3)$$

where $P(w|D)$ is computed using maximum likelihood estimate (MLE) with Dirichlet smoothing [25] as follows:

$$P(w|D) = \frac{tf_{w,D} + \mu P(w|C)}{|D| + \mu} \quad (4)$$

where $tf_{w,D}$ is the term frequency of w in D , $P(w|C)$ is estimated using MLE over the collection C and the smoothing factor μ is a free parameter for this retrieval model.

2.1.2 Time-based Exponential Priors (*t-EXP*)

The *t-EXP* model [10] is a temporal variation of the QL model in which an exponential decay factor is used as a document prior as follows:

$$P(D|Q) \propto P(Q|D) \cdot r \cdot e^{-r \cdot t_d} \quad (5)$$

where r is a decay rate factor, and t_d is the posting time difference in *days* between D and Q . As with the QL model, we rank documents using log-likelihoods by

$$\sum_{w \in Q} \log P(w|D) + \log (r \cdot e^{-r \cdot t_d}) \quad (6)$$

2.1.3 Time-based Query Relevance Modeling (*t-QRM*)

t-QRM [7] is a variant of the typical query relevance modeling approach [9] that uses a temporal query relevance model computed as follows:

$$P(w|Q) = \sum_{t \in T} P(w|t, Q)P(t|Q) \quad (7)$$

where t is a timestamp in unit of days and T is the set of timestamps in the collection. Given an initially-retrieved list R_k retrieved using the QL model, we estimate $P(t|Q)$ as the normalized sum of retrieval scores of documents posted within t . The probability $P(w|t, Q)$ can be computed as follows:

$$P(w|t, Q) = \sum_{D \in t} P(w|D)P(D|t, Q) \quad (8)$$

$P(D|t, Q)$ is assumed to be uniform over all documents in R_k posted within t . $P(w|D)$ is computed using the MLE. Once $P(w|Q)$ is computed for all terms in R_k , we expand the query with the m terms with the highest probability. Given the expanded query, the final results are retrieved using the typical QL model. Both the initial list size k and the number of expansion terms m are free parameters for this model.

2.1.4 PRF-based Query Expansion (*QE*)

Earlier work on microblog search showed that query expansion with Pseudo Relevance Feedback (PRF) [9] has good effectiveness in this task [14, 2, 17]. In typical PRF-based retrieval, a query is expanded using the m top-scoring terms extracted from an initially retrieved list R_k given Q . In this work, we used a *tf-idf* [13] like scoring function to score terms over all documents in R_k as follows:

$$Score(w, R_k) = tf_{w, R_k} \cdot idf(w) \quad (9)$$

tf_{w, R_k} is the term frequency of w in R_k and $idf(w)$ is the inverse document frequency of w computed as $idf(w) = \log\left(\frac{N}{df_w}\right)$. Once we expand the query with m terms, we use the QL model to retrieve the final list of documents using the expanded query. Both m and k are free parameters for this model.

2.2 Evaluation Setup

Similar to last year, the track used the Tweets13 collection of approximately 243 million tweets [11]. Participants can access and retrieve tweets from this collection by submitting a query to the track-provided API¹ [11]. Given the query, the API returns a list of tweets using the QL retrieval model from the Tweets13 collection. Participants can then use their own retrieval model to process this list and produce a final one.

Evaluation of the 2014 ad-hoc search task is performed given a list of 55 new topics released with Tweets13. We submitted four official runs based on three retrieval models (discussed in Section 2.1): PRF-based query expansion, *t-EXP* and *t-QRM*. We tuned the parameters of these retrieval models using 60 topics released with the microblog track in TREC-2013 with the Tweets13 collection [11]. We also removed retweets and non-English tweets from result lists; language of a tweet is detected using an open-source language detection tool². We evaluated retrieval using precision at rank 30 (P@30) and mean average precision (MAP) that were the primary evaluation measures used in previous runs of this task [18, 21, 11].

2.3 Experimental Results

We present each of our four officially-submitted runs in Table 1 below. Whenever the QL model is used, we set $\mu = 1000$.³ We present results on retrieval effectiveness of

Table 1: Description of our ad-hoc search official runs

Run	Model	Parameters
QUQueryExp5D25T	PRF-QE	$k = 5, m = 25$
QUtmpDecay	<i>t-EXP</i>	$r = 0.05$
QUQueryExp10D15T	PRF-QE	$k = 10, m = 15$
QUTQRM	<i>t-QRM</i>	$k = 25, m = 5$

these runs in Table 2. We also compare the performance of our official runs to two baselines:

¹<https://github.com/lintool/twitter-tools/wiki/TREC-2013-API-Specifications>

²<https://code.google.com/p/language-detection/>

³We tried different values for this parameter and found that this value produces best results over Tweets13.

- Baseline14: a run based on the underlying retrieval model of the common API, i.e., Lucene’s implementation of query likelihood model with Dirichlet smoothing [11].
- Median: The median retrieval results of all automatic runs submitted to this task.

Table 2: MAP and P@30 of each run. * and/or ◊ denotes significance difference from Baseline14 and/or Median respectively. Best value per measure is boldfaced.

Run	MAP	P@30
Baseline14	0.4250	0.6461
QUQueryExp5D25T	0.5155 ^{*,◊}	0.6697 [◊]
QUtmpDecay	0.4337	0.6473
QUQueryExp10D15T	0.4932 ^{*,◊}	0.6436
QUTQRM	0.4704 [◊]	0.6267
Median	0.4155	0.6261

We notice that our runs had better MAP compared to both baselines. However, only the two runs based on prf-based query expansion had *significantly* higher MAP than Baseline14. Moreover, these two runs along with QUTQRM had significantly better effectiveness compared to the median run. Interestingly, we see that only one run had better P@30 than Baseline14. Overall, 3 out of 4 runs had a slightly higher P@30 than the median, and as with MAP, the improvement was significant with QUQueryExp5D25T run.

The results showed that the non-temporal run QUQueryExp5D25T had the best performance on both measures compared to other runs and baselines. This shows that the typical and rather simple prf-based QE is an effective retrieval approach with microblog ad-hoc search.

As for temporal models, we see that the run QUTmpDecay has almost the same performance as Baseline14. This is not surprising since the retrieval model *t*-EXP is based on the QL model but using a temporal decay factor as a document prior. This might indicate that using such prior did not help in capturing the temporal aspect of the data and the task. The QUTQRM run had almost the same P@30 compared to Baseline14 but it notably improved MAP suggesting it helped improve the *overall* ranking of tweets, but not necessarily the top 30 ones. To understand the behavior of such models in relation to the given queries, analysis of the temporal nature of queries is needed.

3. TWEET TIMELINE GENERATION

Timeline generation is a new task that has been just introduced this year at the microblog track. It aims at producing a timeline of non-redundant chronologically-ordered tweets that are relevant to a given query issued at time q_t . The timeline basically constitutes a summary for a topic (e.g., event) represented by the given query. The definition of the task inherently imposes the need for an initial list of “potentially-relevant” tweets, which indicates that the new task is highly-dependent on the quality of the retrieval result list R_q (and thus the retrieval model used to retrieve those results).

3.1 Online Clustering Approach

This year, we adopted a simple online-clustering technique [1] to detect sub-events that are not redundant before producing the final timeline for a given query. The rationale behind this technique is that we need to detect such clusters without determining their number in advance. In online-clustering, the data to be clustered is processed in a stream, where the incoming data can either be added to an existing cluster or form a new cluster, thus having a dynamic set of clusters. The approach pipeline illustrated in Fig. 1 is outlined as follows:

1. **Ad-hoc Retrieval:** Given the query q at time q_t , a ranked list of 1000 tweets is retrieved using model m .
2. **Duplicate Removal:** Duplicates (or near-duplicates) of tweets were removed from the retrieval results by normalizing the tweets (i.e., removing stop words, URLs, and mentions) and then hashing the normalized tweets [22].
3. **Tweet Streaming:** Only the top k tweets were considered for timeline generation after removing the near-duplicates. Retrieval results are ordered in some criterion (e.g., chronologically, or based on retrieval scores) to form a stream of k tweets. The algorithm then processes the stream, one tweet at a time.
4. **Clustering:** We then construct clusters that represent “sub-topics” by processing the tweet stream. Initially, there are no clusters. Each incoming tweet is either added to an existing cluster if it exhibits a high similarity to it, or forms a new cluster if none of the existing ones were similar. Similarity between a tweet and a cluster can be measured in different ways, e.g., similarity between the tweet and the cluster’s centroid.
5. **Cluster Filtering:** Singleton clusters (i.e., clusters of only one tweet) can optionally be filtered out (i.e., not represented in the timeline) as they might be outliers.
6. **Tweet Selection:** After clustering all tweets in the stream, each cluster elects one or more tweets to represent it in the final timeline. There are several ways to select such tweets, e.g., the tweet that is most similar to the centroid.

3.2 Baseline

Since the retrieved tweets that appear in the tweet stream above can (by definition) represent a timeline, we used that list as a baseline approach to which we compare our online clustering approach.

3.3 Submitted Runs

Prior to TREC, the track organizers shared with the participants a small training set based on 10 queries from Tweets11 collection. We have conducted preliminary experiments using that set with different configurations and parameters for each of the steps of the proposed approach. We eventually chose the 4 runs described in Table 3.

Two submitted runs (indicated by **BL** postfix) are based on the baseline approach using two different retrieval models. The other two (indicated by **CL** postfix) used the online clustering approach with two other different retrieval models as well.

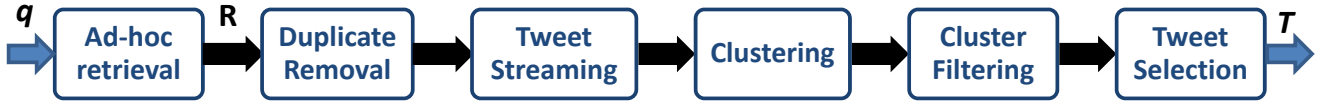


Figure 1: Pipeline of our TTG approach.

Table 3: Description of our official TTG runs

Run	Model	Parameters
QUQEd5t25TTgBL	PRF-QE	$k = 5, m = 25$
QUTqrmTTgBL	t -QRM	$k = 25, m = 5$
QUTmpDecayTTgCL	t -EXP	$r = 0.05$
QUQEd10t15TTgCL	PRF-QE	$k = 10, m = 15$

In all of the submitted runs:

- The top 75 tweets were selected after duplicate detection to form the tweet stream.
- The most similar tweet to the query in a cluster (i.e., the one with highest relevance score, which might generally change over the course of stream processing) acted as its centroid and hence the cluster similarity with any incoming tweet was measured by the similarity between the incoming tweet and the most relevant tweet. That tweet was eventually selected as the representative of the cluster in the timeline.
- Singleton clusters were not filtered out.
- Cosine similarity was used as the similarity function. A similarity threshold of 0.6 was used to guide clustering decisions.

3.4 Evaluation Setup

The same 55 queries used in the ad-hoc search task were also used in the TTG task. The submitted runs in the ad-hoc task constituted the judgment pool for TTG as well. An additional round of manual judgments was performed on the tweets that were judged as relevant to each query to form semantic clusters containing redundant tweets.

System effectiveness is measured using cluster precision and two versions of cluster recall. Cluster precision P is defined as the percentage of distinct semantic clusters that are represented in the generated timeline out of the tweets in that timeline. The *unweighted* version of cluster recall R_U is defined as the percentage of distinct semantic clusters that are represented in the generated timeline out of the judged semantic clusters. The *weighted* version R_W weights the semantic clusters based on the aggregate relevance levels of the tweets included in each cluster.⁴ Two versions of F1 are then used as the figure of merit, $F1_U$ and $F1_W$.

⁴<https://github.com/lintool/twitter-tools/wiki/TREC-2014-Track-Guidelines>

3.5 Experimental Results

Table 4: Evaluation (un-weighted) results of TTG submitted runs. Best precision and recall values are *italicized* and best F1 value is **boldfaced**.

Run	P	R_U	$F1_U$
QUQEd5t25TTgBL	0.2436	<i>0.3795</i>	0.2967
QUTqrmTTgBL	0.2366	0.3727	0.2894
QUQEd10t15TTgCL	0.3049	0.3277	0.3159
QUTmpDecayTTgCL	<i>0.3236</i>	0.3277	0.3256

Table 5: Evaluation (weighted) results of TTG submitted runs. Best precision and recall values are *italicized* and best F1 value is **boldfaced**.

Run	P	R_W	$F1_W$
QUQEd5t25TTgBL	0.2436	<i>0.5660</i>	0.3406
QUTqrmTTgBL	0.2377	0.5637	0.3333
QUQEd10t15TTgCL	0.3049	0.5316	0.3875
QUTmpDecayTTgCL	<i>0.3236</i>	0.5167	0.3980

Tables 4 and 5 show the performance of our submitted TTG runs in the measures described earlier. P and R indicate the average precision and recall respectively over all queries. F1 is just computed using the average precision and corresponding average recall, not as an average F1 over all queries.

The results show that, while the baseline approach had better recall (as it maximizes the number of represented clusters), the online clustering approach exhibited better precision (as it avoids redundant tweets/clusters) and thus better F1 values. Moreover, the exponential-decay-based model had better F1 values than the PRF-based QE model. More experiments and analysis of the results are needed to explain the reason behind that. We also notice that the F1 values are relatively low, which shows either the difficulty of the problem or the opportunity for improvements.

No median results per query (across participants) were shared by the track organizers, however, F1 results of all anonymous submitted runs from all participants (about 45 runs) were shared and illustrated in Figures 2 and 3 for unweighted and weighted versions respectively. In both figures, F1 values of QU runs were circled and the best of them was marked by the corresponding precision and recall values. In both cases, the best QU run was ranked among the top 10 (or probably 11) submitted runs, while all of them were ranked better than the median submitted run. This indicates the potential of the online clustering approach for tweet timeline generation problem.

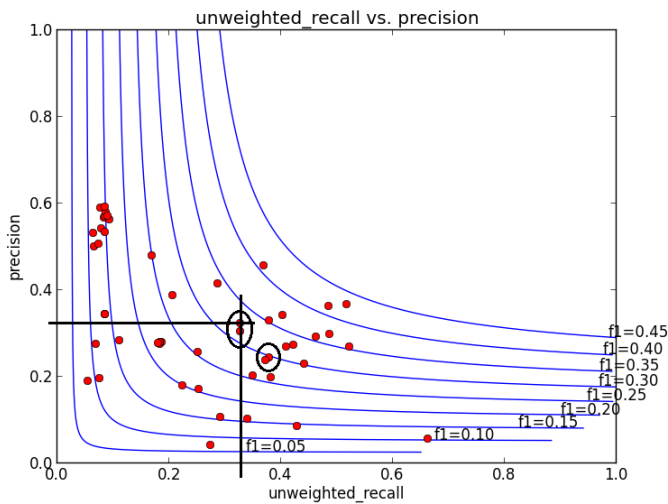


Figure 2: Performance of QU runs relative to other submitted runs (unweighted measures) in the TTG task.

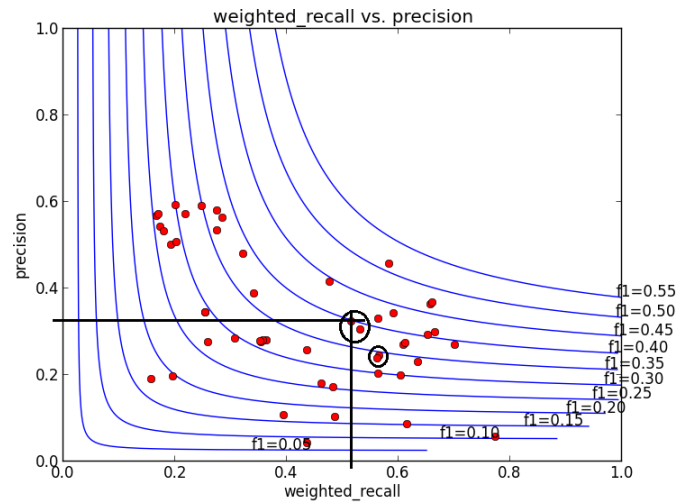


Figure 3: Performance of QU runs relative to other submitted runs (weighted measures) in the TTG task.

4. CONCLUSION

Continuing from our last year participation in the track [6], we again turned to context expansion-based retrieval models to perform ad-hoc search. We used two query expansion retrieval models, one that is the typical prf-based and the other uses temporal aspects of the query in selecting expansion terms. Furthermore, we retrieve tweets using a temporal model that was found effective in this context. The results showed the superiority of prf-based query expansion retrieval over all other retrieval models we used.

In our work on the TTG task, we employed the same retrieval models used in ad-hoc search to retrieve tweets for a given query. Online clustering of tweets with the help of near-duplicate detection was then used to produce a timeline for a given query. The results showed that clustering of tweets retrieved through the temporal query expansion retrieval model had the best effectiveness compared to our other TTG runs. Based on F1 measure, this run was also ranked among the top 10 runs submitted to this task.

5. ACKNOWLEDGMENTS

This work was made possible by NPRP grant# NPRP 6-1377-1-257 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

6. REFERENCES

- [1] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, ICWSM'11*, pages 438–441, 2011.
- [2] J. Choi and W. B. Croft. Temporal models for microblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 2491–2494. ACM, 2012.
- [3] M. Efron. Query-specific recency ranking: Survival analysis for improved microblog retrieval. In *Proceedings of the 1st Workshop on Time-aware Information Access (#TAIA2012)*, TAIA '12, 2012.
- [4] M. Efron, J. Lin, J. He, and A. de Vries. Temporal feedback for tweet search with non-parametric density estimation. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14*, pages 33–42. ACM, 2014.
- [5] 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. ACM, 2012.
- [6] M. Hasanain, L. Al-Marri, and T. Elsayed. QU at TREC-2013: Expansion experiments for microblog ad hoc search. In *22nd Text Retrieval Conference (TREC) 2013*. NIST, 2014.
- [7] M. Keikha, S. Gerani, and F. Crestani. Time-based relevance models. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11*. ACM, 2011.
- [8] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 591–600. ACM, 2010.
- [9] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01*, pages 120–127. ACM, 2001.
- [10] X. Li and W. B. Croft. Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Management, CIKM'03*, pages 469–475. ACM, 2003.
- [11] J. Lin and M. Efron. Overview of the TREC-2013 Microblog Track. 2013.

- [12] J. Lin and M. Efron. Temporal relevance profiles for tweet search. In *Proceedings of the 2nd Workshop on Time-aware Information Access (#TAIA2013)*, TAIA '13, 2013.
- [13] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.
- [14] K. Massoudi, M. Tsagkias, M. d. Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, and V. Mudoch, editors, *Advances in Information Retrieval*, number 6611 in Lecture Notes in Computer Science, pages 362–367. Springer Berlin Heidelberg, Jan. 2011.
- [15] V. Milicic, G. Rizzo, J. L. Redondo Garcia, R. Troncy, and T. Steiner. Live topic generation from event streams. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 285–288, 2013.
- [16] T. Miyanishi, K. Seki, and K. Uehara. Combining recency and topic-dependent temporal variation for microblog search. In P. Serdyukov, P. Braslavski, S. O. Kuznetsov, J. Kamps, S. Rüger, E. Agichtein, I. Segalovich, and E. Yilmaz, editors, *Advances in Information Retrieval*, number 7814 in Lecture Notes in Computer Science, pages 331–343. Springer Berlin Heidelberg, Jan. 2013.
- [17] T. Miyanishi, K. Seki, and K. Uehara. Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 439–448. ACM, 2013.
- [18] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. 2011.
- [19] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281. ACM, 1998.
- [20] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: Continuous summarization of evolving tweet streams. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 533–542, 2013.
- [21] I. Soboroff, I. Ounis, C. Macdonald, and J. Lin. Overview of the TREC-2012 Microblog Track. 2012.
- [22] K. Tao, F. Abel, C. Hauff, G.-J. Houben, and U. Gadiraju. Groundhog day: Near-duplicate detection on twitter. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1273–1284, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.
- [23] J. Teevan, D. Ramage, and M. R. Morris. #TwitterSearch: A comparison of microblog search and web search. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 35–44, 2011.
- [24] S. Whiting, I. A. Klampanos, and J. M. Jose. Temporal pseudo-relevance feedback in microblog retrieval. In R. Baeza-Yates, A. P. d. Vries, H. Zaragoza, B. B. Cambazoglu, V. Murdock, R. Lempel, and F. Silvestri, editors, *Advances in Information Retrieval*, number 7224 in Lecture Notes in Computer Science, pages 522–526. Springer Berlin Heidelberg, Jan. 2012.
- [25] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.