

Interactive Disaster Information Search System for Microblog by Minimal User Feedback

Sayaka Kitaguchi, Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara

Graduate School of System Informatics, Kobe University, Japan.
kitaguchi@ai.cs.kobe-u.ac.jp

Abstract. During disastrous events, much information is posted on microblogging services. These postings sometimes contain important information concerning, for example, safety and support for afflicted people. However, due to the overwhelming volume of information on microblog, it is often difficult to find such useful information using the current microblogging search system. Given the background, this paper proposes two types of interfaces for effectively identifying useful information in the event of disasters. First, postings containing similar contents are grouped and displayed in the chronological order so that users could easily identify a group of messages directly relevant to their urgent needs. Second, a message chosen by the users is used to reformulate the initial search query to refine the search results. In order to show the effectiveness of the proposed interfaces, a user study is conducted on the Twitter Corpus collected during the event of the Great East Japan Earthquake in 2011.

Keywords: Twitter, Disastrous events, User Study, User Interface, Visualization, Interactive Search

1 Introduction

In a disastrous event, much information is posted on microblogging services. For example, when the Great East Japan Earthquake occurred on March 11, 2011, numerous messages were posted on Twitter by the crowd of people. These postings, called *tweets*, naturally related to the event and sometimes contain critical information concerning safety and support for afflicted people. Due to the overwhelming volume of information, however, finding useful information for a specific region and/or a particular need is difficult for the current microblog search system.

Given the background, this paper proposes a disaster information search system on Twitter. The system searches for useful information for disaster victims and their supporters by focusing on the following aspects: (1) Our system uses Maximal Marginal Relevance (MMR) [3], a standard technique often used in multi-document summarization, to organize search results in terms of the contents of tweets, (2) our system displays the groups in the chronological order to organize search results in terms of when the tweets were posted, and (3) our system automatically extends the initial query by leveraging the tweet clicked by

the users in the initial search results to retrieve a new set of tweets supposedly more relevant to the users' particular information need. In order to investigate the effectiveness of the proposed interfaces, a user study is conducted on the Twitter Corpus collected during the event of the Great East Japan Earthquake.

The remainder of the paper is organized as follows: Section 2 details the framework and its components of our proposed disaster information search system. Section 3 describes the design of the user study. Section 4 reports on the results and discusses the implications. Section 5 summarizes the related work in comparison with the present study. Lastly, Section 6 concludes with a brief summary and possible future directions.

2 Interactive Disaster Information Search System

2.1 Interface

Figure 1 shows a screenshot of our proposed system. First, the users type a query that represents their information needs into the search box on the top of the screen (① in Figure 1). Then, the search results are grouped by sub-topics and is displayed in the left-most column (② in Figure 1). We set the number of tweets displayed in each group to $\log_3 N$, where N is the total number of tweets contained in the group. As the number of tweets presented in each group increases logarithmically with the total number of tweets in the group, the users can visually have a rough estimate of how actively the sub-topic is mentioned. When any tweet in a sub-topic group is clicked, the frame of the group is extended vertically and all tweets in the group are displayed.

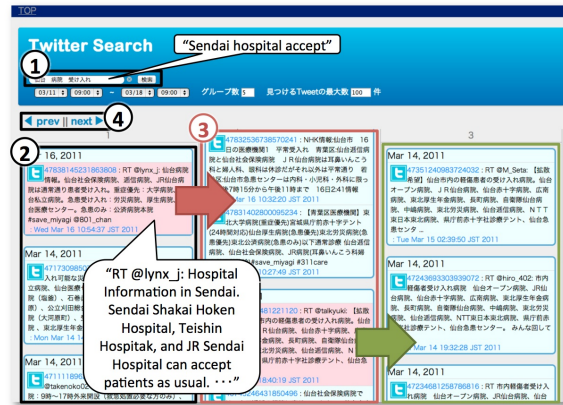


Fig. 1. Screenshot of the proposed system.

Further, when any tweet is double-clicked, the system performs a search again but with a new, reformulated query by taking advantage of the contents of the

clicked tweet, and displays new results in a new column (③ in Figure 1) on the right of the column containing the clicked tweet. Search results can be displayed up to three columns on a screen. When more columns are needed, the oldest, left-most column is pushed out to the left. Note that the old results are still kept by the system. When the users click the “prev” or “next” button at ④ in Figure 1, system slides all columns to the right or the left of the screen and shows hidden results.

2.2 Grouping Search Results

Due to the many active users, Twitter has a characteristic that when an exceptional event occurs, numerous tweets about the event are posted. Those tweets naturally contain topic-related terms, and thus it is possible to group them by sub-topics based on terms. By organizing search results into such groups, users would be able to find the relevant tweets more quickly by reading a few tweets in each group. This way, it is also possible to find minor sub-topics, which may be otherwise overlooked due to the small number of tweets despite its potential importance (e.g., information sent for/from isolated areas). In addition, the users could have a rough estimate of how actively each sub-topic is mentioned by looking at the size of the group as it reflects the total number of tweets in the group.

Our approach uses MMR for grouping related tweets. MMR is one of the extraction-based multi-documents summarization algorithms. It scores each document in terms of two aspects, that is, (a) relevance between a document and a query, and (b) dissimilarity between a document and already selected documents, and selects a document that has the highest score defined as:

$$\lambda Sim_1(d_i, q) - (1 - \lambda) \max_{d_j \in S} Sim_2(d_i, d_j), \quad (1)$$

where Sim_1 is the similarity between a candidate document d_i and a query q . In this study, a normalized search score returned by a search engine is used as Sim_1 . S is a set of already selected documents (tweets). Sim_2 is the similarity between d_i and d_j and is estimated by the cosine similarity based on the inverted document frequency (IDF) of terms in the documents. The parameter λ ($0 \leq \lambda \leq 1$) controls the emphasis on relevance between the query and the document. High λ means little diversity among each group. In this paper, we set $\lambda = 0.8$ based on our preliminary experiment using developing data not used for evaluation.

MMR iteratively selects a tweet based on Equation (1) for specified times. The selected tweet is a representative message of each group. The number of selected tweets here corresponds to the number of groups to be generated by the system. Our system allows the users to specify the number of groups. Then, each remaining tweet that has not been selected by MMR is assigned to the most similar group based on Sim_2 . Finally, the groups are sorted based on the average posting time of the tweets in each group in the chronological order. In order to reduce redundancy in the search results, our system displays only one tweet for multiple, duplicated tweets.

2.3 Query Reformulation by a Clicked Tweet

In disastrous events, there will be urgent needs for information related to, for example, damage or restoration in a specific region. However, users' queries tend to be short and ambiguous, resulting in a large amount of irrelevant information. For instance, when a user who would like to know the situation of damage in the Ishinomaki area may issue a query "Ishinomaki situation." Then, unrelated information such as a situation of medical supports or a situation of rations in Ishinomaki may be also presented. We propose the reformulation of a query to tackle this problem. In searching for relevant information, the users naturally select a tweet satisfying or related to their information needs. Then, our system takes it as a feedback and generates a new query using the selected tweet and searches for tweets similar to the selected tweet. Although the particulars of the query reformulation is different, this functionality is based on the report that search performance was significantly improved by taking advantage of a user-selected relevant tweet [8].

In reformulating a query, it is important to extract topical terms reflecting users' interest. A simple approach to identifying such on-topic vocabulary is to extract infrequent terms in a corpus based on IDF [2]. In addition, it is often the case that nouns are suited as topical terms. Based on these criteria, our system first performs morphological analysis with Kuromoji³, an open-source Japanese morphological analyzer, to the selected tweet and extracts proper nouns, common nouns, and verbal nouns as query terms. Next, we eliminate stop words and morphemes composed of only two or less alphanumeric characters, and select the terms with M highest IDF values as a set of query terms Q_t . Let Q_o be a set of initial query terms given by the users at the beginning. Based on the two term sets, Q_t and Q_o , our system searches for tweets containing at least one query term from Q_t and one query term from Q_o . Using this functionality, the users can dig in potentially relevant information by simply clicking a tweet most pertinent to their interest without manually modifying the initial query.

3 Evaluation

3.1 Research Questions

In order to investigate the effectiveness of the proposed interfaces, we conducted a user study on the Twitter Corpus collected for the Great East Japan Earthquake occurred in 2011. Specifically, this study focused on the following questions:

- **The effect of grouping:** Can users find relevant tweets more efficiently by grouping search results compared to the one without grouping?
- **The effect of query reformulation:** Can users find more relevant tweets by query reformulation?
- **The effect of combination:** Is it effective to combine grouping and query reformulation?

³ <http://www.atilika.org/>

3.2 Experimental Settings

Dataset. In this experiment, we used the Twitter Corpus provided by Twitter Japan Co., Ltd. for the Great East Japan Big Data Workshop⁴. This corpus consists of approximately 180 million tweets written in only Japanese for one week after the Great East Japan Earthquake occurred (i.e., from March 11 to 18, 2011). Figure 2 shows an example tweet in the corpus, which consists of four fields: (i) Tweet ID, (ii) User ID, (iii) Posting time, and (iv) Text (the English translation was inserted by the authors).

(i) 46126534246932480	(ii) 224956629	(iii) 2011-03-11 17:33:17
(iv) とりあえず地震前までのツイート確認 おいらは生きてるぞ w (Checked tweets that have been posted before the earthquake. I'm alive).		

Fig. 2. Example tweets in the Twitter corpus.

Microblog search. We indexed the corpus with the Lucene⁵ full-text search engine, where the language model with Bayesian smoothing using Dirichlet priors was employed. Here, we set the smoothing parameter μ to 2,500 based on our pilot experiment. Kuromoji was used for morphological analysis.

Baselines. The proposed system was compared with two interfaces, namely, Flat and Group, illustrated in Figure 3. The Flat (FL), which is similar to the official search interface of Twitter, displays a flat list of search results in the descending order of the relevance score. We used search scores returned by Lucene as the relevance score. The Group (GR) displays the search results grouped by sub-topics. Both FL and GR retrieve 100 tweets per page and present only one tweet for duplicated tweets having the same contents. More results can be retrieved by clicking a button labeled as “Read More” or “Search More.” Our proposed system, illustrated in Figure 1, is hereafter referred to as “Group + Query Reformulation (GQR).” GQR displays grouped search results and provides a query reformulation leveraging the tweet selected by users. GQR also retrieves 100 tweets at once. We set M , the number of terms to be extracted from the selected tweet, to five based on our pilot experiment. The number of groups was experimentally fixed at five during throughout this experiment. The appropriateness of the number of groups is discussed based on a questionnaire.

We evaluated the effect of grouping by comparing FL and GR, the effect of query reformulation by comparing GR and GQR, and the effect of combining grouping and query reformulation by comparing GR and GQR.

Tasks. We designed a task as a pair of an initial query and its information need. To evaluate the effectiveness of the proposed interfaces during disastrous events,

⁴ www.biglobe.co.jp/pressroom/release/2011/04/27-1

⁵ <http://lucene.apache.org/core/>



Fig. 3. Screenshot of the baseline systems, Flat list (corresponding to the existing typical interfaces) and Group, used for our experiments.

it is important that the tasks reflect the real information needs. To this end, we focused on “inquiry tweets.” We defined inquiry tweets as those asking for certain information and thus contain the phrase, “please tell me” which is expressed in Japanese as “おしえてください”, “おしえて下さい”, “教えてください”, or “教えて下さい.” Inquiry tweets in the corpus can be seen as users’ information needs and, especially in the time of disaster, they would reflect the most critical ones. There are of course many tweets containing the phrase but not related to the event of interest. To identify relevant questions, we took advantage of the “retweet” function of Twitter. A retweet is a tweet copying another tweet posted by others to disseminate the information. The fact that a tweet was retweeted many times means that many users found the tweet important and we used it as an indicator of the importance of a question. A total of 114,315 inquiry tweets were found in the corpus and were sorted in descending order of the retweet rate r_t of tweet t defined in Equation (2) each day.

$$r_t = \frac{\# \text{ of times } t \text{ was retweeted on that day}}{\text{the total } \# \text{ of times } t \text{ was retweeted}} \quad (2)$$

Based on the ranking of tweets for each day, we selected inquiry tweets by manually examining them in the following steps.

1. Selected tweets related to the earthquake.
2. Removed redundant tweets, which ask the same questions as the already selected tweets.
3. Retained only tweets whose answers were found in the corpus.

A total of 36 inquiry tweets were selected. In this study, tweets containing RT at the beginning were simply regarded as retweets. This is partly because it is difficult to obtain tweet-retweet relations from only this corpus, and the precise relations are not very important to generate search topics. Then, search intentions were manually interpreted based on the selected inquiry tweets by the authors. The initial queries used for user study were also prepared from the intentions. Two examples of the constructed tasks are shown in Table 1.

Table 1. Examples of search tasks.

Task#	Initial query	Search intention	Time-stamp
3	ディズニーランド 避難 (Disneyland shelter)	I want to find shelters near Tokyo Disneyland.	2011/03/11 20:00
5	仙台 病院 受け入れ (Sendai hospital accept)	I want to find hospitals in Sendai which can accept patients.	2011/03/11 20:00

Time-stamp identifies when the inquiry tweet was retweeted most actively. We defined the time range for each task to be between the beginning of the corpus and 24 hours after its time-stamp. For example, in retrieving tweets for task#5 (cf. Table 1), the time range was specified from March 11 at 9:00 to March 12 at 20:00.

Procedure. We gathered 18 participants for the user study. Among them, 13 are male and five are female; 17 are students in computer science and one is an office worker; all in their 20's. Each participant performed a total of six tasks (two tasks for each system). The order of using the systems and performing the tasks were allocated equally based on Latin square. All pairs of a system and a task were carried out once. In each task session, the participant conducted a search using the initial query (see Table 1) and was asked to find five relevant tweets within five minutes. When they found five tweets, the session was closed. The following gives a precise procedure.

1. The participant received a general instruction of the study.
2. He performed the assigned tasks in the allocated order.
 - (a) He received an instruction of the particular system to use, and went through a training task to become familiar with the interfaces and the tasks.
 - (b) He performed the assigned two tasks. In each task, we automatically recorded the time taken to complete the task, the time taken to search by the system, and the tweets identified as relevant by the participant.
 - (c) He filled in the questionnaire about the system used.
3. In the end, he filled in the questionnaire about the experiment.

4 Results and Discussion

4.1 Task Completion Time

Table 2 shows the average time taken to complete a task to evaluate the efficiency of the systems. Task completion time T is computed as $T_t - T_s$, where T_t is the time between the beginning and the end of a task by a participant; T_s is a processing time of a system to search the Twitter Corpus.

We observe that the task completion time for GR is shorter than those for FL and GQR, while there is little difference between FL and GQR. When looking

Table 2. Average task completion time with standard deviation, where the shortest time is shown in bold.

System	Average time (s)	Standard deviation
FL	154.0	98.72
GR	119.4	74.27
GQR	160.0	80.49

Table 3. Average task completion time of GR and GQR for different orders of their use, where significant difference is indicated with †.

System	GR→GQR	GQR→GR
GR	121.4	117.4 [†]
GQR	143.3	176.7

at individual tasks, GR recorded the shortest task completion time for 64% of the tasks (not shown in Table 2). These results suggest that the grouping functionality implemented in GR has helped users gather relevant tweets more quickly. There’s no time to lose in a critical situation, so that our system is helpful for volunteers to quickly find useful information and provide it to suffering people in the disaster area. We conducted a multiple comparison with Bonferroni corrected Wilcoxon signed-rank test where the significance level α was set to 0.05. However, we found no significant difference among the three systems and the advantage of grouping is inconclusive.

Comparing GR and GQR (the proposed system), the task completion time of GQR, was found longer than that of GR despite the fact that GQR is also equipped with grouping. A possible explanation is the difficulty of the interface to use. The interface for query reformulation implemented in GQR is not familiar to most users and is supposedly harder to learn than FL or GR. To investigate the interpretation, Table 3 shows the average task completion time of GR and GQR for different orders of system usage. The first row of Table 3 represents the order the two systems were used. For Example, GR→GQR means that the user first used GR and then GQR. A dagger (†) indicates a significant difference from GQR based on a Bonferroni corrected permutation test with 1000 times random sampling where the significance level α was set to 0.05.

We can see from Table 3 that the task completion time of GR is significantly shorter than GQR when the users used GR after GQR. In contrast, when users used GR before GQR, there was no significant difference. In other words, using GQR helped users to learn how to use GR but not the other way around. These results implicate that GQR is more difficult to learn than GR, which may explain the longer task completion time for GQR in Table 2. Moreover, in an additional questionnaire, some participants pointed out that the GQR interface is hard to use.

Table 4. Average P@5, sum of relevance scores, average P@5 without duplicates, and # of relevant tweets found only by each system with ± 1 standard deviation.

System	P@5	Relevance score	P@5 w/o duplicates	# relevant tweets found only by a respective system
FL	0.48 \pm 0.37	466 \pm 6.26	0.39 \pm 0.30	43
GR	0.54 \pm 0.40	534 \pm 5.81	0.39 \pm 0.30	54
GQR	0.53 \pm 0.34	502 \pm 5.81	0.40 \pm 0.26	61

4.2 Relevance of Search Results

Relevance judgement. Through 108 task sessions (18 participants \times 6 tasks from a total of 36 tasks), 462 tweets were marked as relevant by the participants. These results were assessed by two assessors per a pair of a task and a tweet. In total, three assessors participated in the assessment. They were students majoring in computer science. This section details the two types of relevance judgements we conducted.

First, we defined relevant tweets. Annotators assessed each tweet using a relevance grade (“relevant,” “irrelevant,” or “neither”). We regarded the tweets that were judged as “relevant” by two annotators as relevant and the others as irrelevant. As a result, 281 relevant tweets (61% of the all tweets found by the participants) were identified. Additionally, we defined an alternative relevance score of a tweet as the sum of the judgements by two annotators, where “relevant” was treated as two points and “neither” as one point.

Second, we identified relevant tweets similar to each other in order to evaluate the diversity of the information gathered by the participants. Annotators gave a particular label to a group of tweets with the same contents. As a result, 131 relevant tweets were labeled by 36 different labels, which correspond to 36 groups of tweets with the same contents for each.

Search performance. We evaluated the effectiveness of the systems with precision at 5 (P@5) and the relevance score defined in Section 4.2. In this experiment, P@5 is defined in Equation (3). When the participants gathered less than five tweets during a task session, the remaining slots were treated as irrelevant tweets.

$$P@5 = \frac{\# \text{ of relevant tweets gathered in a task session}}{5} \quad (3)$$

Table 4 shows the average P@5 and the sum of relevance scores by each system. The right two columns “P@5 without duplicates” and “# relevant tweets found only by a particular system” will be discussed later. It can be seen that the participants found more relevant tweets by GR and GQR than FL on average, which is presumably attributed to the fact that both GR and GQR group similar tweets. However, using Bonferroni corrected Wilcoxon signed-rank test ($\alpha = 0.05$), there was no significant difference among three systems.

We then evaluated search performance when duplicate tweets were removed. Intuitively, multiple tweets in the same group were regarded as one tweet in

Table 5. A list of questions asked in post-experiment questionnaire.

	Question and Choices
Q1	Was the interface useful for finding relevant information? 5: Very useful 4: Useful 3: Neither 2: Useless 1: Very useless
Q2	How many irrelevant tweets did you have to read for finding relevant information? 5: Very few 4: few 3: Neither 2: Many 1: Very many
Q3	Do you want to use the interface to search microblog again? 5: Strongly agree 4: Agree 3: Neither 2: Disagree 1: Strongly disagree
Q4	Was the number of groups of tweets appropriate? 5: Too many 4: Many 3: Just right 2: Few 1: Too few
Q5	Did grouping make finding relevant information easier? 5: Very easy 4: Easy 3: Neither 2: Difficult 1: Very difficult
Q6	Did query reformulation make finding relevant information easier? 5: Very easy 4: Easy 3: Neither 2: Difficult 1: Very difficult

computing precision. We call this measure precision without duplicates. The result is found in Table 4. GQR scored slightly higher than the other systems but the difference is not significant.

For a more detailed analysis on individual systems, we then looked at the number of relevant tweets found only by a particular system, which indicates the novelty of search results. Comparing the results of three systems presented in Table 4, it is found that participants using GQR were able to find the largest number of tweets that were not found by FL or GR. For a concrete example, let us take the task #3 (see Table 1). While tweets (e.g., “Shelters near Tokyo Disneyland. Maihama Elementary School and Tokai University Urayasu High School are close”) describing the places of shelters were found by all the systems, the participants using GQR found more detailed information, such as “Maihama Elementary School is small, so you should go to Horie Elementary School or Miakegawa Elementary / Junior High School.” This result exemplifies the advantage of query reformulation using a clicked tweet for finding relevant tweets that are overlooked by the other systems.

4.3 User Satisfaction

Finally, we discuss the results of the questionnaire. Table 5 is a list of the questions asked upon the completion of the assigned tasks. The participants answered each question on a five-point scale.

Table 6 shows the results, where GR and GQR had higher scores than FL in Q1–Q3, which means that grouping and query reformulation were on average favorably accepted by the participants. In Q2 and Q3, there is a significant difference between FL and GR, but there is no significant difference between FL and GQR. The main reason would be the unfamiliarity of the query reformulation interface and the longer task completion time as discussed earlier. In Q4, we asked the appropriateness of the number of groups. As more than a half of the participants selected “just right”, the number (five) used in this experiment seems appropriate. However, since the optimum number would depend on a type of a task and the number of tweets to search, it would be ideal to dynamically

Table 6. The questionnaire results, where the average and standard deviation (\pm SD) are reported. Significant improvement ($\alpha = 0.05$) with respect to FL is indicated by †.

System	Q1	Q2	Q3	Q4	Q5	Q6
FL	3.22 \pm 0.92	2.50 \pm 0.90	3.22 \pm 0.92	-	-	-
GR	4.00 \pm 0.58†	3.44 \pm 0.83†	3.89 \pm 0.57†	-	-	-
GQR	4.00 \pm 0.67†	3.17 \pm 0.90	3.72 \pm 0.73	4.33 \pm 0.75	3.17 \pm 0.50	3.89 \pm 0.81

set the number for a given query and the size of the data. Lastly, Q5 and Q6 indicate the satisfaction in grouping and query reformulation, respectively. Both questions had relatively positive responses, supporting their usefulness.

5 Related Work

Microblog has become popular since Twitter launched its service in 2007. Especially, in the last few years, the use of microblog in disastrous events has been drawing much attention. Vieweg et al. [13] analyzed messages posted on Twitter during disasters and reported the behavior of disaster information in microblog. Sakaki et al. [10] proposed earthquake detection by treating Twitter users as a sensor. In addition, an application for searching disaster information was also proposed. Abel et al. [1] presented Twitcident, a Web-based system for filtering and faceted searching for disaster information. The present study focused on the improvement of microblog search interfaces for disaster information.

With the popularization of microblog, microblog search has also been actively studied [12], where traditional web search approaches were often adapted [6, 7]. In addition, Efron et al. [4, 5] proposed relevance feedback by using temporal properties of relevant tweets. Miyanishi et al. [8] proposed tweet selection-based relevance feedback, which uses a tweet selected by the users to refine search results and exploits temporal properties of refined search results. The purpose of this paper is not the improvement of search algorithm but the evaluation of the effectiveness of search interfaces including the grouping and the query reformulation particularly for disastrous events.

Microblog summarization is another research area related especially to grouping search results. O'Connor et al. [9] proposed the microblog search system by grouping search results similarly to our approach. Their approach identified groups of tweets by classifying search results based on the presence or absence of a high frequency phrase in a corpus, whereas our approach identified them based on textual contents of each tweet. Another summarization approach not using groups was proposed by Sharifi et al. [11].

6 Conclusion

In this study, we proposed and implemented the microblog search system, which provides two key features for efficiently find critical disaster information. One is to group search results and the other is query reformulation using a clicked

tweet. In addition, we proposed a strategy to construct search topics reflecting real information needs during disastrous events by taking advantage of retweets. Through evaluation, our user study indicated that, on average, the users were able to find useful information more quickly by grouping, although the difference was not statistically significant. For query reformulation, there was no advantage as to the task completion time. Our analysis on the order of system use suggested that it was in part due to the unfamiliarity of the query reformulation interface. However, another analysis on the novelty of search results indicated that query reformulation was effective to find more detailed information than those found by the other systems.

For future work, we plan to improve the user interface for query reformulation and to perform a larger user study to validate the effectiveness of the proposed interfaces. Also, we would like to consider the credibility of information on Twitter since some information may be false whether deliberately or not. There is much existing work in this area and could be incorporated in our system.

link.springer.com

References

1. Abel, F., Hauff, C., Houben, G.J., Stronkman, R., Tao, K.: Twitcident: fighting fire with information from social web streams. In: WWW. pp. 305–308 (2012)
2. Bendersky, M., Metzler, D., W. Croft, B.: Parameterized concept weighting in verbose queries. In: SIGIR. pp. 605–614 (2011)
3. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for re-ordering documents and producing summaries. In: SIGIR. pp. 335–336 (1998)
4. Efron, M.: The University of Illinois’ graduate school of library and information science at TREC 2011. In: TREC (2011)
5. Efron, M., Organisciak, P., Fenlon, K.: Improving retrieval of short texts through document expansion. In: SIGIR. pp. 911–920 (2012)
6. Metzler, D., Cai, C.: USC/ISI at TREC 2011: Microblog track. In: TREC (2011)
7. Miyanishi, T., Okamura, N., Liu, X., Seki, K., Uehara, K.: TREC 2011 microblog track experiments at Kobe University. In: TREC (2011)
8. Miyanishi, T., Seki, K., Uehara, K.: TREC 2012 microblog track experiments at Kobe University. In: TREC (2012)
9. O’Connor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for twitter. In: ICWSM. pp. 2–3 (2010)
10. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW. pp. 851–860 (2010)
11. Sharifi, B., Hutton, M.A., Kalita, J.: Summarizing microblogs automatically. In: HLT-NAACL. pp. 685–688 (2010)
12. Soboroff, I., Ounis, I., Lin, J.: Overview of the TREC-2012 microblog track. In: TREC (2012)
13. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: CHI. pp. 1079–1088 (2010)