IRIT at TREC Microblog 2015

Abdelhamid Chellal, Lamjed Ben Jabeur, Laure Soulier, Bilel Moulahi, Thomas Palmer, Mohand Boughanem, Karen Pinel-Sauvagnat, Lynda Tamine, Gilles Hubert

IRIT at TREC Microblog 2015

Abdelhamid Chellal, Lamjed Ben Jabeur, Laure Soulier, Bilel Moulahi, Thomas Palmer, Mohand Boughanem, Karen Pinel-Sauvagnat, Lynda Tamine, and Gilles Hubert

{chellal, jabeur, soulier, moulahi, palmer, boughanem, sauvagnat, tamine, hubert}@irit.fr,
Université de Toulouse UPS-IRIT,
118 route de Narbonne F- 31062 Toulouse cedex 9

Abstract. This paper presents the participation of the IRIT laboratory (University of Toulouse) to the Microblog Track of TREC 2015. This track consists in a real-time filtering task aiming at monitoring a stream of social media posts in accordance to a user's interest profile. In this context, our team proposes three approaches: (a) a novel selective summarization approach based on a decision of selecting/ignoring tweets without the use of external knowledge and relying on novelty and redundancy factors, (b) a processing workflow enabling to index tweets in real-time and enhanced by a notification and digests method guided by diversity and user personalization, and (c) a step by step stream selection method focusing on rapidity, and taking into account tweet similarity as well as several features including content, entities and user-related aspects. For all these approaches, we discuss the obtained results during the experimental evaluation.

Keywords: real-time, social media, user profile, novelty, redundancy, filtering, clustering, rapidity, entities, personalization

1 Introduction

It is well-known that social media data-streams include a wide range of useful information that are somehow difficult to exploit for users [1]. One main challenge consists in personalizing and diversifying tweet digests and notifications with the goal to educate the user in the information access context. Although several models have been proposed in the context of ad-hoc tweet search [2, 3], the task of notifying relevant tweets in real-time to a user, which is proposed by the Microblog Track of TREC 2015, is still under-explored. Indeed, the Microblog Track proposes a real-time filtering track aiming at monitoring the social media data-stream in order to push tweets to users with respect to their topical interest-based profile. One main assumption yields in the TREC guidelines is that notifications and digests might enable the user to learn more about a particular content. In this aim, the track is split into two main scenarios:

1. The Scenario *A*, called "*Push notifications on a mobile phone*", consists in an instantly and personalized tweet notification assuming a short time period between the tweet publication and the tweet triggering.
2. The scenario *B*, called "*Periodic email digest*", remains on a tweet aggregation into an email digest, periodically sent to a user.

In this paper, we investigate three main approaches aiming at retrieving tweets in a real-time *fashion* with respect to the push and digest scenarios:

– A novel selective summarization approach wherein the decision of select/ignore is made on a tweet basis without the use of external knowledge. We define salient tweets as those bringing new information and are not similar to the previously selected tweets in the summary. The decision of select/ignore an incoming tweet is based on two dimensions, novelty and redundancy which are evaluated using Hybrid *TF-IDF* and *KL divergence* respectively.

– A processing workflow enabling to index tweets in real-time using filters, meta-data enhancement and topical interest-based clusters so as to notify and digest tweets in real-time by taking into account diversity and users' interest profile.
– A step by step stream selection that focus on rapidity and that take into account several features. These features are divided into three groups, including features about content, entities and user.

This paper is organized as follows. Section 2 introduces the tweet real-time filtering based on novelty and redundancy measurements and discusses the results. Section 3 describes the periodic search for filtering real-time tweets. Section 4 presents the tweet Selection model based on speed and feature scores. Section 5 concludes the paper.

2 Tweet Real time filtering based on novelty and redundancy measurement

The main purpose of the outlined approach is to provide a short number of tweets with maximum coverage, minimum redundancy and low latency. These requirements are fulfilled as follows: (a) The outlined approach is a fully real-time that makes select/ignore decision as soon as the tweet become available. (b) The decision of selecting a given tweet is based on two dimensions: the novelty and the redundancy. The former aims to detect new information regarding ones previously seen in stream while the later is used to avoid pushing an information already selected which keeps the summary from being redundant.

Given an event described by keywords and a stream S of tweets T_i, our approach acts like a filter where only tweets which contain at least two keywords that describe a given event are considered. An incoming tweet T_i with timestamps t_i will be added to the summary R if and only if:

$$\begin{cases} NS(T_i) >= \max\limits_{\forall\, T_j \in S^{t_i},\, t_j < t_i} [NS(T_j)] \\ RS(T_i) >= \max\limits_{\forall\, T_j \in R^{t_i},\, t_j < t_i} [RS(T_j)] \end{cases} \tag{1}$$

Where $NS(T_i)$ and $RS(T_i)$ are the novelty and the redundancy scores of an incoming tweet T_i. S^{t_i} and R^{t_i} are the stream and the summary at t_i (publication time of tweet T_i) respectively.

Combining this two dimensions as a conjunctive condition provides complementarity between them allowing to fulfil the requirements related to novelty, shortness and low redundancy. With a linear combination, a tweet with high novelty and low redundancy scores or vice versa will likely be added to the summary. Also, notice here that the threshold is parametric-free value, it is evaluated according to the previously seen values.

2.1 Novelty score

Novelty detection is generally based on similarity measures where the new document is compared to all previously seen documents or to summary only. Due to the rapid growth of the number of posted tweet in stream, similarity comparison does not fit well a real time filtering scenario. To overcome this limit, we propose to use HybridTF-IDF [4] as measure of novelty. The intuition behind this proposition is that a novel tweet is the one that contains a good mixture of new and important terms in the relevant tweets stream for an event. A tweet with only new terms is more likely to be a spam and irrelevant to the event of interest.

Hence, the Inverse Document Frequency (IDF) at stream level is used as a measure of term novelty [5]. To evaluate the importance of the term within stream, we adopt the formula proposed by [4] in which the entire collection of tweets is considered as one document for computing the term frequency. Notice here that in our approach only tweets that contain keyword describing the event of interest are considered. In addition, to take into account the temporal distribution of terms in the stream, the HybridTF-IDF weight is combined with decay function. It gives a high weight to new words and those

did not appear in last time window. Thereby, the novelty score of the tweet T_i with timestamps t_i is measured as follows:

$$NS(T_i) = \sum_{w_j \in T_i} TF(w_j) \times IDF(w_j) \times Decay(w_j) \tag{2}$$

$$TF(w_j) = \frac{\#of w_j \; InAlltweet}{\#WordInAllTweet}, \; IDF(w_j) = log_2(\frac{\#Tweets}{\#Tweets \; w_j \; Occurs}) \tag{3}$$

$$decay(w_j) = \begin{cases} \left(\frac{\Delta t(w_j)-N}{N}\right)^2 if \Delta t(w_j) <= 2N \\ 1 \; otherwise \end{cases} \tag{4}$$

Where $\Delta t(w_j) = t^i_{w_j} - t^{i-1}_{w_j}$ represents the time since the previous occurrence of the word w_j in the stream. N represents the size of the time window.

2.2 Redundancy score

To assess the redundancy score between the incoming tweet regarding the summary, we propose to measure the divergence between the language model of incoming tweet and language model of each tweet in the summary. In our approach, Kullback-Leibler (KL) divergence [6] was used in which the divergence between two tweets T_i, T_j is evaluated as follows:

$$KL(T_i, T_j) = \sum_{w_k \in T_i \cap T_j} \theta_{T_i}(w_k) \log \frac{\theta_{T_i}(w_k)}{\theta_{T_j}(w_k)} \tag{5}$$

where θ_{T_i} is the uni-gram language model of tweet T_i and $\theta_{T_i}(w_k)$ is the probability of occurrence of term w_k in tweet T_i.

The incoming tweet should have a high divergence with the most similar tweet among the summary. The latter is the one that have the minimum Kl divergence with the incoming tweet. Thereby, the redundancy score of an incoming tweet T_j is defined by the minimum KL divergence regarding each tweet in the summary R^{t_i} at time t_i as follows:

$$RS(T_i) = \min_{\forall \, T_j \in R^{t_i}} KL(T_i, T_j) \tag{6}$$

To avoid the problem of zero probabilities, Jelinek-Mercer (JM) smoothing was used, it linearly combines the tweet model and stream model as follows:

$$\theta_{T_i}(w_j) = \lambda \times P_{T_i}(w_j) + (1-\lambda) P_{S^{ti}}(w_j) \tag{7}$$

Where λ is a smoothing parameter. $P_{T_i}(w_j)$ and $P_{S^{ti}}(w_j)$ are the probability of occurrence of term w_j in tweet T_i and in stream S at time t_i respectively. They are evaluated using the maximum likelihood estimation (ML) as follows:

$$P_{T_i}(w_j) = \frac{tf_{T_i}(w_j)}{|T_i|} \; , \; P_{S^{ti}}(w_j) = \frac{tf_{S^{ti}}(w_j)}{|S^{ti}|} \tag{8}$$

Where $tf_{T_i}(w_j)$ and $tf_{S^{ti}}(w_j)$ are the frequency of w_j in tweet T_i and stream S at time t_i. Smoothing parameter λ was set to 0.9 following [7] recommendation.

Table 1. Performance metric of Real time filtering based on novelty and redundancy score.

	Scenario A		Scenario B
Metric / Run	ELG	nCG	nDCG
IRITKLTFIDF	0.2652	0.26	0.1784
Max across all submitted runs	0.4715	0.4943	0.5114

2.3 Submitted runs and results

According to the used threshold and a way to compute global statistics (TF and IDF), two different configurations of the outlined approach were used for scenario A (Push notifications on a mobile phone) and Scenario B (Periodic email digest). For the former the threshold was set to the maximum of previous seen values and the IDF and TF are evaluated at the time of processing the incoming tweet. In the later, the threshold in the equation 1 was set to the average which might allow to have further tweets in summary. Also, the global statistics are estimated over all collection tweets per event for each day before starting the filtering process. Table 1 reports average performance of the aforementioned configurations per topic (first row) and the average per topic of the maximum performance across all submitted runs. Our approach has shown promise, and is worthy of further investigation, especially the impact of the threshold and the possibility to take into account other features for select/ignore decision making in order to improve cumulative gain.

3 Periodic search for filtering real-time tweets

3.1 Real-time processing of Twitter Stream

Based on a real-time processing framework, we implement a processing work-flow that enables to filter tweets, enrich tweet with meta-data, index tweets and build topical clusters. We present in what follows the main processing actions applied on tweets.

3.1.1 Real-time filtering

We apply a real-time stream processing in order to filter out non interesting tweets. The filtering process addresses the language of the tweet so as to consider only English tweets in this task. Moreover, we filter tweets including swear Words since we assume that it would not be appropriated to push notification containing adult vocabulary. We propose also to discard tweets that do not match tracking topics by indexing only those containing topics terms. Table 2 summarize applied filters on the tweet stream.

Filter	**Description**
English	Check *status.lang* field and remove non english tweets.
Swear Words	Remove tweets including one or more words qualified as swear Words or adult vocabulary.
Keywords	Consider only tweets including at least one term tracked through topic titles.

Table 2. Real-time filtering of the tweet stream

3.1.2 Real-time indexing

Tweets filling previous filter requirements are enriched with additional data and then indexed in real-time. More particularly, we include the title and the description of URLs attached in the tweet. We note that URLs are downloaded in real-time and. However, in order to limit the processing real-time load, we consider only plain text resources (typically HTML pages with at least title tag) and limited the range of the full HTML pages to the meta-data, title and description tags.

In addition to the text field of tweets, we enhance the tweet description by another textual field mentionning only effective words. We called this field *"words"*. In practice, we processed tweet text by removing personal nouns, common Twitter terms (RT, via, http, etc), common English adjectives, common English adverbs, and commons English prepositions. We assume that all of these entities are topic-independent and thus not helpful for further clustering process.

3.1.3 Real-time clustering

We propose to cluster in real-time incoming tweets into similar topics. Inspired by the approach of [8] for clustering top k tweets for each topic, we propose to cluster an incoming tweet in real-time regarding its topic with respect to topic of tweets belonging to cluster already built. More particularly, we compute the similarity between an incoming tweet and the whole set of tweets belonging to already formed clusters. The similarity between two tweets is computed based on the *"words"* field using the Dice coefficient similarity metric as it takes into account the length of tweet. In order to ensure a minimal level of similarity between two tweets as well as the reliability of a tweet assignation to a cluster, we consider two tweets as similar if their respective similarity score overpasses 0.6. In the end, the tweet is either assigned to the cluster involving its most similar tweet (under the constraint that the similarity value is higher than the threshold) or forms a new cluster of none of previously clustered tweets.

3.2 Notifications and digest filtering Scenario

We address in this work the task of real-time filtering of tweets as a periodic search task. In fact, a regular search and retrieval process over new tweets is triggered periodically at the end of a predefined time window. While executing our real-time filtering approach systematically, we stimulate push notifications on mobile phone scenario as well as periodic email digest scenario.

For pushing notifications on mobile phone, the time window is limited to 100 minutes. In respect of the track guidelines, the notifications is considered irrelevant if it is published beyond this time window. In this work, we propose to simply set the time window for push notifications scenario to *60 minutes*.

Since a periodic email digest scenario is devoted to send to users an extended summary of tweets, we assume that the time window may be larger. Therefore, we propose to set the time window for this scenario to *1 day*.

Let δt be the time window, corresponding to *60 minutes* for scenario A and *1 day* for scenario B. While tweets are clustered at the indexing timestamp and each tweet is assigned to a single cluster, we assume that S represents the set of tweet clusters already sent to a user for a topic q, regardless of the scenario type.

For each tweet published in the time window δt, we compute its relevance score in respect of the topic q using the BM25f scoring schema [9]. The relevance score of a tweet is computed based on 3 textual fields including the text of the tweet, the title of the URL and the description of the URL. Weights assigned to each of theses fields is presented in table 3.

In the next step, we removed tweets that respectively belong to clusters S already sent to users. Thus, non-novel tweet cluster willing to be sent to users are discarded. At this level, clusters involving only tweets not already seen by users remains, ensuring the novelty in the pushing and digest task. Among this set of novel clusters, we propose to keep simply the first published tweet as the most

Field	Wieght
tweet.text	4
url.title	1
url.description	1

Table 3. BM25F field weights for textual fields of tweets

representative one of the cluster. Someone may select the tweet with highest relevance score or the most close of cluster Centroid but we believe here that the time factor is a reasonable criterion due to the real-time notification task.

The previous steps allow to collect novel tweets for each time window. These tweets are ranked by their relevance score with respect to the query. A threshold is applied on the relevance score is applied in order to ensure a higher relevance among the results. In this work, we propose to set the threshold to the following value: $0.25 * max_{score}$ where max_{score} is the maximal relevance score of tweets of respective topic q. Finally, we apply a threshold on the number of tweets in accordance to track requirement. We limit the number of tweets to 10 per day for the push notification scenario and to 100 tweets for the periodic email digest scenario.

3.3 Results

We submitted one run for each scenario. Table 4 presents official results obtained by our run and comparison to the maximum of submitted run.

	Scenario A		Scenario B
	ELG	nCG	nDCG
IRIT-RTDig			0.1680
IRIT-RTNotif	0.1950	0.1834	
Max of submitted runs	0.4715	0.4943	0.5114

Table 4. Performance metric of Real time filtering based on rapidity and feature scores.

4 Tweet Selection based on speed and feature scores

Our approach focuses on the answer period: never going beyond the minute regardless of the number of incoming tweets. To achieve this main goal, we filtered tweets on several levels and dropped them as soon as possible. To increase performances, we added also several kinds of features, regarding the different kinds of information available inside tweets in addition to the textual content itself.

4.1 Content

We performed a conventional processing of the content, and every full text part of tweets (as the user description or hashtags, mentions, etc.). The processing comprises four steps: suppressing every non English word and stopwords, suppressing the case, tokenizing texts, and finally using a stemmer (Porter stemmer algorithm). To judge the relevance of the content, we proceed in two steps. First, a comparison between the content and the “title” part of user profile is done. The tweet is accepted if this match

ratio (i.e. shared terms) raises at least a threshold. After that, a similarity (by the cosine measure) is computed between tweet content and the title added to description. If the first step is not completed, the whole process stops and the tweet is rejected. Otherwise, it continued to the second one with the same way out, before the treatment of the features surrounding the tweet. Once again, the threshold defined for the cosine measure has been set up through experiment before the evaluation period.

4.2 Scoring Principle

When the tweet is selected, it went into the second main part of the system: the scoring step. In practical terms, we compute a score for each tweet, according to some features. At the end of this process, a global threshold had been set through experiences in order to determine if a tweet is finally selected. We present next the used features about content, entities (hashtags, mentions, etc...) and user who posted the tweet ([10]).

Before that, we added two features computed on the content of the tweet that we discussed above. They aimed to model the quality of the used language ([11] and [12]). Indeed, the first one is the ratio between the number of "real" words (after treatments mentioned in the first part) on the total number of words of the tweet content. We can model here the percentage of meaningful words ([10]). In the same idea, the second content feature is the relation between the number of hashtags and the full total of words. In addition to them, we add a third feature to complete them, where more than ten significant terms in the document.

4.2.1 Entities Features

In microblogs, and especially in tweets, information is not only contained in the text content. Indeed, a lot of complementary and useful data are stored in all the other fields of the tweet. Considering rapidity as a main factor, all the interesting fields cannot be used, so we had to make a choice. One of the particularity of tweets comes from the entities added to, or inside, the text. This is the most useful information that we can explore.

We select four entities in order to define seven features on them. We exploit hashtags by counting them first, and then by checking their presence, or not, within the queries ([13]). A very similar part of tweets, but carrying information about people or groups, organizations, events, etc., are Mentions. With their structure similar to hashtags, we process them the same way and create the two same features on them.

Furthermore, we consider URLs and Medias. To maintain a very fast process, we limited their use controlling their presence; and giving appropriate score. Indeed, it had already been shown that the simple presence of a URL in a tweet is a factor of quality ([14] and[15]). In practical terms, adding such elements the author wants to confirm, to justify, what he is saying. We extended this observation to medias, as a factor improving tweet relevance.

4.2.2 User Features

Microblogs, and in particular Twitter, are by essence a social network and therefore dependent of users, links between them, their activities, etc. Five features are directly considered from the tweets and compute a final one. We selected the most significant to model the author popularity and legitimacy. The two most classical at the social level are the number of followers and number of friends; in addition, the number of public lists that the author is member of fills our choice in. Finally, last features are based on the number of statuses, and the number of favourites.

For these five features, we set up a threshold for each by previous experiments: we aggregated one month of data and computed the third quartile for each; all tweets above this quartile obtained an

entity score. To finish this part, the last computed feature is the description field. As described above, we calculate the cosine similarity between that user description and the query in order to find a link between the tweet author and the targeted user profile.

In Table 5, we sum up all the features used for this Microblog Task. We expose clearly if they are obtained using a cosine measure or with a score. Their associated thresholds (above which the tweet is selected for this particular feature) or in some cases the presence or absence of such and such field, and finally their impact in the final model.

Features	Score	Cosine Similarity	Threshold	Significance
Content:				
NbRealWords	✓	✗	≥ 10	1
LangQuality	✗	✓	≥ 0.6	1
HashQuality	✗	✓	≥ 0.6	1
Entities:				
NbHash	✓	✗	≥ 1	1
HashSim	✗	✓	Yes/No	1 by hashtag
NbMention	✓	✗	≥ 1	1
MentionSim	✗	✓	Yes/No	1 by mention
PresUrl	✓	✗	Yes/No	1
UrlSim	✗	✓	≥ 0.1	2
PresMedia	✓	✗	Yes/No	1
User:				
FollowCount	✓	✗	≥ 945	2
StatusCount	✓	✗	≥ 27689	2
FriendsCount	✓	✗	≥ 759	2
ListedCount	✓	✗	≥ 7	1
FavourCount	✓	✗	≥ 3166	1
DescSim	✗	✓	≥ 0.1	1

Table 5. Features considered for the Microblog Task

4.3 Results

In this TREC 2015 Microblog Track, after the evaluation period, we submited one run for each scenario; the obtained results are shown in table 6.

Table 6. Performance metric of Real time filtering based on rapidity and feature scores.

	Scenario A		**Scenario B**
Metric / **Run**	ELG	nCG	nDCG
IritSigSG	0.2122	0.2043	0.1329
Max across all submitted runs	0.4715	0.4943	0.5114

Regardless the results, our main goal was achieved: the system returned selected tweets in one or two seconds (ten maximum during peak times) for scenario A and few minutes in scenario B. Future work will be devoted to an in-depth study of the thresholds chosen, and try to develop the feature system to connect them as much as possible with the context surrounding the tweet and its content itself.

5 Conclusion and future work

In this paper, we presented three approaches used in the TREC Microblog Track guided either by a select/ignore decision making, a real-time filtering/clustering or a rapidity-based stream selection. For all these approaches, we underline that further experiments are needed, more particularly in the parameter tuning steps. However, we believe that results are quite promising and could give interesting insights in the future in terms of real-time tweet indexing and retrieval, which are important components in the information access within data-streams.

References

1. Miles Efron. Information search and retrieval in microblogs. *J. Am. Soc. Inf. Sci. Technol.*, 62(6):996–1008, June 2011.
2. Ian Soboroff, Iadh Ounis, J Lin, and I Soboroff. Overview of the trec-2012 microblog track. In *Proceedings of TREC*, volume 2012, 2012.
3. Jimmy Lin and Miles Efron. Overview of the trec-2013 microblog track. In *Proceedings of TREC*, volume 2013, 2013.
4. Beaux P. Sharifi, David I. Inouye, and Jugal K. Kalita. Summarization of twitter microblogs. *The Computer Journal*, 57(3):378–402, 2014.
5. Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. Using temporal IDF for efficient novelty detection in text streams. *CoRR*, abs/1401.1456, 2014.
6. S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, (1):79–86, 03.
7. Arnout Verheij, Allard Kleijn, Flavius Frasincar, and Frederik Hogenboom. A comparison study for novelty control mechanisms applied to web news stories. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2012, Macau, China, December 4-7, 2012*, pages 431–436, 2012.
8. Maram Hasanain and Tamer Elsayed. Qu at trec-2014: Online clustering with temporal and topical expansion for tweet timeline generation. In *Proceedings of The Twenty-Third Text REtrieval Conference (TREC 2014)*. NIST, 2015.
9. Hugo Zaragoza, Nick Craswell, Michael J Taylor, Suchi Saria, and Stephen E Robertson. Microsoft cambridge at trec 13: Web and hard tracks. Citeseer.
10. Firas Damak, Karen Pinel-Sauvagnat, Guillaume Cabanac, and Mohand Boughanem. Effectiveness of State-of-the-art Features for Microblog Search (regular paper). In and, editor, *ACM Symposium on Applied Computing (SAC), Coimbra, Portugal, 22/03/2013-23/03/2013*, pages 914–919, http://www.acm.org/, mars 2013. ACM.
11. Fuxing Cheng, Xin Zhang, Ben He, Tiejian Luo, and Wenjie Wang. A survey of learning to rank for real-time twitter search. In *Pervasive computing and the networked world*, pages 150–164. Springer, 2013.
12. Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.
13. Donald Metzler and Congxing Cai. Usc/isi at trec 2011: Microblog track. In *TREC*. Citeseer, 2011.
14. Lulin Zhao, Yi Zeng, and Ning Zhong. A weighted multi-factor algorithm for microblog search. In *Active Media Technology*, pages 153–161. Springer, 2011.
15. Rinkesh Nagmoti, Ankur Teredesai, and Martine De Cock. Ranking approaches for microblog search. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 153–157. IEEE, 2010.