IRIT at TREC Real Time Summarization 2016

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

IRIT at TREC Real Time Summarization 2016

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

{ moulahi,jabeur, abdelhamid.chellal, palmer, tamine, boughanem, sauvagnat, 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 Real Time Summarization track of TREC 2016. This track consists in a real-time filtering the tweet stream and identifying both relevant and novel tweets to be pushed to user in real-time. Our team proposes three different approaches: *(1)* The first approach consist of a filtering model that combines several summarization constraints *(2)* The second approach for the scenario A is composed of three filters adjusted sequentially in which we use word similarity based function to evaluate the relevance of an incoming tweet. The generation of a batch of up to 100 ranked tweets is formulate as an optimization problem. *(3)* The third approach consist of 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. We describe in this paper the three proposed approaches and we discuss official obtained results for each of them.

Keywords: real-time, social media, user profile, word similarity, filtering, clustering, rapidity, entities, personalization

1 Introduction

Social media streams provide real time updates that cover scheduled and unscheduled events which makes them a valuable source of information for user who wishes to receive timely notification to keep up-to-date on topic of interest. Indeed of the volume and the redundancy of the posted information in the social media stream, one of the main challenge consists in the fact that to be effective, notifications must achieve a balance between pushing too much and pushing too little updates. Push too little and the user misses important updates; push too much and the user is overwhelmed by irrelevant/redundant information [1]. Although several models have been proposed in the context of ad-hoc tweet search [2, 3], the task of prospective notification in tweet stream, which is proposed by the real time summarization Track of TREC 2016, is still under-explored.

The Microblog real time summarization Track aim 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 keep up-to-date on the topic of interest. In this aim, the track is split into two main scenarios:

1. The Scenario *A*, called "*Push notifications*", consists in an instantly tweet notification assuming a short time period between the tweet publication and the tweet pushing. Participating system are allowed to push up to 10 notifications per day per interest profile.
2. The scenario *B*, called "*Periodic email digest*" consist on daily selecting up to 100 tweets for each interest profile to be send to user via email. It's required that tweets should be relevant and novel but timeliness is not important.

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:

– Filtering model that decompose the filtering task to several sub-tasks in accordance to the summarization constraints. The final filtering score is aggregated as the product of scores obtained by sub-filtering functions.
– Real time filtering approach composed of three filters adjusted in sequential way which are related to topicality, relevance and novelty respectively. The decision to push/ignore a tweet is made immediately. The main contribution of this approach is we propose an adaptation of the extend Boolean model based on word similarity to evaluate the relevance score of the incoming tweet with respects to the topic. Indeed, we formulate tweet summarization problem as an optimization problem to general a email digest summary for the scenario B.
– A step by step stream selection that focuses on rapidity and that takes 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 streaming filter model for real-time summarization. Section 3 describes word similarity based approach for Real time tweet summarization. Section 4 presents the tweet selection approach based on speed and feature scores. Section 5 concludes the paper.

2 Streaming filter model for real-time summarization

Real-time summarization of tweets consist in selecting from a continuous stream of tweets $T =< t_1, t_2, \ldots, t_n >$ the set of relevant ones with respect of the tracked topic q. The result summary $S =< t_1, t_2, \ldots, t_m >$ must not include redundant tweets and must respect a length constraint in terms of the number of tweets. This problem can be viewed as a filtering task where filtering function $F(t_i)$ is applied to the incoming tweet t_i in order to decide if the tweet must be included in summary $F(t_i) = 1$ or neglected $F(t_i) = 0$. We propose here a filtering model that combines several summarization constraints that must be verified, namely the tweet quality, the topical relevance and the information redundancy.

2.1 Streaming filter model

Tweets to be included in the summary must respect several constraints, for instance, the summary length, the topical relevance and the non-redundancy of information with regards to past included tweets. Hence, we propose to decompose the filtering task into sub-filtering tasks where each verifies that the tweet respect a particular constraint. A sub-filtering functions $F_k(t_i)$ is defined in accordance to the k^{th} summarization constraint with $F_k(t_i) = 1$ allows to include tweet t_i in the summary or $F_k(t_i) = 0$ otherwise. The global filtering function $F(t_i)$ is computed as the product these functions requiring that tweet t_i must verify all constraints. $F(t_i)$ is computed as the following.

$$F(t_i) = \prod_{\forall k} F_k(t_i) \tag{1}$$

Table 1 lists constraints that we suggest to consider for summarizing real-time stream. Further constraints could be added to this list in order to satisfy advanced users preferences. A detailed description of computation of each filtering function is defined in the next section.

2.2 Computing filtering scores

We details in the section the computation of filtering functions introduced in table 1.

Table 1. Filtering functions respective Summary constraints

Function	Constraint	Description
$F_0(t_i)$	Summary length	Summary must be limited to l tweets
$F_1(t_i)$	Language preference	Tweet t_i must be in user's language
$F_2(t_i)$	Lexical restriction	Tweet t_i must not include "bad words"
$F_3(t_i)$	URL presence	Tweet t_i must include an URL
$F_4(t_i)$	Topical relevance	Tweet t_i must be relevant to tracked topic
$F_5(t_i)$	Non redundancy	Tweet t_i must not be redundant

Summary length. The length constraint suggest that the length of the summary must not overpass l. As proposed in summarization scenarios A and B, the length constraints is defined for a limited time window (i.e. day). The length constraint is set to $l = 10$ and $l = 100$ for scenario A and scenario B, respectively. Let θ_i be the timestamp of tweet t_i and $d(\theta_i)$ the day in witch t_i is published. The current daily summary is defined by the subset of tweets $S_{d(\theta_i)} = \{t_j \in S | d(\theta_j) = d(\theta_i)\}$. Accordingly, we define the filtering function $F_0(t_i)$ in respect of summary length as the following:

$$F_0(t_i) = \begin{cases} 1, & |S_{d(\theta_i)}| < l \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Language preference. It is interesting to push tweets in user's language. Tweets in other languages are useless unless they are translated. For this aim we propose to compute a language filtering score $F_1(t_i)$ as the following.

$$F_1(t_i) = \begin{cases} 1, & \text{lang}(t_i) \in G \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

With $\text{lang}(t_i)$ is the language of the tweet and G is a set of language preferences for the user. In the context of this track, we consider tthat $G = \{en\}$ since only English tweet are taken into consideration.

Lexical restriction. In order to ensure a high quality of tweets, we propose to apply a lexical filter that eliminates tweet containing "bad words". Let $L = \{w_1, w_2, ..., w_p\}$ be the lexicon of undesirable words. The lexical filtering score $F_2(t_i)$ is computed as the following:

$$F_2(t_i) = \begin{cases} 1, & |t_i \cup L| = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The lexical filter may be extended to include other types of lexicon in addition to "bad words" such as "power words" (e.g. "free", "easy" and "best") which are used for catching attention and emotionally impacting the reader.

URL presence. Experiments results on previous TREC microblog track [4] show that presence of URL in the tweet is a good indicator of its relevance. Accordingly we propose a strict filtering constraint that suggest to filter out tweets that do not contain any URL.

$$F_3(t_i) = \begin{cases} 1, & \text{urls}(t_i) \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

With $\text{urls}(t_i)$ is the number of URLS in the tweet.

Topical relevance. Notified tweet may be relevant with regards to interest profile. In order to determine the relevance of the tweet we adopt a strict policy suggesting that a significant number of terms from tracking topic must be included in the tweet. In fact, tracking topic are different from regular search query in terms of motivation since user have an exact idea about what she is looking for and carefully choose a tracking query that target his need. Hence, we propose that tweet must contain at least α terms. If $\alpha > |q|$, all topic terms must be present in the tweet. The topical filtering score $F_4(t_i)$is defined as the following:

$$F_4(t_i) = \begin{cases} 1, & |q \cup t_i| \geq min(\alpha, |q|) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

with $|q \cup t_i|$ is the number of distinct common terms between the topic and the tweet. We propose to set up parameter α to arbitrary value $\alpha = 3$. This value may be adjusted by futher experiments.

Non redundancy. Pushed tweets to notification may be not redundant and deliver a new information to user at each time. Otherwise, redundant notifications may cause user fatigue regardless if there are relevant or not. For this aim, we propose to apply a redundancy filter that eliminates tweet containing repeated information to what have been previously notified. In particular, we propose to check the similarity of incoming tweet with recently pushed ones across the past time window Δt.

Let $Q =< t_1, t_1, \ldots, t_q >$ a timed-queue of previously pushed tweet. All tweets in Q belong to the last window Δt verifying so the condition $\theta_i <= \theta_{now} - \Delta t$, $\forall t_i \in Q$ with θ_{now} is the actual time. We propose to compute a redundancy score of incoming tweet as the maximum similarity score to tweets in queue Q. In this context, we define the similarity between two tweets t_i and t_j as the proportion of common terms while taking into account the length of reference tweet t_i. The redundancy score, noted $r(t_i, Q)$, is computed as follows.

$$r(t_i, Q) = \underset{t_j \in Q}{\text{argmax}} \frac{|t_i \cup t_j|}{|t_i|} \tag{7}$$

With $|t_i|$ is the number of distinct terms in t_i and $|t_i \cup t_j|$ is the number of distinct common terms between t_i and t_j . We note that normalizing the number of common terms over the length of incoming tweet t_i in the similarity quotient $\frac{|t_i \cup t_j|}{|t_i|}$ instead of using the minimum length of the two tweets as in overlap coefficient $\frac{|t_i \cup t_j|}{min(|t_i|, |t_j|)}$ reduces the fallacy of too short tweets in queue Q in terms of redundancy towards longer novel tweets.

In order to compute the redundancy filtering score, we propose to compare the redundancy score $r(t_i, Q)$ of t_i to a fixed threshold. Tweet t_i will be filtered out if respective redundancy score overpass β. The filtering redundancy score $F_5(t_i)$ is defined as follows.

$$F_5(t_i) = \begin{cases} 1, & r(t_i, Q) < \beta \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

In this experiments we set up β parameter to arbitrary value $\beta = 0,6$ which represent a strict threshold for eliminating redundant tweet. The value of β could be configured experimentally as the average overlap similarity between the relevant tweets of each topic from past released datasets of TREC microblogs. Besides, we set up the time window Δt parameter of the timed-queue Q to arbitrary value of $\Delta t = 1day$ requiring that user must not receive a redundant tweet in the same day. This value is reasonable as most news editor refresh their content daily. Increasing this time window to several days may cause over-filtering of novel tweets that may use partial vocabulary from past tweets in summary but announce a novel information.

2.3 Results

As our filtering model is compatible for both scenarios A and B, we submitted the same generated summary for both of them. For scenario A, tweet are pushed in real time to the evaluation API. The same tweets are used to make our run for scenario B while ranking them based on their recency. We notice that we include only 10 tweet per daily digest summary for scenario B although 100 tweets are allowed. We discuss in what follows obtained results for scenario A and scenario B

Table 2. Tweet counting for the two summaries generate by our system (run IRIT-iritRunBiAm-21) and baseline.

	#rel	#redundant	#non-rel	#unjudged	#total-length	precision
Our system	201	6	323	1674	2177	0.09
Baseline	148	12	286	1461	1888	0.07

scenario A. Table 2 compares the number of relevant, redundant tweet, non relevant tweet, judged tweet and submitted tweets the for our system and track baseline. First, we note that a large part of submitted tweets by our system (76%) are unjudged and the same for the baseline (77%). Having the similar proportions of judged tweets, the comparison must be fair but deeper pooling for relevance judgment is required to confirm this analysis. Besides, our system is able to deliver more relevant tweets (201) compared to the baseline (148). We report in the last column of the table the precision of relevant tweet out of the summary length. Precision values show that our system slightly overpass the baseline with overall precision of 0,09 compared to 0,07. We notice that The precision values for both systems are considerably low and this is due to large number of unjudged tweets considered as irrelevant.

Table 3. Official results of our system (run IRIT-iritRunBiAm-21) and baseline for Scenario A.

	EG1	EG0	nCG1	nCG0	GMP.33	GMP.5	GMP.66	mean latency	median latency
Our system	0.2493	0.0332	0.2541	0.0380	-0.5464	-0.3817	-0.2267	102630.1	23.0
Baseline	0.2289	0.0253	0.2330	0.0295	-0.6000	-0.4317	-0.2733	120908.6	8718.0

Table 3 compares values of official measures obtained by our system and the track baseline. We notice that in contrast of EG1 and nCG1 , the EGO and nCG0 ignore silent days where no relevant document is published so participating systems receive equal gain. Our system overpasses values obtained by baseline for the four measures EG1, EGO, nCG1 and nCG0. This highlights that our filtering model is more effective for summarization in terms of the relevance of included tweets as well as recognition that there are no relevant tweets to push.

Table 3 present also obtained results for Gain Minus Pain (GMP) measure with different configurations GMP.33, GMP.5 GMP.66. Our systems present negative values for All the the three configurations. This highlights however that our system generates summaries with more irrelevant tweet than relevant ones. Despite these negative results, our system show always higher GMP values compared to the baseline. Finally and comparing the latency values, we note that our system and baseline show near mean values while our system present a very low median values compared to the baseline. We conclude that our model is more active. It pushs relevant tweets to summary in too short time.

scenario B. Table 4 show nDCG1 and nDCG0 values obtained by our system and track baseline. We remember that these measures are computed as the average of nDCG@10 scores for each day and topic. In contrast of nDCG1, the nDCG0 discard silent days so participating systems receive equal gain. As shown in the table 5. the performances of our system overpass the baseline for nDCG1 with an improvement of 5% for nDCG1 and 7% for nDCG0 . However, this improvement do not imply stable effectiveness of our model due the low values for nDCG0. A further analysis on our submitted run shows that the good performances our system for nDCG1 measure are explained due to its "timid" behaviour in contrast of "verbose" systems continuously sending tweets every day . In fact, Our system submit only 102 out of 560 (18%) expected daily digest emails for all topics confused where 387 are actually not salient. Besides, results for nDCG0 which also evaluate tweet ranking process show that our ranking strategy that suggest to order tweets according to their timestamp have limited performances.

Table 4. Official results of our system (run RunBIch) and baseline for Scenario B.

	nDCG1	nDCG0
Our run	0,2481	0,0321
baseline	0.2352 (+5%)	0.0299 (+7%)

3 Word similarity based approach for Real time tweet summarization

3.1 Relevance score

The main drawback of the use of statistical weighting techniques is that statistics change when a new tweet arrives leading to update the index every time which is a challenging task regarding the velocity in the tweet stream. In addition, given the short length of tweets the statistical features appear to have limited value. We believe that the similarity between the tweet words and topic words is the key feature to estimate the relevance score of incoming tweet. Hence, we propose to use adapt the extended Boolean model [5] to evaluate the relevance score of tweet. We propose to consider the similarity score between topic title words and tweet words to evaluate the weight of topic words instead of using the TF-IDF weighting technique.

Topics provided in TREC RTS 2016 task include a title and a description of the information need. While, participants were permitted to use these two filed for filtering, we use in our participation the title field to estimate the relevance score of the tweet with respect to a topic. The topic title QT is considered as "**AND**ed terms". In the Extended Boolean model, the relevance score of tweet $T = \{t_1, ..., t_n\}$ to "And query" QT is estimated as follows:

$$RSV(T, QT_{and}) = 1 - \sqrt{\frac{\sum_{qt_i \in QT}(1 - W_T(qt_i))^2}{|QT|}} \tag{9}$$

Where $W_T(qt_i)$ is the weight of the query term qt_i in the tweet T.

Instead of using TF-IDF weighting technique, we propose to estimate the weight $W_T(qt_i)$ by evaluating the similarity between the query term qt_i and all the terms of tweet T, which is measured by cosine similarity between their word2vec vector [6] as follows:

$$W_T(qt_i) = \max_{t_j \in T}[w2vsim(t_j, qt_i)] \tag{10}$$

Where $w2vsim(t_j, qt_i)$ is the similarity between tweet word t_j and query word qt_i. It is measured by cosine similarity between their vector which are generated by word2vec model [6] using a training

tweet stream collection. The intuition behind this proposition is that tweets that have words sharing many contexts with the query words will be more relevant.

3.2 Novelty score

Novelty can be evaluated by conducting a pairwise comparison between incoming tweet and those already selected in the summary using cosine similarity and Kl-divergence. However due to the shortness of tweet, meaningful words rarely occur more than once which implies that cosine similarity and Kl-divergence are not suitable for evaluating the distance between two tweets. Indeed, a pairwise comparison dose not fit real time filtering (particularly for the scenario A). Hence, in this run we use the novelty estimation proposed in [7] in which all summary's tweets are merge into one "summary word set" and word overlap is used to evaluate the novelty score of the incoming tweet. Assume that *RW* is the set of words that occur in current summary then the novelty score of the incoming tweet is evaluated as follows:

$$NS(T, RW) = 1 - \frac{|RW \cap T|}{|T|} \tag{11}$$

3.3 Run for Scenario: A Real Time summarization

For the scenario A, our approach is composed of three filters related to the topicality, relevance and novelty. These filters are adjusted in sequential way. To reduce the latency between tweet creation time and tweet notification time, the decision of pushing/ignoring an incoming tweet is made immediately as soon as the tweet occurs in the stream. A tweet passes a filter if its novelty score is above a certain threshold.

The first filter eliminates non-English tweets and those containing less than three tokens. It also drops all an incoming tweets that do not contain a predefined number of title words. An incoming tweet is considered related to a topic if and only if its number of words overlap with the query title is higher than a minimum of either a predefined constant (K) or the length of the query title ($min(K, |QT|)$). Based on pilot experiments on TREC MB RTF 2015 data-set [4], we set the value of the threshold k to $k = 2$.

In the second filter, the relevance score of incoming tweet that passes the first is estimated. A tweet pass to the next filter if its score is above the following threshold:

$$Rel\_Threshold = MAX(0.4, 1 - \sqrt{\frac{2}{|QT|}}) \tag{12}$$

In the third filter, a novelty score of a tweet is estimated as described in the equation 11. Tweets with novelty score less than 0,6 were discarded. We based the novelty threshold on pilot experiments conducted on TREC MB RTF 2015 date set.

3.4 Run for Scenario B: Retrospective notification

To generate run for scenario B, we formulate tweet summarization problem as an optimization problem based on the tweet relevance function defined in the equation 12. we use Integer Linear Programming to select tweets that optimize a global objective function. To solve the problem, we use the GNU Linear Programming kit footnotehttps://www.gnu.org/software/glpk/, which is a free optimization package.

More specifically, we would like to select from M candidate tweets (those that pass the topicality filter) N tweets which receive the highest relevance score with respect to the user interest subject to a series of constraints related to redundancy and length limit (maximum number of tweets allowed in the summary). The tweet summarization problem can be formulated as the following ILP problem:

We include indicators variable X_i which is 1 when tweet T_i is added in a summary and 0 otherwise. Notice here that the first constraint simply states that the indicator variables are binary.

$$\forall i \in [1,M], X_i \in \{0,1\}$$

Objective function: Top ranked tweets are supposed to be the most relevant tweets which we want to include in the final summary. Thus, we are looking for maximizing the global relevance score of selected tweets to improve the overall coverage and relevance of the final summary. The objective function is defined as follows:

$$\max(\sum_{i=1}^{M} X_i \times RSV(T_i, Q))$$

Redundancy Constraints: To prevent redundancy, we compute a pairwise similarity between tweets and if the similarity is above a certain threshold (Sim_threshold) then we drop the tweet that have a lower relevance score. This constraint is formulated as follows:

$$\forall i, j \in [1,M], (X_i + X_j) \times sim(T_i, T_j) \leq 2 \times Sim\_threshold$$

The similarity between two tweets is computed based on the word2vec similarity function as it takes into account the semantic relation between two words as follows:

$$Sim(T, T') = \frac{1}{|T|} \sum_{t_i \in T} \max_{t_j \in T'} w2vsim(t_i, t_j) \tag{13}$$

The similarity threshold used in the redundancy constraints is set to 0.75, which means that to be added to the summary, a tweet has to have a similarity score higher than 0.75 with all other selected tweets.

Length Constraints: We add this constraint to ensure that the length of the final summary is limited to the minimum of either a 100 (the maximum allowed number of pushed tweets) or $M-1$ where M is the number of candidates tweets (those that pass the topicality filter).

$$\sum_{i=1}^{C} \sum_{j=1}^{L_i} X_{ij} \leqslant min(N, M-1)$$

3.5 Results

For Scenario A, Table 5 reports the performance of our submitted run based on word similarity extended Boolean model (WSEBM). In terms of EG1 and nCG1 our approach dose not outperform the organizer baseline. These results reveals that our strategy of setting the relevance threshold is not effective. In fact, in terms of EG1 and nCG1 systems achieved high scores by simply pushing few tweets. In these metric system that push zero tweet for sailing day receive score of 1 for that day and the system receive score of 0 otherwise. Unfortunately, in our run we used a threshold that is not restrictive enough. Hence, 6887 tweets were pushed and among them only 1417 tweets were judged. Our run contains 354 relevant which represents 24.98% of pushed tweets. These results can also be partially explain by the fact that only 20.57% of pushed tweets were judged by assessors. In terms of EG0 and nCG0 our approach outperforms the baseline which shows that our approach perform well in the case of "eventful days" where there are relevant tweets in the stream. The results for Scenario A also indicates that the latency of our approach is better than the latency of the baseline.

Table 6 reports our results for Scenario B, which aggregated the the user's interest in a daily summary with a maximum of 100 tweets. in terms of nDCG1 in which pushing tweet for salient day is penalizing the performance of our run is bellow the baseline but the difference is not significant. However, in terms of nDCG1 our run outperforms the baseline the improvement is to 422.07%. These results reveals that our approach achieve a good balance between pushing too much tweets and pushing too little tweets.

Table 5. Results of the real-time summarisation task for scenario A.

	EG1	**EG0**	**nCG1**	**nCG0**	GMP.33	**GMP.5**	**GMP.66**	**mean latency**	**median latency**
IRIT-WSEBM	0.1224	0.0402	0.1916	0.1095	-2.5321	-1.8348	-1.1785	112089.1	74.5
Baseline	0.2289	0.0253	0.2330	0.0295	-0.6000	-0.4317	-0.2733	120908.6	8718.0

Table 6. Results of the real-time summarisation task for scenario B.

	nDCG1	**nDCG0**
IRIT-ILPWSEBM	0.2208	0.1262
TREC Baseline	0.2352	0.0299

4 Retweet Recommendation Using Contextual Features

Our approach aims to perform an automatic retweet recommendation with a model that automatically relays a selective set of relevant tweets from data streams given an interest profile. This model relies on contextual features that are associated with every tweet, combined to content processing. This model focuses on respecting time constraints for processing and keeps prompt retweet.

4.1 Context-based Model

Our context-based model relies on content and context evaluations, which serve as the basis for a retweet decision function. These model constituents are described in the following sections.

4.1.1 Content Evaluation

First, each profile p and each tweet t_i associated with content $\mathscr{CO}_{t_i}$ receive the same text processing, in order to maximize the possible matchings. After pre-processing we obtain a profile representation $p' = \{w'^{p}_{0}, ..., w'^{p}_{k'}\}$ composed of $k' \leq k$ terms w'^{p}_{j} ; and a pre-processed tweet content $\mathscr{CO}'_{t_i} = \{w'^{t_i}_{0}, ..., w'^{t_i}_{n'}\}$ composed of $n' \leq n$ terms $w'^{t_i}_{j}$. The aim of the content model is to evaluate tweet relevance with respect to the interest profile, i.e., to match $\mathscr{CO}'_{t_i}$ with p'. We evaluate a content acceptance criterion (*CAC*) as:

$$CAC(p', \mathscr{CO}'_{t_i}) = \frac{\sum_{j=1}^{n'} \mathbb{1}_{\{w'^{t_i}_{j} \in p'\}}}{\sum_{j=1}^{k'} \mathbb{1}_{\{w'^{p}_{j} \in p'\}}} \tag{14}$$

where $\mathbb{1}$ is an indicator function which is equal to 1 if the attached condition is validated and 0 otherwise (e.g., $\mathbb{1}_{\{x \in \mathbb{R}\}} = 1$; if $x \in \mathbb{R}$).

Concretely, we evaluate the ratio of common terms between the tweet content $\mathscr{CO}'_{t_i}$ and the interest profile p', by the number of terms in p'.

4.1.2 Context Evaluation

In addition to content, we associate each tweet with a context $\mathscr{CX}_{t_i}$ composed of additional information. These information are either extracted or calculated from tweet metadata $\mathscr{M}_{t_i}$. $\mathscr{M}_{t_i}$ is defined as $\{st_{t_i}, hash_{t_i}, men_{t_i}, url_{t_i}, med_{t_i}, us_{t_i}\}$ where :

- st_{t_i} is the tweet status (initial tweet or retweet of another user),
- $hash_{t_i}$ is the set of hashtags in t_i,

– men_{t_i} is the set of mentions in t_i,

– url_{t_i} is the set of urls in t_i,

– med_{t_i} is the set of medias (image, sound, video, etc.) in t_i,

– us_{t_i} is the set of information about the author of t_i. $us_{t_i} = \{fol_{t_i}, stat_{t_i}, fr_{t_i}, list_{t_i}, fav_{t_i}, desc\}$ where fol_{t_i} is the number of followers, $stat_{t_i}$ is the number of status (number of tweets and retweets issued by the author), fr_{t_i} is the number of friends, $list_{t_i}$ is the number of public lists the author is member of, fav_{t_i} is the number of favourites, and $desc_{t_i}$ is the author profile description composed of *nDesc* terms $w^d{}_j$.

$\mathscr{CX}_{t_i}$ is defined as a set of features f_l, whose values are evaluated using $\mathscr{M}_{t_i}$. Each feature f_l is associated to a threshold λ_l, over which a tweet is considered relevant regarding the feature. We distinguish two types of features, major and minor ones, regarding their influence according to the state of the art. Table 7 sums up all of the features f_l with their associated thresholds λ_l, as well as their significance set (Major or Minor). f_9 is for instance categorized as a major feature, since [8] showed that the presence of an URL in tweets is an indicator of relevance. λ_l may be evaluated using a set of resources $\mathscr{R}$ detailed in next section. Features are organized into three categories : Content, Entities, and Author.

Table 7. Features considered for the context evaluation. Acquisition way is either M (metadata extraction) or C (calculated). Threshold determination ways are $*$ for statistic method using $\mathscr{R}$, † for pilot study method, or ‡ for defined in the literature. Description references the related work when appropriate.

	Feature f_l	Acquisition	Threshold λ_l	Significance	Description
Content	f_1	M	0 ‡	Major	Initial tweet or retweet of another user. 1 if initial tweet ; 0 otherwise [9]
	f_2	C	10 †	Minor	Number of terms after text pre-processing (n')
	f_3	C	0.6 †	Minor	Ratio between n' and n (number of terms before pre-processing) [9]
	f_4	C	0.6 †	Minor	Ratio between the size of $hash_{t_i}$ and n
Entities	f_5	C	1 †	Minor	size of $hash_{t_i}$. Derived from [10]
	f_6	C	0 ‡	Minor	For h in $hash_{t_i}$ presence of h in profile p
	f_7	C	0 †	Minor	Size of men_{t_i}. Derived from [10]
	f_8	C	0 †	Minor	For m in men_{t_i} presence of m in p
	f_9	M	0 ‡	Major	Size of url_{t_i}. Derived from [8]
	f_{10}	M	0 †	Minor	Size of med_{t_i}
Author	f_{11}	M	945 $*$	Major	fol_{t_i}. Derived from [10]
	f_{12}	M	27689 $*$	Major	$stat_{t_i}$.[9], and derived from [10]
	f_{13}	M	759 $*$	Major	fr_{t_i}. Derived from [10]
	f_{14}	M	7 $*$	Minor	$list_{t_i}$
	f_{15}	M	3166 $*$	Minor	fav_{t_i}
	f_{16}	C	0.3 †	Minor	Cosine similarity between author description $desc_{t_i}$ and p

For each feature f_l we evaluate a score $Sc_l(f_l, p, \mathscr{R})$ as follows:

$$Sc_l(f_l, p, \mathscr{R}) = \begin{cases} \sum_{y=1}^{|hash_{t_i}|} \mathbb{1}_{\{f_l > \lambda_l\}} & \text{if } l = 6 \\ \sum_{y=1}^{|men_{t_i}|} \mathbb{1}_{\{f_l > \lambda_l\}} & \text{if } l = 8 \\ \mathbb{1}_{\{f_l > \lambda_l\}} & \text{otherwise} \end{cases} \tag{15}$$

Each feature is associated to a score equal to 1 if its value is greater than λ_l. The only particularity comes from f_6, respectively f_8, which score is summed up for each hashtag, respectively mention, if present in interest profile p.

A feature-based score (FBS) is then evaluated for the tweet as:

$$FBS(\mathscr{C}\mathscr{X}_{t_i}, p, \mathscr{R}) = \sum_{l=1}^{m} \left[(Sc_l(f_l, p, \mathscr{R}) \,.\, \mathbb{1}_{\{l \in \mathscr{C}\mathscr{X}^m_{t_i}\}}) + 2\,(Sc_l(f_l, p, \mathscr{R}) \,.\, \mathbb{1}_{\{l \in \mathscr{C}\mathscr{X}^M_{t_i}\}}) \right] \tag{16}$$

where $\mathscr{C}\mathscr{X}^m_{t_i}$ is the set of minor features and $\mathscr{C}\mathscr{X}^M_{t_i}$ is the set of major ones ; with $\mathscr{C}\mathscr{X}_{t_i} = \mathscr{C}\mathscr{X}^M_{t_i} \cup \mathscr{C}\mathscr{X}^m_{t_i}$ and $\mathscr{C}\mathscr{X}^M_{t_i} \cap \mathscr{C}\mathscr{X}^m_{t_i} = \emptyset$.

4.1.3 Context Evaluation

The retweet decision function f, which relies on the aforementioned content and context evaluations, is defined as follows:

$$f(t_i, p, \mathscr{T}_s, \mathscr{R}) = \begin{cases} \{rt_i\} & \text{if } CAC(p', \mathscr{C}\mathscr{O}'_{t_i}, \mathscr{T}_s) > \lambda_{\mathscr{C}\mathscr{O}} \text{ and } FBS(\mathscr{C}\mathscr{X}_{t_i}, p, \mathscr{R}) > \lambda_{\mathscr{C}\mathscr{X}} \\ \emptyset & \text{otherwise} \end{cases} \tag{17}$$

where $\lambda_{\mathscr{C}\mathscr{O}}$ and $\lambda_{\mathscr{C}\mathscr{X}}$ are thresholds associated with CAC and FBS. A tweet is thus retweeted if its content and context are considered as relevant enough according to the user profile p.

4.2 Model implementation

We focused on scenario A even though responded to scenario B. For the track, a usual pre-processing is applied on profiles p and tweet contents $\mathscr{C}\mathscr{O}_{t_i}$ consisting in stopword removal, lower case standardization, and Porter stemming [11]. A language detection is also performed on tweets to keep only those written in English.

4.2.1 Scenario A

For our context model, the set of resources $\mathscr{R}$ used to evaluate $\mathscr{C}\mathscr{X}_{t_i}$ was composed of Twitter data collected last year. $\mathscr{R}$ was used to fixed the thresholds λ_{11} to λ_{15} associated to features f_{11} to f_{15} by computing the third quartile of the features. The thresholds for the remaining features were manually determined by observing feature distributions or according to the state of the art (see Table 7).

Regarding the two global thresholds $\lambda_{\mathscr{C}\mathscr{O}}$ and $\lambda_{\mathscr{C}\mathscr{X}}$, associated to CAC and FBS, we implemented the model with best pairs of values ($\lambda_{\mathscr{C}\mathscr{O}}$, $\lambda_{\mathscr{C}\mathscr{X}}$) computed during tests on the TREC Microblogs 2015 collection. We fixed $\lambda_{\mathscr{C}\mathscr{X}}$ to 5 (to eliminate tweets with poor contexts), and $\lambda_{\mathscr{C}\mathscr{O}}$ to 0.6 (to keep high related tweets). Finally, we added the imposed limitation of ten notifications per day.

The major differences regarding our last year participation ([12]) resides in considering retweets (i.e., adding a feature related to the tweet status), reducing processing time, removing "query expansion" treatments on *description* query part, and improving the content evaluation (regarding profile matching).

4.2.2 Scenario B

We used the same implementation of our model as for scenario A, except that $\lambda_{\mathscr{C}\mathscr{O}}$ which was set to 0.3 in order to accept more related tweets in the system. The main difference lies in the additional step after the context model: ranking (the 100 best) results for each profiles at the end of the day. We compute a global score $GCFS$ gathering both content and context scores (CAC and normalized FBS) as:

$$GCFS(p, p', \mathscr{C}\mathscr{O}'_{t_i}, \mathscr{C}\mathscr{X}_{t_i}, \mathscr{R}) = CAC + FBS \ . \ 0.02 \tag{18}$$

At the end we ordered tweets for each profile according to $GCFS$.

4.3 Results

In this TREC 2016 Real-Time Summarization Track, we submitted one run for each scenario; the obtained results are shown in Table 8 for counting and Table 9 for official results in scenario A, and in Table 10 for scenario B. Details about metrics are in official Guidelines.

Table 8. Tweet counting generate by our system (run IritIris-SDA-22) and baseline.

	#rel	#redundant	#non-rel	#unjudged	#total-length
Our system	136	17	448	1467	2060
Baseline	148	12	286	1461	1888

Table 9. Official results of our system (run IritIris-SDA-22) and baseline for Scenario A.

	EG1	EG0	nCG1	nCG0	GMP.33	GMP.5	GMP.66	mean latency	median latency
Our system	0.2181	0.0270	0.0270	0.0406	-1.1275	-0.8161	-0.5229	123012.6	13047
Baseline	0.2289	0.0253	0.2330	0.0295	-0.6000	-0.4317	-0.2733	120908.6	8718.0

Table 10. Official results of our system (run IritIris-SDB) and baseline for Scenario B.

	nDCG1	nDCG0
Our system	0.1062	0.0651
Baseline	0.2352	0.029

Regardless the results, our main goal was achieved: the system returned selected tweets in five seconds maximum during peak times for scenario A and one hour for scenario B. Unfortunately, this response time was not taking into account into basic metrics and we obtained high latency measures due to earlier tweets belonging to the same evaluation cluster. Future work will be devoted to an in-depth study of the context features, their impact on the global model and interactions between them. We will add profile classification to fix thresholds according to these different groups.

5 Conclusion and future work

We presented in this paper three different approaches for real time summarization of tweet stream. The proposed approaches compute either a relevance score or filtering score which allow to determine if a new tweet should be included in the summary. We have have compared official results with those obtained by the track baseline. We believe that these results are quite promising but should be however compared with further experiments, more particularly to study the impact of some tuning parameter on the effectiveness of proposed models.

References

1. Charles L. A. Clarke Jimmy Lin Luchen Tan, Adam Roegiest. Simple dynamic emission strategies for microblog filtering. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, 2016.
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. Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, Richard McCreadie, and Tetsuya Sakai. Overview of the trec 2015 microblog track. In *Text REtrieval Conference, TREC, Gaithersburg, USA, November 17-20*, 2015.
5. Gerard Salton, Edward A. Fox, and Harry Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036, November 1983.
6. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
7. Abdelhamid Chellal, Mohand Boughanem, and Bernard Dousset. Multi-criterion real time tweet summarization based upon adaptive threshold. In *2016 IEEE/WIC/ACM International Conferences on Web Intelligence (WI16, Omaha, Nebraska USA, October 13-16, 2016*, pages 264–271, 2016.
8. Firas Damak, Karen Pinel-Sauvagnat, Mohand Boughanem, and Guillaume Cabanac. Effectiveness of state-of-the-art features for microblog search. In *Proceedings of Int. Conf. SAC'13*, pages 914–919, 2013.
9. Fuxing Cheng, Xin Zhang, Ben He, Tiejian Luo, and Wenjie Wang. A survey of learning to rank for real-time twitter search. In *Proceedings of Joint Int. Conf. ICPCA/SWS'12*, pages 150–164, 2013.
10. Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora Varvarigou. Content vs. context for sentiment analysis: A comparative analysis over microblogs. In *Proceedings of Int. Conf. HT'12*, pages 187–196, 2012.
11. Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
12. Abdelhamid Chellal, Lamjed Ben Jabeur, Laure Soulier, Bilel Moulahi, Thomas Palmer, Mohand Boughanem, Karen Pinel-Sauvagnat, Lynda Tamine, and Gilles Hubert. IRIT at trec microblog 2015. In *34th Text REtrieval Conference (TREC 2015)*, 2015.