Aggregative information retrieval on social media stream: Tweet summarization

Abdelhamid CHELLAL *, Bernard Dousset *
abdelhamid.chellal@irit.fr, bernard.dousset@irit.fr

(*) IRIT, 118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9 France.

Mots clefs :
Synthèse de tweets ; Optimisation, Signes sociaux ; filtrage par apprentissage.
Keywords:
Tweet summarization; Optimization; social signal; learning to filter.
Palabras clave :
Resumen de tweets; Optimización; Señal social; Aprender a filtrar

Abstract

This paper addresses the challenge of tweet stream filtering and summarization, which is an important task for keeping users up to date on topics they care about without overwhelming them with irrelevant and redundant posts. To cut down the noise and shield users from unwanted posts, tweet stream is filtered and a concise summary containing relevant and non-redundant posts is generated. Rather than rely on traditional threshold filter based only on tweet content, we exploit social signals as well as query dependent features to train a binary classifier in attempts to filter out irrelevant tweets with respect to the topic of interest. The core intuition is that the use of machine learning algorithm allows to overcome the issue of threshold setting and to examine how effective is the use of social signals in tweet filtering. Unlike existing approaches that generate a summary by selecting iteratively top weighted tweets, we formulate the summary generation as an optimization problem to select a subset of tweets that maximizes the global summary relevance and fulfills constraints related to non-redundancy, coverage, temporal diversity and summary length. Our experiments were conducted on TREC RTF 2015 and TREC RTS 2016 datasets. The former was used to train the classifier, while the latter was used to evaluate the proposed approach. The obtained results have shown the effectiveness of our approach.

1 Introduction

The freshness and diversity of information published in social media such as Twitter have raised them as an important source of real-time information to up-to-date about an ongoing event. When an important event occurs, users are increasingly relying on Twitter as source of real-time information in order to make up for what they would have missed regarding the event of interest. This is because, Twitter provides, in many cases, the latest news before traditional media, especially for unscheduled events. Hu et al. [17] have shown that quite a few big news stories have been broken on Twitter earlier than in more traditional news media. An example of this is the news about Osama bin Laden's death. Several sources on Twitter leaked the information before the President of the United States announced that bin Laden had been killed [17]. However, due to the high volume of daily produced posts, monitoring all published tweets to extract key information that describes the development of a given event over time turns out to be time-consuming with a risk of overloading users with irrelevant and redundant posts. Automatically producing tweets summaries containing key information (tweets) about an event or an entity is one possible solution to cope with these issues. The purpose of tweet summarization system is helping users to efficiently acquire social media information and follow the development of an event of interest. To achieve such a goal, a tweet summarization system has to monitor the live stream of tweets and generate summary that captures what has occurred up until now.

In this paper, we address the tweet summarization task in the social media stream to help users to follow the development of long-ongoing events. The goal is to produce a concise summary that captures key aspects of the underlying event and its development over the time. To be effective, such summaries are expected to fulfill some important properties such as relevancy, redundancy, coverage, diversity (in terms of time-period since important information may be spread out over the lifetime of the event) and the conciseness.

Optimizing all these criteria jointly is a challenging task especially for long-running events. In [15] the tweet summarization problem is proven to be NP-hard. This is because the inclusion of relevant tweets relies not only on properties of tweets themselves but also on the properties of every other tweet in the summary. Most of the approaches tackle tweet summarization by iteratively selecting the most relevant tweets and discarding those having their similarity with respect to the current summary above a certain threshold [3,5,7,15,17,20]. Such approaches ignore the mutual relation among tweets. In addition, these approaches do not consider the fact that important information may be spread out over the lifetime of the event of interest.

Additionally, to filter irrelevant tweets, the majority of existing approaches [1,6,11,18] exploit query dependent features that correspond to statistics of query terms such as term frequency, and term distribution in the stream. The decision to select or to discard an incoming tweet depends on whether its relevance scores fall above a predefined threshold. In these approaches, the threshold setting has a substantial impact on the quality of the tweet filtering [1,6]. Also, while considerable work has been done in leveraging social features as additional relevance factor in ad-hoc tweets retrieval, there is still a lack of studies that analyze how effective is the use of social signals in real-time tweet filtering. A key limitation of many social features used in the literature is that they are not suitable for real-time filtering because they either require several API calls for crawling the required information (e.g. the popularity in the social network) or are not yet available (e.g. the number of time a tweet has been retweeted).

To tackle these issues, we propose a novel approach that follows a different paradigm with the goal of increasing the coverage of different subtopics and time windows of a long ongoing event by considering the mutual relation between tweets. We tackle the aforementioned issues as follows:

1. To overcome the issue related to threshold setting in tweet filtering, we introduce a learn to filter approach based on machine learning to build a binary classifier that produces tweet filtering predictions. To study the impact of social signals in tweet stream filtering, We proposed and evaluated a set of social and other non-content features suitable for real-time tweet filtering. We distinguish two classes of features. The first one consists of tweet specific features that include particular characteristics of tweets, such as the presence of URLs and whether it is a reply to another tweet or a retweet. The second class consists of user account features, which refer to the activity and the influence of the author of the post on the social network. To fit real-time filtering scenario, our method leverages only the available and accessible features in the tweet without retrieving any further information from Twitter's servers. We argue that considering social features enhances the effectiveness of the relevance filter.
2. To optimize all the aforementioned criteria, we formulate the summary generation as an optimization problem modeled using Integer Linear Programming (ILP)[14]. An ILP problem is a constrained optimization problem, where both the cost function and constraints are linear in a set of integer variables. The summary generation is considered as an optimization problem that consists of selecting a subset of tweets that maximizes the global summary relevance and fulfills constraints related to non-redundancy, coverage, temporal diversity and summary length. To achieve this, two incremental clusters of posts are determined, namely topical cluster, and temporal cluster. The former is based on tweet content while the latter is based on publication times. A set of tweets are selected for inclusion in the summary such as that they can cover as many important subtopics and time windows as possible, while within the specified length limit. This is realized by using the ILP-based framework with objective function set to maximize the overall relevance and constraints ensure that at most one post per cluster from the two categories of clusters (topical and temporal) is selected.

2 Related work

2.1 Real-time tweet filtering

In the majority of existing approaches, the decision of selecting or ignoring an incoming tweet is based on its relevance and redundancy scores. The relevance score is evaluated using terms frequency, query term occurrence [6] in a tweet, stream statistics [1,11] and the redundancy is estimated using word overlap or cosine similarity [19]. The TREC MB RTF-2015 official results reveal that the PKUICSTRunA2 [11] and UWaterlooATDK [6] runs are the two best performing approaches among 37 runs from 14 groups [12]. In the former, the relevance score of tweets is evaluated by using the normalized KL-divergence distance, and the decision to select a tweet is based on a predefined threshold set manually. The ranked list of top-10 selected tweets of the previous day is manually scanned from top to bottom, and the relevance score of the first irrelevant tweet is chosen as a threshold in the next day for the related topic.
The analysis of these approaches reveals the following drawbacks: (i) The majority of state-of-the-art approaches consider only query dependent features to compute tweet relevance, which include features corresponding to particular statistics of query terms such as term frequency, and term distribution in the stream. Although it is recognized that social signals features are important for relevance, it is unclear how effective is the use of social signals in tweet filtering. (ii) The effectiveness of prospective notification system mainly depends on the setting of an appropriate threshold. Indeed, the relevance threshold value controls the number of pushed tweets. The use of a tight threshold value may lead to missing important updates. Conversely, a broad threshold value yields to overwhelm the user with irrelevant notifications. A considerable improvement can be achieved through better threshold setting regardless of the relevance scoring function.

2.2 Microblog summarization

Tweets summarization can be considered as an instance of the more general problem of multi-documents automatic summarization which can be categorized into two classes namely, extractive summarization and abstractive summarization. The former consists of selecting of the most meaningful sentences from documents being summarized exactly as they appear in the original documents whereas the latter may generate sentences that do not appear in the original documents. As in traditional document summarization, extractive approaches are predominant on tweet summarization. This is due to the difficulty of abstractive summarization which usually requires advanced language generation and compression techniques. We focus in this paper on extractive approaches since our approaches fall within this category.

In extractive summarization, two categories of approaches were proposed to measure the relevance of tweets namely graph-based and feature-based. In graph-based approaches, a tweet stream is modeled as a graph where a vertex denotes a tweet and an edge represents the similarity between tweets. Feature-based approaches are mostly based on text statistical such as term frequency [10] TF-IDF [8], HybridTF-IDF [5], Temporal TF-IDF [18] and language model [11]. To rank a set of candidates tweet, some work suggest combining text statistical features and social features of users such as the number of followers as well as tweet features such as the number of retweets [9]. These approaches rely on tweet stream statistics. Alternative features based on the query terms occurrences in the text of tweet [6] and in the external URL webpage [22] have shown to be effective. The approach proposed in [3] is one of the first real-time summarization approaches for scheduled events. It is based on term frequency in order to measure the salience of tweets, and the Kullback-Leibler divergence in order to reduce redundancy. Sharifi et al [5] introduced a HybridTF-IDF approach, where the *TF* component is calculated over the overall set of tweets (considered as one document). Top-weighted tweets are iteratively extracted with the exclusion of those having cosine similarity above a predefined threshold with tweets belonging to the current summary. Shou et al. [15] proposed (Sumblr), a continuous tweet summarization approach that provides two types of summaries; online, and historical. In Sumblr tweets are clustered. Those with the highest score in each cluster are selected for inclusion in the summary. However, In this approach authors presume the availability of a topic-related tweet stream.

3 Tweet summarization

Our goal is to periodically generate the summary that can best convey the main ideas of the user information need within length limit and a minimum of redundancy. To achieve this purpose, the proposed approach includes two main components: (i) **An online component** that crawls, filters and clusters tweets in real time after prepossessing step, and (ii) **An off-line component** that generates a summary periodically after the end of a predefined time window for instance one day. The online component consists of three main steps as listed below:

- **Pre-possessing and quality filtering:** The pre-processing step consists of stop-words removal, stemming and tokenization. After that, it filters out trash tweets and those that do not have enough overlap with the query (at least 2 query terms).
- **Relevance estimation and filtering:** In this step, first a relevance score of the incoming tweet with respect to the query is evaluated. This score is computed using the model (WSEBM) proposed in [2]. Then potential irrelevant tweets are discarded. To filter tweets in this stage, we rely on the binary classification model build on Random Forest algorithm that leverage social feature as well as query dependent features as described in the following section.
- **Incremental tweet clustering:** The purpose of this step is to identify the different subtopics (aspects) of an event and to gather tweets in different time windows over the lifetime of the given event. Tweets are clustered in real-time while they arrive.

The off-line component consists of the summary generation process that selects a subset of tweets from a set of candidate tweets that pass the filtering step. The goal is to select tweets that fulfill requirements related to the non-redundancy, the topical coverage, temporal diversity, and the summary length. To achieve this goal, we propose to use an Integer Linear Programming (ILP) model. This step is executed periodically within a predefined time period.

3.1 Real-time tweets filtering

Instead of relying on threshold-based relevance filter to discard irrelevant tweets, we propose a learn to filter approach based on machine learning to build a binary classifier that predicts the relevance of an incoming tweet with respect to the topic of interest. We consider the tweet filtering task as a binary classification problem in which each tweet is classified as either relevant or irrelevant. One of the most important tasks of a machine learning algorithm is the selection of features. A key limitation of many social features used in the literature is that they are not suitable for real-time filtering because they either require several API calls for crawling the required information (e.g. The popularity in the social network) or are not yet available (e.g. The number of time a tweet has been retweeted).
To overcome this issue, we proposed and evaluated a set of social and other non-content features suitable for real-time tweet filtering. We distinguish three classes of features. The first category of features includes query dependent features that measure the relevance of the incoming tweet with respect to the query. The second one consists of tweet specific features that include particular characteristics of tweets, such as the presence of URLs and whether it is a reply to another tweet or a retweet. The third class consists of user account features, which refer to the activity and the influence of the author of the post on the social network. To fit real-time filtering scenario, our method leverages only the available and accessible features in the tweet without retrieving any further information from Twitter's servers. We argue that considering social features enhances the effectiveness of the relevance filter. The proposed features are described in Table 1.

3.2 Incremental tweet clustering

The summary should cover all the aspects users are interested in. For example, a summary of a natural disaster should include aspects of what happened, when/where it happened, damages, rescue efforts, etc., and these aspects are provided by different tweets. We assume that an effective summary should also contain information nugget from different time window in order to give an overview of the development of the event. Hence, we propose to consider both dimensions, topical similarity and temporal distance between tweets in order to enhance the coverage and diversity in the summary. Given a tweet stream, we automatically cluster tweets into two types of clusters namely topical and timeline clusters. In the former, tweets sharing similar terms are absorbed into the same cluster and in the latter, tweets published in the same time window are gathered in the same timeline cluster.

Query dependent features	Tweet specific features	User account features
1 The number of words overlaps between the query's terms and hashtags in the tweet. 2 The cosine similarity between the query title and the tweet's text vectors using a word embedding model. 3 The relevance score of the incoming tweet with respect to the title. 4 The number of words that overlap between the text of the tweet and the query's terms	1 The ratio of the number of hashtags and the number of tokens in the tweet; 2 HasURL: Whether the tweet contains a URL; 3 HasEntity: This feature is a boolean property which indicates whether an entity (PERSON, ORGANIZATION, LOCATION) is mentioned in the tweet. 4 isReply: Whether a tweet is a reply to another tweet. 5 NbUser: The number of user-names mentioned in the tweet 6 TimeofPublication: At which hour during the day a tweet was published; 7 The length of the tweet: The number of tokens that a tweet text contains after removing stop words	1 Follower: Number of followers the account of the author of the given tweet currently has; 2 Friend: The number of users the account of the author of the given tweet is following; 3 Follower/day: Ratio of the number of user's followers and the age of the account (in days). 4 Friend/day: Ratio of user friends and the age of the account. 5 Fol/Fr: Ratio of the numbers of followers and friends (followees) of the user; 6 List/day: Ratio of the number of lists a user appears in and the age of the account. 7 (List + Fol/Fr)/day: combination of followers/friends ration and the number of lists the user appears in; 8 Tweet/day: Ratio of the number of tweets (including retweets) issued by the user and the age of the account.

Table 1 : Description of the proposed features used in the relevance classifier.

3.2.1 Subtopic clustering

The subtopic clustering is based on a pairwise similarity comparison between an incoming tweet and centroids of existing clusters. For an incoming tweet T, the key problem is to decide whether to absorb it into an existing cluster or to upgrade it as a new cluster. We first find the cluster whose centroid is the nearest to T. Tweet T is added to the closest cluster if its similarity score is greater than a predefined threshold (λ); otherwise, T is upgraded to a new cluster with T as the centroid. Each time an incoming tweet is added to an existing cluster its centroid is updated. We choose as new centroid the tweet that has the highest value of the sum of similarity scores with all other tweets in the cluster.

To overcome the issue of word mismatch when measuring the tweet-tweet similarity, we use word embedding model to estimate the similarity between tweet's terms. The similarity between two tweets T and T' is computed as follows:

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

Where $w2vsim(t_i,t_j)$ is the cosine similarity between vectors of terms and $t_i$and t_j which are generated by the word2vec model [20].

The use of word embedding allows considering different words with the same semantic meaning which can be valuable when calculating the similarity between two terms. Hence, tweets that contain different terms but sharing the same semantic context get high similarity score.

The use of maximum instead of average allows getting a similarity score equal to 1 if the term t_i occurs in both tweets. In the other case, where term t_i of tweet T does not occur in T', the maximum will return the similarity score of the most similar term in T' to t_i whereas the average may return a small score if terms that occur in T' are very different from t_i. This fact holds even if tweet T' contains term t_i. In the case that a tweet term is out of the vocabulary in the word embedding model, the similarity score is set to zero.

3.2.2 Timeline clustering

The aim of timeline clustering is to capture the development of the event over the time. We would like to avoid that the summary contains tweets published in the same time window. Indeed, we believe that all tweets that are published in the same time window are more likely to be related to each other. In timeline clustering,

tweets posted in the same time window are absorbed in the same timeline cluster. The decision to whether the incoming tweet is added to the current cluster is based on the delay (in seconds) between its timestamp and the timestamp of the first tweet used to create the actual cluster. If the delay is higher than a certain time window size, a new time cluster is created; otherwise, the incoming tweet is added to the current time cluster.

3.3 Summary generation

After filtering and clustering steps, the final step is the generation of the summary. We propose to formulate the tweet summarization as an Integer Linear Programming (ILP) problem in which both the objective function and constraints are linear in a set of integer variables. More specifically, we would like to select from *M* candidate tweets (those that pass the filter) *N* tweets that maximize the relevance score with respect to the query and fulfill a series of constraints related to redundancy, coverage, temporal diversity, and length limit. To find the optimal solution, we use the branch and bound algorithm[14].

Assume that there is a total of *M* candidate tweets that are clustered in *A* subtopic clusters (denoted C_j) among them there are *s* clusters that contain at least two tweets. In the same way, assume that there is a total of *W* timeline clusters (denoted TW_l) that contain at least two tweets. The tweet summarization problem can be formulated as the following ILP problem:

We include a binary variable X_i that is set to 1 if tweet T_i is added to the summary and 0 otherwise. The goal of the ILP is to set these indicators variables to maximize the payoff subject to the set of constraints that guarantee the validity of the solution. Notice here that the first constraint states that the indicator variables are binary.

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

3.3.1 Objective function

Top-ranked tweets are the most relevant tweets corresponding to the related aspects which we want to include in the final summary. Thus, the goal is to maximize the global relevance score of selected tweets that optimize the overall coverage, temporal diversity, and relevance of the final summary. The objective function is defined as follows:

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

Where $RSV(T_i, Q)$ is the relevance score of tweet T_i with respect to query Q that is computed according to the approach proposed in [3].

3.3.2 Constraints

1. **Coverage and redundancy constraints:** These constraints fulfill both redundancy and coverage requirements. In order to avoid redundancy, we just choose at most one tweet from each topical cluster. Indeed, the limitation of the number of tweets from each cluster guarantees that a maximum of sub-topics (aspects) will be presented in the summary such that the summary can cover most information of the whole tweet set. Assume that C_j is the j^{th} subtopic cluster and *s* is the number of subtopic clusters that contain at least two tweets. Then, these constraints are formulated as follows:

$$\forall\, C_j \in \{C_1, \dots, C_s\} \sum_{i;T_i \in C_j} X_i \leq 1$$

2. Temporal diversity constraints: To guarantee that the summary contains tweets from different time windows, we choose in maximum one tweet from each time window cluster. Assume that TW_l is the l[th] temporal cluster and *w* is the number of temporal clusters that contain at least two tweets. Then These constraints are formulated as follows:

$$\forall\, TW_l \in \{TW_1, \dots, TW_w\} \sum_{i;T_i \in C_j} X_i \leq 1$$

3. Length constraint: We add this constraint to ensure that the length of the final summary is limited to the minimum of either a predefined constant N (i.e. the maximum length) or (*M-1*) where *M* is the number of candidate tweets.

$$\sum_{i=1}^{M} X_i \leq \min(N, M-1)$$

4 Experimental Evaluation

We conducted several experiments to evaluate different aspects of our approach using large-scale real-world datasets. In the first experiments, we evaluate the effectiveness of leaning to filter approach and the impact of taking into account social in the task of real-time tweet filtering. In the second experiments, we compare our summary generation approach based on ILP against to the traditional method that consists of selecting iteratively the TOP-10 tweets. Third and last, we compare the performance of the outlined approach with those obtained in TREC RTS 2016 tasks [13].

4.1 Dataset

Experiments were conducted by using replay mechanisms of scenario A and B of TREC 2016 RTS [13] and TREC RTF 2015 [12] tasks to evaluate the performance of our real-time tweet filtering and summary generation methods respectively. In scenario A, a system is allowed to return a maximum of 10 tweets per day and per topic, while scenario B consists of identifying a batch of up to 100 ranked tweets per day per topic. For TREC RTF 2015, 51 topics were adopted for judging. The judgment pool contained 94068 tweets among which 8164 tweets were labeled as relevant. In TREC RTS 2016, 56 topics were selected among them 19 topics are new and unseen in TREC 2015. The judgment pool contained 67525 tweets, among which only 3339 tweets were considered as relevant. Topics provided in these tracks included a title and a complete description of the information need, indicating what is and is not relevant.

In our experiments, we used the TREC RTF 2015 dataset to tune the time window size and the similarity threshold that control the timeline and subtopic clustering receptively. The same dataset was used to train the binary classifier as follows: We extract for each topic tweets from the judgment pool of TREC 2015 MB RTF dataset. We obtain 94068 tweets among them 8164 tweets were labeled by assessors as relevant. We notice that the classes of these sets are unbalanced. There are many more instances in the irrelevant class than in relevant class in a training collection. In this case, a classifier tends to predict samples from the majority class which corresponds to irrelevant tweet in our training data. To get a balanced training dataset from classes' distribution point of view, we filter out all tweets that do not contain at least two query's words. Thus, we obtain a training dataset that contains 6663 tweets in which the distribution of relevant and irrelevant tweet is 50.18\% and 49.81% respectively. The binary classifier is built using a Random Forest algorithm [16]. Regarding word-embedding models, we used two separate models, the first for the dataset TREC 2015 and the second for the dataset TREC 2016. As training data, we used tweets crawled by Twitter stream API during 9 days before the official evaluation period from 11 to 19 July 2015 for TREC RTF 2015 and from 23 July to 01 August 2016 for TREC RTS 2016. We obtain a corpus of 264173 words and 8085225 tweets and a corpus of 348690 words and 11953129 tweets to train the model used for TREC 2015 and TREC RTS 2016 respectively. These models were generated using the skip-gram learning schema of word2vec model, which produces better word vector for infrequent words than Continuous Bag-of-Words

(CBOW) learning schema [20]. The dimension of the word vector was set to 300 and the context window (the maximum distance between two words) was set to 5 since the average length of the tweet is 11 words.

4.2 Evaluation metrics

The quality of tweet summary is evaluated using the Normalized Discounted Cumulative Gain (nDCG) which gives higher value to the well ranked list. nDCG@10 was defined as the official metric for TREC 2015 and 2016 track [12]. For each topic, the list of tweets returned per day is treated as a ranked list and from this nDCG@10 is computed. The score of a topic is the average of the nDCG@10 scores across all days in the evaluation period. The score of the system is the average over all topics. Notice that tweets in judgment pool were clustered, and only the first tweet from each cluster receives any gain. The performance of real-time tweet filtering approaches (scenario A) are evaluated using the Expected Gain (EG) and the normalized Cumulative Gain (nCG) metrics which are defined as follows [13]:

$$EG(S) = \frac{1}{N}\sum_{T \in S} G(T), \qquad nCG(S) = \frac{1}{Z}\sum_{T \in S} G(T)$$

Where *S* is the generated summary, *N* is the number of returned tweets and *Z* is the maximum possible gain (given the 10 tweet per day limit). G(T) is the gain of each tweet, set as follows: irrelevant tweets receive a gain of 0, relevant tweets receive a gain of 0.5 and highly relevant tweets receive a gain of 1.0.
In order to better evaluate the ability of systems to identify the case where no relevant tweet appears for some days and for some topics (silent days), two variants of the aforementioned metrics were considered, namely (nCG-1, nCG-0), (EG-1, EG-0) and (nDCG-1, nDCG-0). In nCG-1, EG-1 and nDCG-1, systems that do not push any tweets for a silent day are rewarded by receiving a perfect score (one) and systems that push tweets for a silent day are penalized by receiving a score of zero for that day. In nCG-0, EG-0 and nDCG-0, all systems receive a gain of zero no matter what they do for the silent day.

4.3 Results and Discussion

4.3.1 Evaluation of the learning to filter approach

In this section, we examine the performance of the learning to filter approach as well as the impact of considering social signals features in real-time tweet stream filtering. First, we compare results obtained when all features are combined (BC(QF+TF+UF)) against the following degenerate versions of the classification model: (i) BC(QDF) classification model based solely on query dependent features,(ii) BC(QDF-UF) takes into account the query-dependent features (QDF) as well as user account features, (iii) BC(QDF-TF) combines query-dependent features with tweet specific features. Second, we compare our results with the high-performing official results from the TREC RTS 2016 track. In this comparison, we present the obtained results when all topics are considered as well as when only 19 new topics of TREC 2016 are taken into account. Notice here that these topics were unseen in TREC 2015 dataset which was used to train the binary classifier.

From Table 2, we can see that the use of social signals features improves the quality of the classifier over EG-1 and nCG-1 metrics. We found performance improvements up to EG-1 and nCG-1 measures of about 13.45% and 18.25% respectively for the filter that does not take into account social signals BC(QDF). It is interesting to observe that the use of social signal improves EG-1 and nCG-1 since these metrics are quite similar to precision and recall and in general systems try to make a trade-off along them. Comparing the two classes of social signals, we observe that the tweet features contribute to enhancing performance in terms of EG-1 and nCG-1 whereas the consideration of the user account features yield to improve performance in terms of EG-0 and nCG-0 metrics in which systems are not penalized for pushing tweets for a silent day. These results reveal that the user account features allows identifying more relevant information whereas the use of tweet specific features yields to remain silent on days when there are no relevant posts.

Results shown in Table 2 reveal that our method outperforms the best automatic TREC run [19] in all metrics. We notice that the performance improvements of our method are up EG-1, EG-0, nCG-1, and nCG0 of about 5.29%, 55.14%, 17.87%, and 305.09% respectively. The positive improvements obtained by our approach were found to be statistically significant in terms of EG-0, nCG-1, and nCG-0. This result reveals that our approach achieves a good balance between pushing too many tweets and pushing too few tweets. To get a more detailed understanding of the effectiveness of machine learning-based approach to filter tweet stream with respect to unseen topics in training dataset used to build the binary classifier, we present in the last block of Table 2 results within 19 new topics introduced in TREC RTS 2016. We observe that our approach significantly outperforms the best automatic TREC run overall metrics. We also notice that our approach enhances the EG-1 and nCG-1 obtained by the high-performing TREC 2016 automatic run with an improvement of 20.96% and 54.79% for EG-1 and nCG-1 respectively. We also remark that the performances' improvements in terms of cumulative gain measures (nCG-1 and nCG-0) are more significant than the improvement in terms of expected gain measures. This result can be explained by the fact that our approach pushed more relevant tweets than the two others runs. With this experiment, we show clearly that the proposed approach is topic independent.

Method	**EG-1**	**EG-0**	**nCG-1**	**nCG-0**
BC(QDF+TF+UF)	**0.2793**	**0.0498**	**0.2828**	0.0636
BC(QDF+UF)	0.2705	0.0461	0.2750	**0.0643**
BC(QDF+TF)	0.2647‡	0.0396*	0.2701‡	0.0519
BC(QDF)	0.2613‡	0.0435	0.2641*	0.0453*
TREC RTS 2016 official Results				
TREC 1st automatic run	0.2643‡	0.0321*	0.2479†	0.0157†
TREC 2nd automatic run	0.2552*	0.0230†	0.2455†	0.0133†
Results within 19 new topics of TREC RTS 2016				
BC(QDF+TF+UF)	**0.2482**	**0.0733**	**0.2750**	**0.0999**
TREC 1st automatic run [19]	0.2052‡	0.0452‡	0.1776†	0.0176†
TREC 2nd automatic run	0.1878*	0.0278*	0.1741†	0.0141†

*Table 2: Comparison with TREC RTS 2016 scenario A results. Note. Symbols *,†,‡ denote the Student test significance:* $* : 0.01 \leq P - value < 0.05$, $\dagger : P - value < 0.1$, $\ddagger : 0.05 \leq P - value < 0.1$

4.3.2 Evaluation of tweet summarization

4.3.2.1 Impact of the use of ILP

In this section, we compare the impact of the use of ILP to generate the summary against the TOP-10 selection strategy within TREC RTF 2015. In [6] authors show that the treatment of silent days has a large impact on system scores in TREC MB RTF 2015. For this reason and to better perceive the impact of the use of the ILP, we present the obtained results over both all 51 topics and over only the 14 eventful days topics (for which there is no silent day). In this experiment, we gradually vary the similarity threshold from 0.5 to 0.95 at the step of 0.05. Recall that in TOP-10 selection strategy, we discard tweets that have a similarity score regarding already selected tweets above the predefined threshold.

Figure 1 reports the results obtained overall judged topics (51) in terms of nDCG-1@10 by varying the similarity threshold used in subtopic clustering (λ) gradually. As shown in this Figure, the use of ILP yields better performances overall similarity threshold. The positive improvements are statistically significant with p values between 0.01 and 0.05 for the similarity threshold <=0.55 and between 0.05 and 0.1 for the similarity threshold >=0.6. We found performance improvements of about 3.48% for the similarity threshold equal to 0.5 and of about 6.20% for the similarity threshold equal to 0.6.

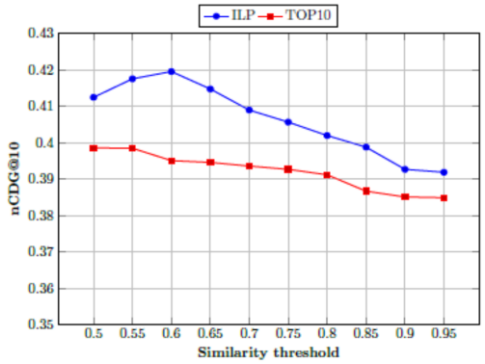

Figure 1: ILP vs TOP10 over all topics.

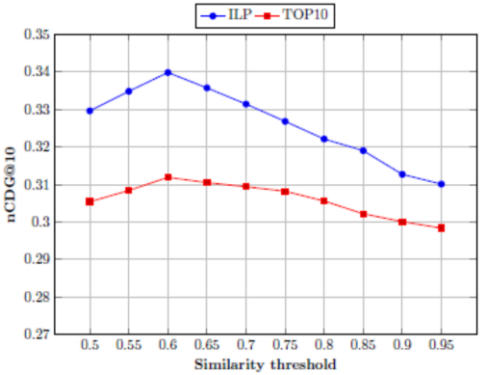

Figure 2: ILP vs TOP10 over eventful top

Method	nDCG-1@10	nDCG-0@10
BC(QDF+TF+UF)+ ILP	**0.2950**	**0.1201**
TREC 1st automatic run (nudt sna)	0.2708‡	0.0529†
TREC 2nd automatic run (QUJM16)	0.2621*	0.0301†

*Table 3: Comparison with TREC RTS 2016 results. Note. Symbols *,†,‡ denote the Student test significance: $* : 0.01 \leq P - value < 0.05$, $\dagger : P - value < 0.1$, $\ddagger : 0.05 \leq P - value < 0.1$*

From Figure 2, we can see that the performance improvements of ILP compared to the TOP-10 approach in terms of nDCG-1@10 are better over eventful days topics than overall 51 topics and overall the similarity threshold values. When only the eventful topics are considered, the obtained performance improvements of ILP vary between 7.92% and 8.94% for the similarity threshold equal to 0.5 and 0.6 respectively. Whereas when considering all topics, the use of ILP improves the performance with about 3.48% and 6.20% for the same similarity thresholds. These results reveal that the proposed method is more effective for events that raise many reactions in social media. In fact, the impact of tweets clustering and the use of ILP to generate a summary is more significant when the number of candidate tweets M is greater than the desired length limit of the summary N (set to 10 in our experiments). In the case of $M \leq N$, the ILP component acts almost like top-K ranking methods since it selects all candidate tweets with discarding the redundant tweets.

4.3.2.2 Comparative evaluation with the official TREC MB-RTS 2016 results

In Table 3, we compare our results with the high-performing results from the TREC RTS 2016 track [13]. We present results that were obtained by using ILP after filtering tweet stream with the filter based on word overlap and binary classifier that combine query depends and social features. In this experiment, the similarity threshold and time window were set to 0.6 and 600s receptively based on tuning experiments conducted on TREC 2015 dataset. Table 3 shows results in terms of nDCG-1@10 and nDCG-0@10 obtained by our approach and the two best automatic runs in TREC task. Results shown in Table 3 are promising because we outperform the best automatic TREC run overall metrics. We found performance improvements up to nDCG-1@10 values of about 6,82% and 9.60% for the first and second automatic TREC run respectively. The improvement of performance in terms of nDCG-1 can be explained by the fact that the filter used by our system is able to identify silent days. These results reveal that our approach achieves a good balance between pushing too many tweets and pushing too little tweets.

5 Conclusions

To tackle the difficult task of tweet stream filtering and summarization for a long on-going event, we introduced a new approach that combines filtering and optimization frameworks to generate a periodic summary of tweet streams. The main contribution of the proposed method is a filtering stage based on machine learning approach that takes into account social signals. We make use of word embedding, which counters the shortness of tweets and the word mismatch issues. To avoid redundancy and to enhance summary coverage, tweets are clustered by similarity, and to provide temporal diversity, a timeline clustering is used. The tweet selection problem is formulated as an ILP that maximizes the objective function based on the tweet's relevance score subject to a series of constraints related to redundancy, coverage, temporal diversity, and length limit. Experimental results based on a real word dataset revealed that the proposed approach outperforms the best automatic TREC RTS 2016 systems. We highlight the importance of social signals in tweet stream filtering tasks. The learning based filter achieves a good balance between pushing too many or too few tweets at the cost of low latency. The results also showed that more improvements are achieved on the queries with eventful days in a tweet stream.

Bibliographie

[1] ABDELHAMID CHELLAL, MOHAND BOUGHANEM, AND BERNARD DOUSSET. *Multi-criterion real time tweet summarization based upon adaptive threshold. In 2016 IEEE/WIC/ACM , WI 2016, Omaha, NE, USA, October 13-16, 2016, pages 264–271, 2016.*

[2] ABDELHAMID CHELLAL, MOHAND BOUGHANEM, AND BERNARD DOUSSET. *Word similarity based model for tweet stream prospective notification. In Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings, pages 655–661,* 2017.

[3] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. *Towards real-time summarization of scheduled events from twitter streams. In Proceedings of the 23rd ACM Conference on Hypertext and Social Media, HT '12, pages 319–320,* 20120.

[4] A. NENKOVA AND L. VANDERWENDE. *The impact of frequency on summarization. Technical Report MSR-TR-2005-101, MSR-TR-2005-101, January* . 2005.

[5] BEAUX SHARIFI, MARK-ANTHONY HUTTON, AND JUGAL K. KALITA. *Experiments in microblog summarization. In Proceedings of the 2010 IEEE Second International Conference on Social Computing, SOCIALCOM '10, pages 49–56,* 2010.

[6] 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.

[7] DAVID INOUYE AND JUGAL K. KALITA. *Comparing twitter summarization algorithms for multiple post summaries. In PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011, pages 298–306,* 2011.

[8] DEEPAYAN CHAKRABARTI AND KUNAL PUNERA. *Event summarization using tweets. In Proceedings of the Fifth International Conference onWeblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21,* 2011 .

[9] DOUGLAS W. OARD MOSSAAB BAGDOURI. *Clip at trec 2016: Liveqa and rts. In Proceedingsof The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland,USA, November 15-18, 2016,* 2016.

[10] FEI LIU, YANG LIU, AND FULIANGWENG. *Why is "sxsw" trending?: Exploring multiple text sources for twitter topic summarization. In Proceedings of the Workshop on Languages in Social Media, LSM '11, pages 66–75*, 2011.

[11] FEIFAN FAN, YUE FEI, CHAO LV, LILI YAO, JIANWU YANG, AND DONGYAN ZHAO. *Pkuicst at trec 2015 microblog track: Query-biased adaptive filtering in real-time microblog stream. In Text REtrieval Conference, TREC, Gaithersburg, USA, November 17-20,* 2015.

[12] 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.

[13] JIMMY LIN, ADAM ROEGIEST, LUCHEN TAN, RICHARD MCCREADIE, ELLEN VOORHEES, AND FERNANDO DIAZ. *Overview of the trec 2016 realtime summarization. In Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18,* 2016.

[14] JURAJ HROMKOVIC AND WALDYR M. OLIVA. *Algorithmics for Hard Problems. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition*, 2002. (Hromkovic:2002)

[15] LIDAN SHOU, ZHENHUA WANG, KE CHEN, AND GANG CHEN. *Sumblr: Continuous summarization of evolving tweet streams. In the 36th International ACM SIGIR Conference, SIGIR '13, pages 533–542,* 2013.

[16] L. BREIMAN, "*Random forests," Machine Learning, vol. 45, no. 1, 2001, pp. 5–32.*2001

[17] MENGDIE HU, SHIXIA LIU, FURU WEI, YINGCAI WU, JOHN STASKO, and KWAN-LIU MA. *Breaking news on twitter. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, pages 2751–2754* 2012

[18] NASSER ALSAEDI, PETE BURNAP, AND OMER F. RANA. *Automatic summarization of real world events using twitter. In Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016, pages 511–514*, 2016 2016.

[19] TAMER ELSAYED REEM SUWAILEH, MARAM HASANAIN. *Light-weight, conservative, yet effective: Scalable real-time tweet summarization. In Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016,* 2016.

[20] TOMAS MIKOLOV, KAI CHEN, GREG CORRADO, AND JEFFREY DEAN. *Efficient estimation of word representations in vector space. CoRR, p1301.3781,* 2013.

[21] WANG ZHENHUA, SHOU LIDAN, CHEN KE, CHEN GANG, AND MEHROTRA SHARAD. *On Summarization and Timeline Generation for Evolutionary Tweet Streams IEEE Trans. Knowl. Data Eng., (5):p 1301–1315,*2015

[22] Zhongyuan Han, Song Li, Leilei Kong, Liuyang Tian, and Haoliang Qi. *Hljit at trec 2017 real-time summarization. In Proceedings of The Twenty-Six Text Retrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017, 2017.*