Optimization Framework Model For Retrospective Tweet Summarization

Abdelhamid Chellal, Mohand Boughanem

Optimization Framework Model For Retrospective Tweet Summarization

Abdelhamid Chellal
RIT UMR 5505 CNRS, University of Toulouse III
118 route de Narbonne
Toulouse, France 31062
Abdelhamid.chellal@irit.fr

Mohand Boughanem
RIT UMR 5505 CNRS, University of Toulouse III
118 route de Narbonne
Toulouse, France 31062
mohand.boughanem@irit.fr

ABSTRACT

Twitter is a valuable source of information to keep users up to date on topics they care about. However, timely following the development of long-running events is too difficult due to the velocity and the volume of the published information. Automatically generating a concise summary containing relevant and non-redundant posts that capture key aspects of information need, is one solution to keep users up to date. In this paper, we propose a novel approach that formulates the summary generation as an optimization problem modeled using Integer Linear Programming whereas the majority of traditional methods generate the summary by selecting iteratively top weighted tweets and ignores the mutual relation among messages. To overcome this issue, the generation of the summary is formulated 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 on TREC RTF 2015 and TREC RTS 2016 datasets have shown the effectiveness of our approach.

CCS CONCEPTS

•**Information systems** →**Information retrieval diversity;** *Combination, fusion and federated search;*

KEYWORDS

Tweet summarization, optimization, temporal diversity.

Abdelhamid Chellal and Mohand Boughanem. 2018. Optimization Framework Model For Retrospective Tweet Summarization. In *Proceedings of SAC 2018: Symposium on Applied Computing , Pau, France, April 9–13, 2018 (SAC 2018),* 8 pages.
DOI: 10.1145/3167132.3167210

1 INTRODUCTION

Twitter has shown to be a useful source of real-time information about what is happening or what is being said about an entity. Indeed, in many cases, the most current news is provided by Twitter before traditional media especially news about unscheduled events such as natural disaster. However, due to the high volume of daily produced posts, monitoring and following all published information describing the development of a given event over time or referring to an entity turns out to be time-consuming with a risk of overloading users with irrelevant and redundant posts.

Automatically producing summaries containing key information (tweets) about an event or an entity is one possible solution to cope with this issue. However, to be effective such summaries are expected to fulfill some important properties such as relevance, low redundancy, topical and temporal coverage and length of the summary. Optimizing all these criteria jointly is a challenging task especially for long-running events.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.

Several approaches have been proposed to tackle this issue [4, 12, 18, 20–22]. Most of these approaches generate summaries by iteratively selecting the most relevant tweets and discarding those having their similarity with respect to the current summary above a certain threshold. Such approaches ignore the mutual relation among messages. Indeed, they mainly focus on ended events making them unsuitable to provide a summary of a long and ongoing event.

In this paper, we introduce a novel approach for retrospective tweet summarization in which incoming tweets are filtered and clustered continuously and the summary is generated periodical using an Integer Linear Programming (ILP) [3]. More precisely, the proposed method relies on a three-stage approach. First, tweets that do not have sufficient word overlap with the query are discarded. Second, two incremental clusters of posts are determined, namely topical cluster, and temporal cluster. The first one is based on tweet content and another is based on publication times. Third and last, a subset of posts is selected so as to maximize their overall relevance to the query subject to constraints related to, summary length, temporal diversity, coverage, and redundancy. In order to handle this selection, we formulate the tweet summary generation as integer linear problem in which unknowns variables are binaries and both the objective function (to be maximized) and constraints are linear in a set

of integer variables. Constraints ensure that at most one post per cluster from the two categories of clusters (topical and temporal) is selected with respect to the defined summary length limit.

To measure query-tweet and tweet-tweet similarity, we make use of word embedding, which counters the shortness of tweets as well as the term mismatch problem which is frequent in the context of tweets. To evaluate the relevance score of tweets with respect to the query, we use the adaptation of the Extended Boolean Model (EBM)[19] proposed by [1] in which the word embedding is used to estimate the weight of query terms.

The main contributions of the proposed approach are:

- We adopt Integer Linear Programming technique to periodically generate a summary in order to optimize all the aforementioned criteria. To reduce the computational complexity and handle the coverage issue, the tweet stream is filtered and clustered in real-time;
- We take into account the temporal diversity of tweets as one criterion that needs to be fulfilled in the summary generation process.

To evaluate the proposed approach, we carried out several experiments on TREC Microblog Real-Time Filtering 2015 (MB-RTF) dataset [7] and TREC Real-Time summarization (RTS) 2016 track [8].

2 RELATED WORK

2.1 Microblog summarization

Most of the proposed approaches attempt to generate summaries incrementally, by first selecting candidate relevant tweets, and then by discarding redundant one. The selection of a tweet in a summary is based on query-tweet and tweet-tweet matching. The approach proposed in [22] 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 Kullback-Leibler divergence [5] to reduce redundancy. Sharifi et al [20] 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 of the current summary. The Sumbasic approach [16], initially proposed for document summarization, was reported to be efficient as well for microblog summarization [12]. In this approach, the sentence that contains more frequent words has a higher probability of being selected for summaries than the one with words occurring less frequently. Shou et al. [21] proposed (Sumblr), a continuous tweet summarization approach that provides two types of summaries (online and historical). Tweets are clustered and those with the highest score in each cluster are selected for inclusion in the summary.

TREC 2015 Microblog Real-Time Filtering (MB-RTF) [7] and TREC Real-Time Summarization 2016 [1] are two evaluation campaigns. The TREC MB RTF-2015 official results for scenario B (identifying a batch of up to 100 ranked tweets per day and per topic) reveal that the best automatic run is CLIP-B-0.6 [15]. In this run, the relevance model is based on Okapi BM25 term weights and title expansion using word embedding. Tweets are clustered incrementally using the Jaccard similarity in which the incoming tweet is assigned to the cluster containing the most similar tweet if the similarity falls above a certain threshold. At the end of each day, the highest ranked tweet for each cluster is selected for the summary. In TREC RTS 2016, the best run PolyURunB3 [2] evaluates the relevance score by adding the number of occurrence of query terms in tweet text and in the external URL web-page text. In this run, tweets are filtered according to the relevance and the redundancy predefined thresholds. The similarity between two tweets is determined by occurrences of their common vocabulary. At the end of the day, the top-10 tweets are selected for the summary. The third best performing run QUJM16 [17] first retrieves tweets using a language model with Jelinek Mercer smoothing and then drops tweets that have a relevance score less than a predefined relevance threshold. For inclusion in the summary, the top-ranked tweets are selected iteratively but with discarding those having overlap with any tweet that was previously selected higher than the predefined threshold.

While our approach falls within this line of research, it differs from the previous ones by (i) it does not rely on stream statistics, which may change when new tweets arrive while the aforementioned methods are based on stream statistics to assess both the relevance of tweets and similarity score between two tweets. (ii) In the existing methods, top-K tweets are selected iteratively and the final score of tweets is evaluated by combining several criterion scores while in the proposed method, we formulate tweet summarization problem as ILP to select a subset of tweets that optimize all the criteria. The approach introduced in [22] is dedicated to scheduled events whereas the proposed approach is applicable to any kind of event.

2.2 ILP and Microblog summarization

Integer Linear Programming (ILP) techniques have been used in multi-document summarization [6, 13]. The selection of sentences is formulated as an optimization problem that is solved through a standard branch-and-bound algorithm to provide an exact solution. In [6], authors proposed an event-aspect model based on LDA for sentence clustering that uses ILP for sentence selection. The optimization problem is based on sentence ranking information that selects one sentence which receives the highest possible ranking score from each aspect cluster subject to two other constraints related to redundancy and summary length. For microblog summarization, a concept-based ILP formulation was proposed in [9]. This approach first extracts, for each topic,

[1] http://trecrts.github.io/TREC2016-RTS-guidelines.html

a set of important concepts which represent n-grams that appear frequently in tweets related to a topic but do not appear frequently in a corpus. The summary is constructed by selecting a set of tweets that can cover as many important concepts as possible with the objective function sets to maximize the sum of the weight of concepts and constraints related to the length (number of tweets and words) and the coverage (number of concepts). The optimization problem proposed in our work differs from those proposed in state of the art by:(i) It takes into account the temporal dimension which is not the case in the related works. (ii) The coverage and redundancy requirement are represented in the same constraint while in [9] a redundancy constraint is created for each pair of tweets which increases the computational complexity of the generated ILP.

3 TWEET SUMMARIZATION

For an ongoing event, our goal is to periodically generate the summary that can best convey the main ideas of the user interest within length limit and a minimum of redundancy. To achieve this purpose, the proposed approach includes two main components: (i) **tweet stream filtering and clustering component**, and (ii) **summary generation component**. The tweet stream filtering and clustering component consists of three main steps as listed below:

(1) *Tweet filtering:* This step discards potential irrelevant tweets which yields to reduce the number of candidates tweets and to decrease the computational complexity. By doing this, we make feasible the use of ILP.
(2) *Tweet relevance estimation:* In this step, a relevance score with respect to the query is evaluated.
(3) *Incremental tweet clustering:* The goal of this step is to identify the different subtopics (aspects) of an event.

The summary generation component 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 and temporal coverage, and the summary length. To achieve this goal, we propose to use an Integer Linear Programming (ILP) model which selects tweets that optimize a global objective function under certain constraints. This step is executed periodically within a predefined time window.

3.1 Tweet filtering

In our approach, we adopt TREC like query representation in which a query Q (user interest) consists of a title Q^t and a description Q^d of the information need. To discard potential irrelevant tweets, first, we apply a simple ad-hoc filter that excludes any tweet that does not match at least one term in the title of the query and with text length shorter than five terms, or contain more than one URL or three hashtags. At this stage, the incoming tweet T is considered as a candidate tweet only if it passes the relevance filter introduced in [10] which is based on the number of occurrence of query terms in the incoming tweet as follows:

$$(3 \times |T \cap Q^t| + |T \cap Q^d|) \times \frac{|T \cap Q^t|}{|Q^t|} >= GT \qquad (1)$$

Where $|T \cap Q^t|$ and $|T \cap Q^d|$ are the number of title terms and descriptions terms that occur in the tweet respectively. The main advantage of this relevance filter is that a single static global threshold (GT) can be used across all queries.

3.2 Relevance estimation

We argue that statistics based approaches such as Okapi BM25, vector space model or language model are not suitable to evaluate the relevance of tweets with respect to an ongoing event for the following reasons: first tweets are short and commonly expressed in an informal way which leads to the issue of word mismatch. Second, the availability of statistics and their update. In the starter, statistics require to be estimated (with a previous stream from which the statistics were computed) and an update strategy needs to be set up (update statistics with every tweet, periodically or in batch). To overcome these issues, we adopt an approach proposed by [1] in which the relevance score of an incoming tweet is evaluated at the time the new tweet arrives, independently of the previously seen tweets in the stream and without the need to maintain any tweet stream statistics. In this approach, denoted by Word Similarity Extended Boolean Model (WSEBM), an adaptation of Extended Boolean Model is proposed and authors take advantage of word embedding to overcome the word mismatch and the shortness issues. The query title Q^t is considered as "**AND**ed terms", whereas the description of the information need Q^d is considered as"**OR**ed terms". The relevance scores of the tweet $T = \{t_1, ..., t_n\}$ to "AND query" Q^t and "OR query" Q^d are estimated respectively as follows:

$$RSV(T, Q^t_{and}) = 1 - \sqrt{\frac{\sum_{q_i^t \in Q^t}(1 - W_T(q_i^t))^2}{|Q^t|}} \qquad (2)$$

$$RSV(T, Q^d_{or}) = \sqrt{\frac{\sum_{q_i^d \in Q^d}(W_T(q_i^d))^2}{|Q^d|}} \qquad (3)$$

Where $W_T(q)$ is the weight of the query term q in the tweet T. q stands for the term q_i^t in the title Q^t or the term q_i^d in the description Q^d of the query. This weight of a query word q is determined as the similarity score between q and all tweets' terms as follows:

$$W_T(q) = \max_{t_i \in T}[w2vsim(t_i, q)] \qquad (4)$$

Where $w2vsim(t_i, q)$ is the cosine similarity between word2vec vectors [14] of the tweet word t_i and the query word q.

The final relevance score of the tweet with respect to the query is given by combining linearly the relevance score of tweet T regarding the query title ($RSV(T, Q^t_{and})$) and the query description ($RSV(T, Q^d_{or})$) as follows:

$$RSV(T, Q) = \lambda \times RSV(T, Q^t_{and}) + (1 - \lambda) \times RSV(T, Q^d_{or}) \quad (5)$$

Where $\lambda \in [0, 1]$ is an interpolation parameter that determines the trade-off between the title and the description of

the query. We choose to set λ to 0.5 which means that the relevance scores of the tweet with respect to the title and the description have the same importance.

3.3 Incremental tweet clustering

The summary should cover all aspects that users are interested in. For example, a summary of a natural disaster should include aspects about 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 temporality) in order to bring coverage and diversity in the summary. Given a tweet stream, our goal is to 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 tweet published in the same time window are gathered in the same timeline cluster.

3.3.1 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. The decision of whether T is added to the closest cluster is made if the similarity score is greater than a predefined threshold γ; 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 propose an adaptation of Jaccard similarity. We use word embedding model to estimate the similarity between tweet's terms instead of the intersection as follows:

$$Sim(T, T') = \frac{\sum_{t_i \in T} \max_{t_j \in T'} w2vsim(t_i, t_j)}{|T \cup T'|} \quad (6)$$

Where $w2vsim(t_i, t_j)$ is the cosine similarity between vectors of terms t_i and t_j which are generated by the word2vec model. The use of word embedding allows the exploitation of the semantic relationship between terms. Tweets that contain different terms but sharing the same semantic context get high similarity score which is not the case with word overlap and Jaccard coefficient. 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 the 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 the tweet T' contains one the term t_i. In the case of the word out of the vocabulary of in the word embedding model, the similarity score is set to zero.

3.3.2 Timeline clustering. We argue that all tweets that are published in the same time window are more likely related. Hence, these tweets 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, a new time cluster is created; otherwise, the incoming tweet is added to the current time cluster.

3.4 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 [3] .

Assume that there is a total of M candidate tweets that are clustered in A 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 an indicator variable X_i which is set to 1 when 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.

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

3.4.1 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 improve the overall coverage, temporal diversity and relevance of the final summary. The objective function is defined as follows:

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

3.4.2 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 sup-topics (aspects) will be presented in the summary such that the summary can cover most information of the whole tweet set. These constraints are formulated as follows:

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

3.4.3 Temporal coverage constraints. To guarantee the coverage of different time slot in the summary, we choose in maximum one tweet from each time window cluster. This constraint is formulated as follows:

$$\forall TW_l \in \{TW_1, ..., TW_w\} \sum_{i;T_i \in TW_l} X_i \leqslant 1$$

3.4.4 Length Constraints. 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 candidates tweets.

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

4 EXPERIMENTAL EVALUATION

To evaluate our approach, we use a large-scale real-world data-sets of tweets. We carried out twofold objectives experiments: First we conducted a series of experiments on TREC RTF 2015 dataset to tune parameters used in our approach. Second, we compare our approach with the state-of-the-art methods and with the three best performing run in TREC MB-RTS 2016 task. As baselines, we use the three approaches that were recommended by [12] to be considered as baselines since it turned out to be the best one among 11 different tweet summarization approaches. These approaches are TF-IDF, HybridTF-IDF[20] and sumbasic [16]. Indeed, To evaluate the impact of ILP, we consider as a baseline a variant of our approach in which we disable the ILP. In this baseline denoted by WSEBM-TOP10, we select iteratively the TOP-10 tweets but with discarding those having a similarity score above the predefined threshold (the same value of the one used for the topical clustering). We choose to select TOP-10 tweets because the evaluation metrics are computed on top-10 tweets.

4.1 Data set

Experiments were conducted by using replay mechanism of scenario B over tweets captured during the evaluation period of the TREC 2015 Microblog Real-Time Filtering (MB RTF) and the TREC 2016 Real-Time summarization (RTS) [8]. This task (scenario B) is more like a top-100 retrieval task based on a one-day. It consists of identifying a batch of up to 100 ranked tweets per day and per interest profile and then these tweets are delivered to the user daily at the end of the day. To tune our approach, we use TREC MB-RTF 2015 data-set. This dataset consists of 40,242,516 tweets and 225 predefined topics from which only 51 were assessed. The judgment pool contains 94,068 among them 8164 tweets were judged relevant. To evaluate the proposed approach, we use TREC RTS 2016 data-set. In TREC RTS 2016, the dataset size is 36,908,568 tweets and from 203 topics 56 were assessed. The judgment pool contains 67,525 tweets among them only 3339 tweets were considered relevant by assessors. Topics provided in these tracks included a title and a complete description of the information need that indicates what is and is not relevant. Notice here, that corpus used in our experiments were crawled using Twitters streaming API during the evaluation period of each TREC track (10 days: from 20 to 29 July 2015 and from 02 to 11 August 2016) which means that there is no lost data. If we had not crawled tweets during the TREC evaluation period, we would have got a corpus with missing data because not all tweets remain available.

4.2 Evaluation metrics

The Normalized Discounted Cumulative Gain (nDCG) was used to evaluate the performance of the system for periodic top-100 push task (scenario B). The nDCG gives higher value to the well ranked list. nDCG@10 was defined as the official metric for TREC 2015 MB-RTF task [7]. The gain of each tweet was set as follows: (i) Irrelevant tweets receive a gain of 0. (ii) Relevant tweets receive a gain of 0.5. (iii) Highly relevant tweets receive a gain of 1.0. Notice that tweets in judgment pool were clustered and only the first tweet from each cluster receives a gain. In this task, a special attention was paid to the case where no relevant tweet appears in the judgment pool for some days and topics. Days in which there are no relevant tweets for a particular topic are called "silent days", in contrast to "eventful days" (where there are relevant tweets) [11]. Systems that do not push any tweet for a silent day are rewarded by receiving a score of one (i.e., perfect score) and systems that push any tweet for a silent day are penalized by receiving a score of zero for that day. Hence, in TREC RTS 2016 two variants of the metric nDCG were considered namely nDCG-1 and nDCG-0. In nDCG-1, the system receives a score of one if it does not push any tweets for a silent day, or zero otherwise. In nDCG-0, all systems receive a gain of zero no matter what they do for the silent day.

4.3 Parameter Setting

4.3.1 Relevance Thresholding. To filter tweets, our approach makes a threshold-based decision based on the occurrence of query terms according to the Equation 1 in which only tweets with a score above a global threshold are considered as candidate tweet for the summary. To understand the impact of this threshold, Figure 1 shows the effectiveness of our approach on TREC 2015 dataset. The baseline for this plot is the performance of the best automatic run (CLIP-B-0.6)[15] in TREC RTF 2015 track. The best performance is obtained with the global threshold GT=4, which we retain for the remainder of this paper. We notice that our approach outperforms the best automatic run in TREC 2015 (CLIP-B-0.6) in terms of nDCG overall values of the global threshold. Clearly, we observe that our best configuration (GT=4) achieves substantial improvements compared to run (CLIP-B-0.6). This improvement in terms of nDCG@10 is statistically significant with p values < 0.05. We found performance improvement up to nDCG@10 of about 42.32%.

4.3.2 Word embedding model. The word embedding models used in our experiments were generated using the skip-gram schema of word2vec model, which produces better word

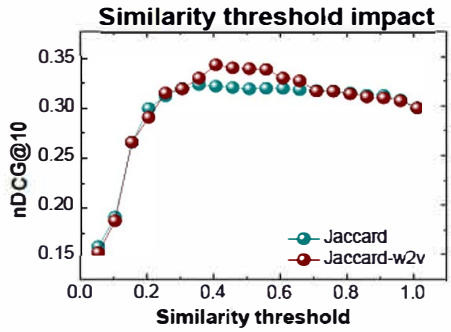

Figure 2: Effect of similarity threshold.

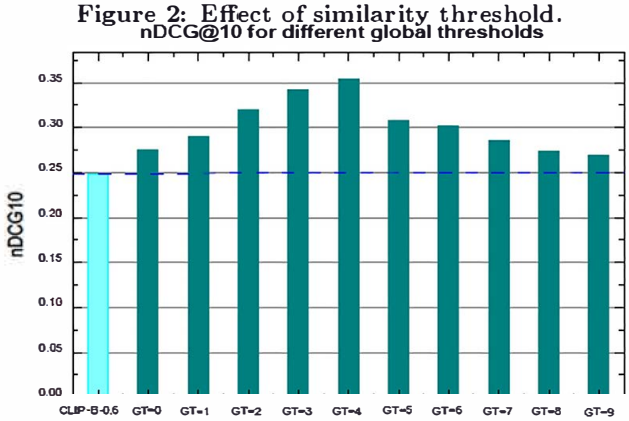

Figure 1: nDCG@10 for different global thresholds and the best TREC 2015 automatic run.

vector for infrequent words than Continuous Bag-of-Words (CBOW) schema [14]. As training data, we used tweets crawled by Twitter stream API from 11 to 19 July 2015 for TREC RTF 2015 and from 23 July to 01 August 2016 for TREC RTS 2016 which corresponds to 9 days before the official evaluation period. 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. 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.

4.3.3 Impact of topical clustering. The topical clustering is controlled by the tweet-tweet similarity measurement and the similarity threshold γ. Figure 2 shows the effect of the similarity threshold in terms of nDCG@10 obtained by the proposed similarity function denoted by (Jaccard-w2v) and the standard Jaccard similarity measurement. In this experiment, we gradually vary the similarity threshold γ from 0.05 to 1 at the step of 0.05 and we disable the temporal clustering by setting the time window size to zero($\tau = 0s$). From Figure 2, we can see that the performance improves when γ increases but it decreases when γ comes near to 1. We also notice that both functions (Jaccard-w2v, Jaccard) have the same performance when γ is small as well as when $\gamma > 0.6$. These results were expected for the following reasons: In the

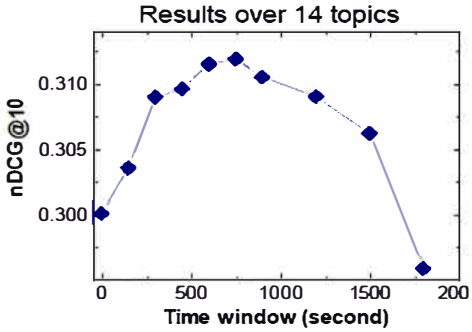

Figure 3: Effect of time window size.

first case (γ small) the number of clusters that contain at least two tweets decreases (all tweet may be gathered in the same cluster) and only one or few tweets (with the highest relevance score) are selected causing damage in terms of coverage. In the second case (γ near to one), there are no clusters with at least two tweets which means that there are no constraints related to topical coverage and regardless of the similarity measurement the ILP selects the top-k tweets without discarding the redundant ones. These results reveal that $\gamma = 0.4$ appears as a good choice as it gives a good balance between the number of clusters and the number of tweets in each cluster. We observe that the best performance is obtained by Jaccard-w2v. These results can be explained by the fact that when the word embedding is used, tweets that contain different terms that share the same context obtain a high similarity score whereas with the standard Jaccard measurement they obtain a low similarity score.

4.3.4 Impact of timeline clustering. The timeline clustering is based on the size of the time window. Hence to test the effect of the use of timeline clustering, we conducted experiments in which we vary τ from 0 to 1800 seconds and we keep others parameters fixed. The obtained results are shown in Figure 3. We notice that the performance decreases when τ, increases. In one hand, when τ is large, we obtain clusters that contain a lot of tweets causing to discard many tweets which damage the quality of the summary. In the other hand, when τ is very small no time cluster is created which means that we do not have any constraint related to temporal diversity. It seems that $\tau = 600s$ is a good value that leads to a good balance between the number of timeline clusters and the number of tweets in each cluster.

4.4 Results and Discussion

4.4.1 Impact of the use of ILP . We compare the impact of the use of ILP to generate the summary against the TOP-10 selection strategy within TREC RTF 2015. In [11] 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). Figure 4 reports the results obtained overall judged topics (51) in terms of nDCG@10 by varying the similarity threshold

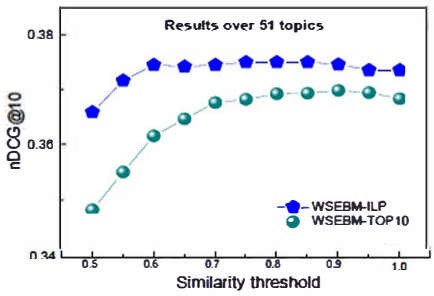

Figure 4: ILP vs TOP10 over 51 topics.

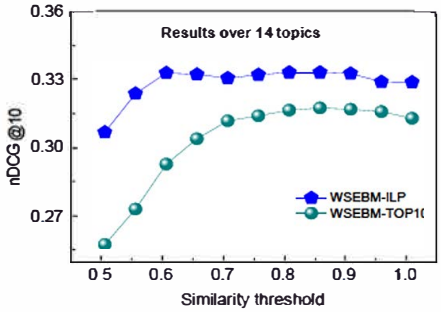

Figure 5: ILP vs TOP10 over 14 topics.

γ gradually. As shown in this Figure, the use of ILP yields better performances overall similarity threshold. The positive improvements is statistically significant with p values between 0.01 and 0.05 for the similarity threshold $\gamma <= 0.55$ and between 0.05 and 0.1 for the similarity threshold $\gamma >= 0.6$. We found performance improvements of about 6.78% for the similarity threshold $\gamma = 0.5$ and of about 3.11% for the similarity threshold $\gamma = 0.7$. From Figure 5, we see that the performance improvements of ILP compared to the TOP-10 approach in terms of nDCG@10 are better over eventful days topics than over 51 topics and overall the similarity threshold values. The performance improvements is varying between 21.12% and 6.32% for the similarity threshold $\gamma = 0.5$ and $\gamma = 0.7$ respectively. These reveal that the proposed method is more effective for events that raise a lot of 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 but with discarding the redundant tweets.

4.4.2 Comparative evaluation with state-of-the-art baselines and TREC MB-RTS 2016 results. In this section, we compare our best configuration with the three high-performing runs (PolyURunB3 [2], nudtsna, QUJM16 [17]) from the TREC RTS 2016 track [8] and against state-of-the-art baselines within TREC RTS 2016 dataset. To get a deeper understanding of the effectiveness of the proposed method, we show in Table 1 the obtained results in terms of nDCG-1@10 as well as in terms of nDCG0@10. We recall that in the latter metric, systems are not penalized for pushing tweets in a silent day. First, we notice that our approach outperforms the state-of-the-art methods overall metrics with an improvement up to 75.85%, for the HybridTF-IDF and up to 69.10% for TFIDF. We also notice that our best configuration slightly outperform the best performing run (PolyURunB3) in TREC 2016 in terms of nDCG1@-10 with a significant improvement of the performance in terms of nDCG0@10 in which systems are not penalized for pushing tweets for a silent day. The performance improvements are up to nDCG-1@10 and nDCG-0@10 values of about 1.79% and 75.58% respectively. The positive improvement in terms of nDCG0@10 is statistically significant with (p-value ¡0.01). Notice here that these performances are achieved despite the fact that our method is automatic, while in the best TREC runs (PolyURunB3) [2] the threshold used in tweet filtering stage is based on the observation on the Tweet Stream for days before evaluation period. To improve performance in terms of nDCG-1, the system needs to identify silent day which can be achieved with better tweet filter setting. These results show that both approaches (ours and PolyURunB3) perform well when it comes to not pushing tweets for the silent day which improves performance in terms of nDCG1@10. However, for the eventful day, our method pushes more relevant and not redundant tweets than PolyURunB3 which explains the improvement of the performance in terms of nDCG0@10. These results are consistent with previous findings that our approach is more efficient for the event that catches a lot of attention in social media. In addition, we observer that our approach outperforms the best automatic TREC 2016 run (nudtsna) overall metrics. We found the performance improvements up to nDCG-1@10 and nDCG0@10 values of about 6.31% and 127.03% respectively.

These results reveal that our approach achieves a good balance between pushing too many tweets and pushing few tweets. These trends can be explained by first, constraints related to the temporal and the topical coverage allow to take into consideration the mutual relation between tweets which is not the case in the state-of-the-art approaches based on the selection of the top-k tweets. Second, the use of word embedding in computing the tweet-query relevance score leads to boost tweets that contain different terms but sharing the same semantic context with query terms whereas the state-of-the-art baselines are based on stream statistics. Third and last, our approach acts as a top-k selection method when the number of candidate tweets is less than the summary length or when there are no clusters that contain more than one tweet. Somehow, the top-k method can be considered as a particular case of the proposed ILP.

5 CONCLUSIONS

To tackle the task of tweet summarization for a long ongoing event, we introduced a new approach based on an

Table 1: Comparative of effectiveness with state-of-the-art baselines on RTS 2016 datasets.

Method	nDCG1@10	nDCG0@10	%
WSEBM-ILP	**0.2950**	**0.1201**	
HybridTF-IDF	0.1678 †	0.0767‡	+75.85%
TFIDF	0.1745†	0.0834‡	+69.10%
SUMBASIC	0.1655†	0.0536‡	+78.30%
TREC RTS 2016 official Results			
PolyURunB3	0.2898	0.0684†	+1.79%
nudtsna	0.2708	0.0529†	+6.31%
QUJM16	0.2621	0.0301†	+9.84%

Note. % indicates the proposed method improvements in terms of nDCG-1@10. The symbols *, †, and ‡ denote the Student test significance: $* \ 0.01 < t \leq 0.05$, † $t \leq 0.01$, ‡ $0.05 < t \leq 0.1$.

optimization framework to generate a periodic summary of tweet stream. The main contribution of the proposed method is that tweet selection problem is formulated as ILP that maximizes objective function based on the tweet's relevance score subject to a series of constraints related to redundancy, coverage, temporal diversity, and length limit. To enhance summary coverage, we take into account the topicality as well as the temporality of tweets in order to provide an overview of the development of the given event. In order to overcome the word mismatch issue in the computation of tweet-tweet similarity, we take advantage of word embedding model. Experimental results based on a real-word dataset revealed that the proposed approach outperforms the baseline methods and the best automatic TREC RTS 2016 systems. The results also showed that more improvements are achieved on the queries with eventful days in a tweet stream.

REFERENCES

[1] Abdelhamid Chellal, Mohand Boughanem, and Bernard Dousset. 2017. 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*. 655–661. https://doi.org/10.1007/978-3-319-56608-5_62

[2] Wenjie Li Haihui Tan, Dajun Luo. 2016. PolyU at TREC 2016 Real-Time Summarization. In *Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*.

[3] Juraj Hromkovic and Waldyr M. Oliva. 2002. *Algorithmics for Hard Problems* (2nd ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[4] David Inouye and Jugal K. Kalita. 2011. 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*. 298–306.

[5] S. Kullback and R. A. Leibler. [n. d.]. *The Annals of Mathematical Statistics* 1 (03 [n. d.]), 79–86.

[6] Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. 2011. Generating Aspect-oriented Multi-document Summarization with Event-aspect Model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1137–1146. http://dl.acm.org/citation.cfm?id=2145432.2145553

[7] Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, Richard McCreadie, and Tetsuya Sakai. 2015. Overview of the TREC 2015 Microblog Track. In *Text REtrieval Conference, TREC, Gaithersburg, USA, November 17-20*.

[8] Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. 2016. Overview of the TREC 2016 RealTime Summarization. In *Text REtrieval Conference, TREC, Gaithersburg, USA, November 15-18*.

[9] Fei Liu, Yang Liu, and Fuliang Weng. 2011. Why is "SXSW" Trending?: Exploring Multiple Text Sources for Twitter Topic Summarization. In *Proceedings of the Workshop on Languages in Social Media (LSM '11)*. 66–75.

[10] Charles L. A. Clarke Jimmy Lin Luchen Tan, Adam Roegiest. 2016. 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)*.

[11] Jimmy Lin Charles L. A. Clarke Luchen Tan, Adam Roegiest. 2016. An Exploration of Evaluation Metrics for Mobile Push Notifications. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. 745–754. https://doi.org/2911451

[12] Stuart Mackie, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. Comparing Algorithms for Microblog Summarisation. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings*. 153–159.

[13] Ryan T. McDonald. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*. 557–564.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[15] Douglas W. Oard Mossaab Bagdouri. 2015. CLIP at TREC 2015: Microblog and LiveQA. In *Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*.

[16] A. Nenkova and L. Vanderwende. [n. d.]. *The Impact of Frequency on Summarization*. Technical Report MSR-TR-2005-101. MSR-TR-2005-101. 8 pages.

[17] Tamer Elsayed Reem Suwaileh, Maram Hasanain. 2016. Lightweight, 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*.

[18] Zhaochun Ren, Shangsong Liang, Edgar Meij, and Maarten de Rijke. 2013. Personalized Time-aware Tweets Summarization. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13)*. 513–522.

[19] Gerard Salton, Edward A. Fox, and Harry Wu. 1983. Extended Boolean Information Retrieval. *Commun. ACM* 26, 11 (Nov. 1983), 1022–1036.

[20] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K. Kalita. 2010. Experiments in Microblog Summarization. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing (SOCIALCOM '10)*. 49–56.

[21] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. 2013. Sumblr: Continuous Summarization of Evolving Tweet Streams. In *the 36th International ACM SIGIR Conference (SIGIR '13)*. 533–542.

[22] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. 2012. Towards Real-time Summarization of Scheduled Events from Twitter Streams. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media (HT '12)*. 319–320.