Detecting Newsworthy Topics in Twitter

Steven Van Canneyt
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
steven.vancanneyt@ugent.be

Matthias Feys
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
matthias.feys@ugent.be

Steven Schockaert
Cardiff University
5 The Parade
Cardiff, United Kingdom
s.schockaert@cs.cardiff.ac.uk

Thomas Demeester
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
thomas.demeester@ugent.be

Chris Develder
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
chris.develder@ugent.be

Bart Dhoedt
Ghent University - iMinds
Gaston Crommenlaan 8
Ghent, Belgium
bart.dhoedt@ugent.be

Abstract

The task of the SNOW 2014 Data Challenge is to mine Twitter streams to provide journalists a set of headlines and complementary information that summarize the most newsworthy topics for a number of given time intervals. We propose a 4-step approach to solve this. First, a classifier is trained to determine whether a Twitter user is likely to post tweets about newsworthy stories. Second, tweets posted by these users during the time interval of interest are clustered into topics. For this clustering, the cosine similarity between a boosted *tf-idf* representation of the tweets is used. Third, we use a classifier to estimate the confidence that the obtained topics are newsworthy. Finally, for each obtained newsworthy topic, a descriptive headline is generated together with relevant keywords, tweets and pictures. Experimental results show the effectiveness of the proposed methodology.

In: S. Papadopoulos, D. Corney, L. Aiello (eds.): Proceedings of the SNOW 2014 Data Challenge, Seoul, Korea, 08-04-2014, published at http://ceur-ws.org

1 Introduction

Social media is an excellent source to detect events due to their large data volume, broad user base and real-time nature. Extensive work has shown that social media can successfully detect events [2, 4, 10, 15, 18], even before they are reported in traditional media [16, 17]. Therefore, social media may be an excellent source for news professionals to monitor the newsworthy topics that emerge from the crowd. However, we have to deal with noisy text fragments which are in addition often very short (e.g. Twitter posts).

In this paper, we propose our methodology for a solution to the SNOW 2014 Data Challenge. The task of this challenge is to automatically mine social streams to provide journalists with a set of headlines and complementary information that summarize the newsworthy topics for a number of timeslots (time intervals) of interest. For an overview of the details of this challenge, we refer to [11]. Given a stream of tweets and a time interval of interest, we first determine the users who posted the tweets during that time interval which are most likely to post about newsworthy stories. This is accomplished by a classifier trained on profile features of the users. Second, the tweets posted by these users are clustered into topics based on the cosine similarity of their boosted *tf-idf* representations. This boosting is considered, on the one hand, to raise the importance of bursty words. On the other hand, proper nouns and verbs are boosted as they are essential keywords in most discussed topics (e.g. topic subjects and actions). Third, several features of the obtained

topics are determined which are used to classify them as 'newsworthy' or 'not newsworthy'. Finally, for each detected newsworthy topic, a headline that summarizes the topic, accompanied by a set of relevant tweets, pictures and keywords are determined. The quality of the extracted newsworthy topics will be evaluated by a panel of news professionals selected by the challenge organizers. However, initial observations show the effectiveness of our methodology.

The remainder of this paper is structured as follows. We start with a review of related work in Section 2. Next, in Section 3, we describe our methodology for discovering newsworthy topics. Subsequently, Section 4 presents the experimental results. Finally, we conclude our work in Section 5.

2 Related work

There has been a lot of interest in detecting events and trending topics in social media. This research can be divided in two types of approaches. In the first type, social media documents (e.g. tweets) are clustered. This is referred to as *document-pivot*. An event or topic is thus represented by a cluster of documents. The second line of work first selects the most important words, which are then clustered. In this approach, referred to as *feature-pivot*, an event or topic is represented by a cluster of words.

Document-pivot approaches cluster social media documents by leveraging some similarity metric between them. TwitterStand [17], for instance, only uses the tweets of Twitter users who usually post news related tweets. They however did not use a classifier to determine these users, but manually constructed an initial set of these users. This set is updated based on the number of times the tweets of a user is associated with a newsworthy topic. Subsequently, an online clustering algorithm is used, which assigns the news related tweets to the closest cluster if the distance to this cluster is smaller than a given threshold. Otherwise, a new cluster with this tweet as the only member is created. The distance between a cluster and a tweet is based on the words in the tweet and the time at which the tweet was posted. The obtained clusters are considered as newsworthy topics. Finally, for each obtained topic, additional relevant tweets are searched using the hashtags present in the tweets of its corresponding cluster. Becker et al. [2] clustered social media documents based on their textual, time and location similarity features. They used a classifier with these similarity scores as features to predict whether a pair of documents belongs to the same cluster. To train the classifier, known clusters of social media documents were used which were constructed manually and by using the Upcoming database. When the probability that a document belongs to an existing cluster is smaller than a threshold, a new cluster is generated for this document. Becker et al. [3] introduced an additional step which classifies the clusters corresponding to candidate events as 'event' or 'non-event' based on e.g. the burstiness of the most important words in the clusters. Using the methodology described in [2, 3], the authors were able to detect events using Flickr and Twitter data.

Feature-pivot methods use statistical models to extract sets of words that are representative for the most important topics and events described in a corpus of documents. In [4], for example, the authors analyze the temporal and locational distributions of Flickr tag usage to detect bursty tags in a given time window, employing a wavelet transform to suppress noise. Afterwards, the tags are clustered into events such that each cluster consists of tags with similar locational distribution patterns and with similar associated photos. Finally, photos corresponding to each detected event are extracted. EDCoW [18] uses wavelet transformations to measure the bursty energy of each word used in Twitter posts, and then filters words with low energy in a given time window. Finally, the remaining words are clustered using modularity-based graph partitioning to detect events. Twevent [10] improved the approach of EDCoW by first splitting the incoming tweets in n-grams. An n-gram was then considered as an event segment in a given time window when the occurency of that n-gram was significantly higher than its expected occurency. The obtained event segments were finally clustered into events using Jarvis-Patrick clustering and ranked based on the importance of their event segments in Wikipedia. SocialSensor [1] selects the most bursty n-grams in a time window t based on their *df-idf*$_t$ score. This score is an adapted version of the *tf-idf* metric, penalizing n-grams whose popularity began in the past and which are still popular in the present. In addition, a boost factor is considered to raise the importance of proper nouns. The top ranked n-grams are then clustered using a hierarchical clustering algorithm and the co-occurences of the n-grams in the tweets. Finally, the clusters are ranked according to the highest *df-idf*$_t$ score of the n-grams contained by the cluster. They compared their approach with a standard feature-pivot, a standard document-pivot, and a Latent Dirichlet Allocation (LDA) approach. The document-pivot approach outperformed the feature-pivot and LDA approach. However, the quality of the top ranked topics was higher for their proposed approach than for the document-pivot approach. The authors also introduced two approaches which are based on Frequent Pattern Mining with similar or worse performance.

3 Methodology

For a stream of tweets (called test set, T^n), we want to determine the most newsworthy topics. In particular, for each time interval of interest $i \in I$, $m \geq 1$ newsworthy topics will be automatically extracted. To easily interpret the extracted topics, each topic will be in the form of a short headline that summarizes the topic, accompanied by a set of tweets, URLs of relevant pictures, and a set of keywords. To optimize the proposed methodology, we use a training set T^k of tweets with known newsworthy topics.

For a given stream of tweets T^n and a time interval $i \in I$, we first determine the users who posted the tweets during time interval i who are most likely to post about newsworthy stories. The tweets of these users are then clustered into topics. Thereafter, the obtained topics are ranked based on the confidence that they are newsworthy. Finally, for each detected newsworthy topic, the headline, most relevant tweets, tags and pictures are determined. The implementation of our methodology has been made publicly available to the research community.[1] In the rest of this section, we will explain each step in more detail.

3.1 News Publisher Detection

The first step of the proposed methodology is to estimate the likelihood that a Twitter user will post tweets about newsworthy topics. We indicate Twitter users who almost always publish newsworthy tweets as 'news publishers'. Examples are official twitter accounts of news papers, news programs or news websites. Given a set of tweets, the corresponding authors can then be ranked based on the probability that they are news publishers. Only tweets of the top ranked users will be used to detect newsworthy topics.

We first manually annotate 10 000 Twitter users as 'news publisher' or 'other'. We call this set of user U. Second, we use 5-fold cross-validation on the set U to find relevant user features and to train a classifier that optimizes the average precision of the users, which are sorted based on the likelihood that they are news publishers. As candidate classifiers, we consider all methods implemented in WEKA [8] as well as the Support Vector Machine (SVM) implementations of LibLinear [9]. The obtained features are shown in Table 1. The classifier which led to the largest average precision is a Bayesian belief network that uses a local K2 search algorithm [5].

Finally, user set U is used to train a Bayesian belief network which estimates the probability that the users which posted the tweets in test set T^n during time interval i are news publishers. The users with probability larger than α are considered as 'news publishers', noted as set P_i^n. Similarly, for each $i' \in I'$, the news publishers who posted tweets in the training set T^k during time i' are contained in the set $P_{i'}^k$. Set I' contains the considered time intervals corresponding to the training set T^k.

[1] https://github.com/svcanney/twittertopics

3.2 Topic Detection

In the second step of our methodology we cluster the tweets posted by users in P_i^n. Using only the tweets of news publishers, we significantly reduce the noisy tweets leading to 'junk'-topics. The clustering is performed using the DBSCAN [6] algorithm with parameters ϵ and minimum number of points required to form a cluster *minPts*.

As distance measure we use the cosine distance between the boosted *tf-idf* representations of the tweets. The boosted *tf-idf* value of a word w in tweet t posted during time interval i is given by

$$\textit{tf-idf}_i^w = \textit{tf-idf}^w \cdot \textit{E-boost}^w \cdot \textit{T-boost}_i^w \qquad (1)$$

Factor $\textit{tf-idf}^w$ is the standard term frequency-inverse document frequency for word w in tweet t. The document frequencies used for this $\textit{tf-idf}^w$ value are obtained from a set of tweets T^e which is unrelated to T^k and T^n. As T^k and T^n may contain tweets which are related to a specific event (see Section 4.1), we would have much lower $\textit{tf-idf}^w$ values for the event-specific words when these sets were used to calculate the document frequencies. Nonetheless, these words can be very relevant in the detected topics. By using an unrelated set of tweets, we are thus able to use more general event-independent document frequencies.

The first boosting factor $\textit{E-boost}^w$ is the boosting of proper nouns and verbs, similar as in [12], since they are typically more important than other words. The authors of [12] discovered that a boosting value of 1.5 for this kind of words and 1 for other words led to the best clustering results. Therefore, we use the same boosting values in this paper.

The second boosting factor $\textit{T-boost}_i^w$ is temporal boosting, in which we boost the words based on their relative document frequency in this time interval i versus the previous time intervals, thus the burstiness of the words. More concretely, we define

$$p_i^w = \frac{df_i^w}{N_i} \qquad (2)$$

as the relative frequency of word w in time interval i, with df_i^w the document frequency of the word in the time interval and N_i the total number of tweets posted during i. We boost each term with the following tem-

Table 1: Features used to detect Twitter accounts of news publishers.

Textual features	
username bag-of-words	term frequencies of the words in the user name
description bag-of-words	term frequencies of the words in the user description
Meta-data features	
#followers	number of followers
#following	number of following
$\frac{\#\text{follower}}{\#\text{following}+1}$	number of followers in comparison to the number of following
#tweets	number of tweets the user posted
#favorites	number of tweets the user favorited
#lists	number of lists the user follows
verified?	is the user account verified or not?
URL?	contains the user profile an URL or not?

poral boosting factor:

$$T\text{-}boost_i^w = \frac{p_i^w}{p_{0,i-1}^w} \tag{3}$$

with $p_{0,i-1}^w$ the exponential moving average of the relative frequencies of the word w for the time intervals 0 until $i-1$, using a smoothing factor λ.

Finally, we define the center of a cluster $c \in C_i^n$ as vector $center_c$, obtained by averaging out all boosted *tf-idf* representations of the tweets in cluster c.

The detected topics from tweet test set T^n during time interval i are given by C_i^n. Similarly, the detected topics of training set T^k during interval $i' \in I'$ are given by $C_{i'}^k$. Additionally, we define set $C^k = \bigcup_{i'} C_{i'}^k$.

3.3 Topic Ranking

We explore different features to describe the detected clusters of C_i^n in order to identify newsworthy topics. A classifier trained on C^k is then used to detect newsworthy topics in the set of clusters C_i^n during interval i, indicated by the set S_i^n.

The training set of detected topics C^k is used to find the optimal features and classifier. Similar to the approach described in Section 3.1, we consider all methods implemented in WEKA [8] as well as the Support Vector Machine (SVM) implementations of LibLinear [9] as candidate classifiers. We first manually label the topics in training set C^k as 'newsworthy' or 'not newsworthy'. Second, C^k is partitioned into two disjoint subsets of topics, based on their time intervals: development set C^d comprises the first two thirds, the validation set C^v the last third. The topics of the development set C^d are used to train a classifier. This classifier is then used to estimate the likelihood that a topic $c \in C^v$ is newsworthy. For a particular time interval, the corresponding topics can then be ranked based on this likelihood. The objective is thus to optimize the mean average precision of these rankings. The obtained features are shown in Table 2. These features are divided in four categories. The first category takes the number of tweets in the clusters and their type into account. For instance, a cluster with just a few associated tweets may not be related to a newsworthy topic. The second category considers the features of the users. If the users who posted the tweets in the clusters are very likely to be news publishers (e.g. with probability higher than 0.9), the cluster probably corresponds to a newsworthy topic. The third category of features describes the topical coherence of the cluster, based on the hypothesis that newsworthy clusters tend to address a central topic, whereas noisy non-newsworthy topics cover more heterogeneous topics. The last category of features is used to exclude clusters corresponding to a topic that was already detected in a previous time interval, as we consider topics only as newsworthy when they occur for the first time. The classifier that leads to the highest mean average precision is Support Vector Machines (SVM) trained using sequential minimal optimization [13].

3.4 Topic Enrichment

The final step in our methodology is the topic enrichment. This step starts from each obtained newsworthy topic $s \in S_i^n$ and generates a headline, extracts keywords, a list of associated tweets and a list of pictures. These steps are mostly handled individually and are discussed in the following subsections.

3.4.1 Headline Creation

The headline of newsworthy topic s is constructed as a cleaned up version of the most representative tweet sentence in the set of tweets related to s. These tweet sentences are obtained by splitting each tweet in tweet sentences based on the presence of punctuation marks, and only retaining sentences containing at least one verb. To retrieve the most representative tweet sentence, we select the sentence with maximum cosine similarity between its boosted *tf-idf* representation and the vector associated with the topic center $center_s$. Subsequently, we apply a set of rules to clean the obtained sentence: (1) Removing the mentions of users if they are part of a retweet mention. (2) Removing all URLS and emoticons. (3) Removing hashtags if they do not syntactically belong in the sentence. (4) Removing the '@' and '#'-symbols from the remaining hashtags and user mentions. (5) Removing parts of sentences inside parentheses. (6) Splitting the camel

Table 2: Features used to detect newsworthy topics.

Tweet features	
#tweets	number of tweets in the cluster
%original tweets	percentage of tweets in the cluster which are original tweets
%retweets	percentage of tweets in the cluster which are retweets
%replies	percentage of tweets in the cluster which are replies
%mentions	percentage of tweets in the cluster which contains user mentions
User features	
#users	number of users who posted the tweets in the cluster
%news publishers	percentage of users whose probability that they are news publishers is larger than x, with $x \in \{0.6, 0.7, 0.8, 0.9\}$
Topical coherence features	
%topic tweets (1)	percentage of tweets in the cluster containing the word of $center_c$ with highest $tf\text{-}idf_i^w$ value
%topic tweets (2)	percentage of tweets in the cluster containing the word of $center_c$ with second highest $tf\text{-}idf_i^w$ value
%topic tweets (3)	percentage of tweets in the cluster containing the word of $center_c$ with third highest $tf\text{-}idf_i^w$ value
Non duplicates features	
max similarity (1)	highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters
max similarity (2)	second highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters
max similarity (3)	third highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters
max similarity (4)	fourth highest cosine similarity between the cluster center and the center of previous detected newsworthy clusters

case words into different words. (7) End the headline with a punctuation mark.

3.4.2 Keywords Extraction

The keywords are chosen as the words present in the headline which are in the top 50% of the most important words associated to topic s. This importance of a word w is given by its $tf\text{-}idf_i^w$ value in $center_s$.

3.4.3 Representative Tweets extraction

To extract a representative set of tweets, we first expand the list of tweets related to our topic by including tweets from users which are not indicated as 'news publishers'. In particular, we consider all tweets in T^n posted during i with a cosine similarity between their boosted *tf-idf* representation and the center of the topic which is higher than ω. Next, these tweets are ordered based on their relevance to the topic, denoted as $relevance_s^t$. The $relevance_s^t$ value of tweet t is defined as the cosine similarity between its boosted *tf-idf* representation and the center of the topic $center_s$, multiplied by the *user_factor*. This factor is $\upsilon \geq 1$ if the user who posted tweet t is indicated as a 'news publisher' and 1 otherwise. This ordered list of tweets related to topic s is denoted by T_s^n.

The tweets associated with a single topic should be sufficiently different from each other, therefore we discard tweets in T_s^n which are near-duplicates of tweets that are ranked higher in the list. To measure the similarity between the tweets in T_s^n, we use the cosine similarity between the non-boosted version of the *tf-idf* representations of the tweets. In particular, tweets are considered as 'near-duplicates' if their similarity is higher than φ. We discard boosting in this step, since the goal of boosting was to increase the impact of the topic-related words, thereby diminishing the impact of the other words in the tweet. However, the tweets in T_s^n are all related to the same topic, and all contain these topic-related words leading to a high cosine similarity of their boosted *tf-idf* representations, mainly caused by the presence of these topic-related words. As we want to obtain a coherent diverse set of tweets describing this topic, we want tweets that contain these topic-related words, but have a significant number of different non-topic-related words. If we had used the boosted *tf-idf*, the cosine similarity would almost only be impacted by the number of matching topic-related words. Finally, the top 5 tweets of this filtered T_s^n list are considered as representative for topic s.

3.4.4 List of pictures

In order to obtain a full list of pictures related to topic s, the tweets of T_s^n containing the same picture URL are grouped. Picture URLs are obtained by using the media entities associated with the tweets. The picture URLs are then sorted based on the sum of the $relevance_s^t$ values of the tweets containing the URL. Finally, the top 5 picture URLs are considered as relevant to topic s.

4 Evaluation

4.1 Data Acquisition and Settings

In order to evaluate our approach, we crawled the Twitter posts meta-data of the given Twitter id's related to the 2012 US elections event posted on Twitter between November 6, 2012 23:30 GMT and November 7, 2012 7:00 GMT (training set, T^k). The test set T^n contains tweets related to the Syria, Ukraine, terror and bitcoin-problems mentioned on Twitter between February 25, 2014 18:00 GMT and February 26, 2014 18:00 GMT. More details about the training and test set can be found in [11]. Additionally, an unrelated set tweets was obtained from the sample-stream of the Twitter Streaming API from November 29, 2013 until February 5, 2014 (external set, T^e). Non-English tweets were removed using LDIG[2]. To calculate the term frequencies in the obtained tweets, TweetNLP [7] was used to tokenize the tweets and to remove words

[2]https://github.com/shuyo/ldig

related to punctuations, URLs, determiners, etc. The obtained words were then transformed to lower case and words with fewer than three characters were removed. Finally, the words were Porter stemmed [14]. As a result of this process, we obtained 928 791 tweets for training our methodology (training set, T^k), 973 658 tweets for evaluating our methodology (test set, T^n), and 77 741 801 tweets which have been used as external set T^e. User set U contains 10 000 Twitter users who are randomly selected from the users who posted the tweets in T^e. The time intervals of interest for the test set and training set are given by the challenge organizers and are respectively 15 minutes and 10 minutes long. We empirically set $\alpha = 0.04$, $\epsilon = 0.4$, $minPts = 3$, $\lambda = 0.5$, $\omega = 0.6$, $\upsilon = 1.5$ and $\varphi = 0.7$.

4.2 Experimental Results

4.2.1 News Publisher Detection

As described in Section 3.1, we use 5-fold cross-validation on the user set U to optimize and evaluate the methodology which detects news publisher. User set U contains 10 000 Twitter users who are manually annotated as 'news publisher' or 'other'. As a result of this process, 1.64% of the users were labeled as 'news publisher'. The proposed methodology to rank users based on the likelihood that they are news publishers resulted in an average precision of 88.83%. In general, 99.41% of the users in U were correctly classified, which is significantly higher than the 98.36% accuracy when all users are classified as 'other' (sign test, $p < 0.001$).

4.2.2 Topic Ranking

The training set of detected topics C^k is used to optimize and evaluate the topic ranking methodology, as described in Section 3.3. Set C^k contains 116 manually annotated clusters, of which 54 are labeled as 'newsworthy'. For each considered time interval i' corresponding to clusters in validation set C^v, the clusters of C^v associated with i' are ranked based on the confidence that they are related to a newsworthy topic. The mean average precision of these rankings is 99.17%. In general, 82.05% of the clusters in the validation set were classified correctly.

4.2.3 Methodology Performance

Our methodology extracted 433 newsworthy topics from the test set, given by set $S^n = \bigcup_i S_i^n$. The newsworthy topics of time intervals February 26, 2014 09:15 until 10:15 GMT are shown in Table 3. These results show the effectiveness of our methodology to discover newsworthy topics in Twitter. As we only use tweets posted by 'news publishers' to detect topics, most of the discovered topics are indeed newsworthy. However, we observe that some duplicates are not removed mainly because users sometimes discuss one topic in different words, i.e. the high similarity of these topics can not be detected using cosine similarity on their associated words (e.g. topic 7 and 10). In addition, some non-newsworthy topics were incorrectly extracted due to users who are classified as 'news publisher' who post non-newsworthy content (e.g. topic 15). Finally, we observe that the obtained headlines are informative and that they are constructed in a syntactically correct way.

The extensive summary of the newsworthy topics extracted during time interval February 26, 2014 09:15 can be found in Table 4. We observe that the representative tweets for a particular topic are sufficiently different from each other, i.e. no near-duplicates or retweets are given. Additionally, we note that the coherence of the tweets associated with topic 2 is higher than the coherence of the tweets associated with topic 1. In particular, topic 2 covers one clear topic (i.e. about Sofia monument's makeover), in contrast, topic 1 covers very similar, but different, topics (i.e. about a military vehicle in Kiev, Ukraine; and about a military vehicle in Sevastopol, Ukraine). The discovered pictures related to these newsworthy topics are shown in Figure 1. In total, 24% of the discovered newsworthy topics contains at least one related picture.

The newsworthy topics in S^n and their summaries are evaluated across a mixture of quantitative and qualitative dimensions by a panel of news professionals selected by the SNOW 2014 Data Challenge organizers. These official evaluation results of our methodology are included in [11].

5 Conclusions

We proposed a methodology which automatically mines Twitter streams to provide journalists with a set of headlines and complementary information that summarizes the most important topics for a number of time intervals of interest. As we are only interested in newsworthy topics, we only use tweets of users who are classified as 'news publishers'. These tweets are then grouped into topics using a DBSCAN clustering algorithm, whereby the similarity between the tweets is determined using the cosine similarity on their boosted *tf-idf* representations. Thereafter, a classifier is trained to estimate which of the detected topics is newsworthy. Finally, for each obtained newsworthy topic, a descriptive headline, together with relevant tweets, keywords and pictures is determined. Experimental results show the effectiveness of the proposed methodology.

Table 3: Automatically extracted newsworthy topics from Twitter.

nr	time interval	headline
1	26-02-14 09:15	Jubilant protesters driving military vehicle from a Kiev Museum around Parliament building.
2	26-02-14 09:15	Sofia monument's latest makeover provokes protest from Russia.
3	26-02-14 09:30	I'm in Charge of Miltary Now, Ukraine's Interim President Says.
4	26-02-14 09:30	GDP grew 0.7% in Q4, unrevised from preliminary estimate.
5	26-02-14 09:30	Russia urges OSCE to condemn "neo-fascist" sentiment in west Ukraine.
6	26-02-14 09:30	The price of Bitcoin on MT. Gox is US $135.0000.
7	26-02-14 09:30	Bitcoin Has Made A Really Impressive Recovery.
8	26-02-14 09:45	Ukraine minister disbands Berkut riot police blamed for violence.
9	26-02-14 09:45	Japanese Authorities Probing Collapsed Bitcoin Exchange.
10	26-02-14 09:45	How bitcoin can turn it around.
11	26-02-14 09:45	Hezbollah says Israel bombed its positions near Syrian border 2 days ago, vows response.
12	26-02-14 10:00	Russia's deputy finance minister says no multilateral talks on financial aid to Ukraine are taking place.
13	26-02-14 10:00	Japan donates $14 mil. for Syria weapons disposal.
14	26-02-14 10:15	This is Beijing, less than three weeks apart.
15	26-02-14 10:15	Your spring tweet has appeared in our latest Edition mag.
16	26-02-14 10:15	Ukraine 'set to unveil new government'.

Table 4: Summaries of extracted newsworthy topics during time interval 26-02-14 09:15.

nr	tags	representative tweets
1	Jubilant,protesters,driving, vehicle,Museum,Parliament	Jubilant protesters driving military vehicle from a Kiev Museum around Parliament building #Kiev #Ukraine Another #Russia—n armored vehicles spotted in #Sevastopol in #Crimea. #Ukraine http://qn.quotidiano.net/esteri/2014...
2	Sofia,monument, makeover,provokes	Pro-Ukraine paint job - Sofia monument's latest makeover provokes protest from Russia http://bbc.in/1frf9UN Kijw w Sofii. RT: @BBCWorld Pro-Ukraine paint job in Sofia provokes protest from Russia http://bbc.in/1frf9UN Pro-#Ukraine paint job-Sofia monument's latest makeover provokes #protest from R http://bbc.in/1frf9UN via @BBCWorld

(a) **(b)** **(c)**

Figure 1: Pictures related to newsworthy topic number 1 (a,b) and number 2 (c).

6 Acknowledgments

Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology (IWT).

References

[1] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, and A. Jaimes. Sensing trending topics in Twitter. *IEEE Transactions on Multimedia*, 15(6):1268–1282, 2013.

[2] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proc. of the 3rd ACM Int. Conf. on Web Search and Data Mining*, pages 291–300, 2010.

[3] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on Twitter. In *Proc. of the 5th Int. AAAI Conf. on Weblogs and Social Media*, pages 438–441, 2011.

[4] L. Chen and A. Roy. Event detection from Flickr data through wavelet-based spatial analysis. In *Proc. of the 18th ACM Conf. on Information and Knowledge Management*, pages 523–532, 2009.

[5] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[6] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[7] K. Gimpel, N. Schneider, B. O. Connor, and D. Das. Part-of-speech tagging for Twitter: Annotation, features, and experiments. *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 42–47, 2010.

[8] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software : An update. *SIGKDD Explorations*, 11(1), 2009.

[9] S. Keerthi, S. Sundararajan, and K. Chang. A sequential dual method for large scale multi-class linear SVMs. In *Proc. of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 408–416, 2008.

[10] C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *Proc. of the 21st ACM Int. Conf. on Information and Knowledge Management*, pages 155–164, 2012.

[11] S. Papadopoulos, D. Corney, and L. M. Aiello. SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media. In *Proceedings of the SNOW 2014 Data Challenge*, 2014.

[12] S. Phuvipadawat and T. Murata. Breaking news detection and tracking in Twitter. In *Proc. of the 2010 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 120–123, Aug. 2010.

[13] J. Platt. Fast training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. 1998.

[14] M. Porter. An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 14(3):130–137, 1980.

[15] T. Reuter and P. Cimiano. Event-based classification of social media streams. In *Proc. of the 2nd ACM Int. Conf. on Multimedia Retrieval*, page 22, 2012.

[16] T. Sakaki. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proc. of the 19th Int. Conf. on World Wide Web*, pages 851–860, 2010.

[17] J. Sankaranarayanan, B. E. Teitler, and H. Samet. TwitterStand: News in tweets. In *Proc. of the 17th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pages 42–51, 2009.

[18] J. Weng, Y. Yao, E. Leonardi, and F. Lee. Event detection in Twitter. In *Proc. of the 5th Int. AAAI Conf. on Weblogs and Social Media*, pages 401–408, 2011.