2016 IEEE 28th International Conference on Tools with Artificial Intelligence

Online Clustering for Topic Detection in Social Data Streams

Carmela Comito, Clara Pizzuti, Nicola Procopio
National Research Council of Italy (CNR)
Institute for High Performance Computing and Networking (ICAR)
Via Pietro Bucci, 7-11C
87036 Rende (CS), Italy
Email: {carmela.comito,clara.pizzuti,nicola.procopio}@icar.cnr.it

***Abstract*—Microblogs have become an important origin of information regarding events happening in a location during a time period. Analyzing and clustering these streams of short textual messages is an important research activity which is attracting the interest of both public and private organizations, since the extracted knowledge can be exploited to enhance the comprehension of people behavior and the onset of emergency situations. Clustering these streams requires efficient algorithms capable of analyzing this continuos deluge of data. The paper proposes an online algorithm that incrementally groups tweet streams into clusters. The approach summarizes the examined tweets into the cluster centroids generated so far. The assignment of a tweet to a centroid uses a similarity measure that takes into account both the cluster age and the terms occurring in the tweet. Experiments on messages posted by users in the Manhattan area show that the method is able to extract events effectively taking place in the examined period.**

***Keywords*-Twitter, online clustering, topic detection**

I. Introduction

Social media, in the last years, gained an increasing popularity and became an important source of information diffusion and communication. Twitter, especially, offers a microblogging service that allows users to broadcast messages, of up to 140 characters, that can rapidly be reached by a huge number of people. The growing number of users, that post personal opinions regarding discussion topics and report information on events happening in real-time, generates an enormous quantity of data that can be analyzed to extract knowledge not only regarding user behavior, but also events related to cultural exhibition, sport, rallies, emergency situations due to natural disasters, or situations that may put people lives in jeopardy. The analysis of the high speed textual streams posted by Twitter users can thus be a fundamental help for discovering the arise of events. By leveraging the knowledge derived from these events, it is possible to understand what people are interested in a time period and what they are doing, by giving the opportunity to both users and authorities to make informed decisions.

In this paper we propose an online and incremental method for clustering streams of tweets with the aim of detecting interesting topics, eventually related to emergency events. Analogously to classical data stream approaches [1], clustering tweet streams which are continuously generated over time poses two main challenges for the development of efficient methods. The first is that tweets can be examined only once. Moreover, since user's interests regarding particular topics change with time, some clusters can become obsolete and should be discarded if no new tweet is added after a time period.

In order to deal with the former problem, the information gathered by all the similar tweets, that is those dealing with a common subject, are summarized into the centroid of the cluster they have been assigned to. Each centroid is characterized by a number of features, such as the time stamps of the cluster creation and that of the last update, along with the set of terms appearing in the tweets examined so far and belonging to that cluster. A novel similarity measure between a tweet and a centroid is defined, which is a based on an new data signature and takes into account both term frequency and temporal closeness. To guarantee greater value to more outstanding topics, a *fading function*, that diminishes the importance of clusters that do not continue to increase for a time period, is introduced. The concept is alike that defined for data points when clustering text data streams [1], however in our approach the decay rate of the importance of historical data is associated with clusters, instead of data. A thorough experimentation on the tweets posted by users in Manhattan during the month of December 2014 shows the method is capable to group tweets addressing common issues.

The paper is organized as follows. In the next section a brief overview of the most recent proposals for discovering topics in social data is given. Section III introduces the concepts used to model the clustering problem of tweet streams. Section IV describes the online algorithm and defines the similarity measure between a tweet and a centroid. Section V reports the experimental results obtained by executing the method on the tweets collected on December 2014 in the Manhattan area. Finally Section VI concludes the paper and discusses future developments.

2375-0197/16 $31.00 © 2016 IEEE
DOI 10.1109/ICTAI.2016.59

II. Related Work

In the last years there has been a significant research effort on detecting topics and events in social media. Generally, the approach is to extract different data features from social media streams and then summarize such features for event detection tasks by exploiting the information over content, temporal, and social dimensions.

The problem of determining topics and events in social streams is closely related to the problem of stream clustering which has been studied extensively in the context of the topic detection and tracking problem [2], [3], [4], [5]. In the following some of the most recent techniques are described. A comprehensive overview can be found in [6]. More in detail, existing work on streaming clustering techniques for social media can be classified into three categories:

(1) **Tweet-based clustering** [7], [8], which is an extension of the traditional text clustering by proposing advanced similarity measures to deal with the fact that tweets are short and noisy. For example, Aggarwal and Subbian [7] proposed an online clustering method which use the content, structural and temporal information in a holistic way in order to detect relevant clusters and events in social streams. Yin et al. [8] presented an online clustering method for topic discovery, inspired by Yang et al. [3]. They represent the textual content of tweets by a traditional vector-space model, in which a tweet is a vector of words $(v_1, v_2, \ldots, v_d)$, where d is the size of word vocabulary and v_j is the weight of j-th term in the tweet computed through the *Term Frequency-Inverse Term Frequency TF-IDF* concept. The authors use the cosine and Jaccard similarities to compute the closeness of a tweet to a centroid, multiplied by a time factor. Each new incoming tweet is assigned to the cluster whose centroid has minimum distance, then the centroid of a cluster C is updated as the normalized vector sum of all the tweets in it.

(2) **Burst-keyword-based clustering** [9], [4]. These works focus on bursty keywords, instead of single tweets covering only events with high burstyness. In [9] an approach to detect a target event by monitoring tweets in Twitter is proposed. Tweets are searched and classified using a support vector machine. A target event is detected by a temporal model which is constructed as a probability model. Location of a certain event is estimated by the Bayesian filters. This method is designed for a certain event and customized for earthquake application.

(3) **Hashtag-based clustering** defines an event as a cluster of hashtags. The most relevant works are [10], [11]. However, [11] considers hashtags as static documents, while the work in [10] deals with content-evolving hashtags. Specifically, in [10] Feng et al. focused on hierarchical spatio-temporal hashtag clustering techniques. The system has the following features: (1) Exploring events with different space granularity. (2) Exploring events with different time granularity. (3) Single-pass algorithm for event identification, which provides hashtag clusters. (4) Event ranking which aims to find burst events and localized events given a particular region and time frame. To support aggregation with different space and time granularity, they proposed a data structure, which is an extension of the data cube structure from the database community with spatial and temporal hierarchy.

Analogously to the work of Yin et al. [8], our approach is inspired by the method of Yang et al. [3]. However, only the basic structure is similar while the differences are meaningful in several aspects and can be considered as important contribution to the topic. First of all, we propose a novel compact data summarization structure of tweets and cluster centroids that dynamically builds the term vocabulary and includes different tweet features, not limited to textual information. Then, a new similarity measure, based on the Jaccard measure, that takes into account the importance of words, hashtags, and mentions according to their frequency is introduced. Finally, a novel fading function for accounting historical data in the similarity measure is presented. Most of the work in literature, including [8], to represent textual content of tweets use a traditional vector-space model, in which a tweet is represented as a vector of words, where the vector dimension is the size of word vocabulary using the TF-IDF approach. In streaming scenarios, because word vocabulary dynamically changes over time, it is very computationally expensive to recalibrate the inverse document frequency of TF-IDF. Differently, we deal with content evolving signature structure of cluster centroids by either updating frequencies of already present terms, or including new terms; since we refer to term frequencies as relative values, there is no need of recalibrating the vocabulary size. Moreover, another limitation of current approaches is treating tweets text without considering data features (e.g., hashtag, mention) that instead allow us to better catch tweets semantics and, consequently, improve the similarity measure.

These concepts will be clearer in the next two sections, that describe the novel data representation and the online clustering method.

III. Problem formulation and Model

In this section we introduce the notations and definitions supporting the formulation of the problem of topic mining from social data streams. We first define the concepts of social objects and social data streams.

Definition 1. *A tweet tw posted by a user u at time t from a location l, can be represented as a tuple, denominated* social object*, $so = (o, u, t, l, sgn)$ where o is the object identifier and $sgn = (w_u, w_b, h_u, h_b, m_u, m_b)$ is a feature vector, called* signature*. Each feature is a list of items defined as follows:*

- *w_u are the words appearing in the tweet;*

- w_b *are the word bigrams;*
- h_u *are the hashtags appearing in* tw*;*
- h_b *are hashtag bigrams;*
- m_u *are the mentions contained in the tweet;*
- m_b *are mention bigrams.*

The location l is expressed in terms of the geographic coordinates (latitude and longitude). The signature sgn of a tweet tw summarizes its content by extracting words, hashtags and mentions from the text, along with the item bigrams, i.e. the adjacent item pairs appearing in the tweet.

An example of social object signature built from a tweet is the following.

Example 1. *The tweet* `"Christopher Walken is making me SO NERVOUS as Captain Hook...and I love it #PeterPanLive"`*, was posted in New York on December 5th, after a television special that was broadcast by NBC on December 4, 2014, regarding a musical adaptation of Peter Pan in which Christopher Walken had the role of Captain Hook. The signature of this tweet is* $w_u = \{$ *Christopher, Walken, making, NERVOUS, Captain, Hook, love*$\}$, $w_b = \{$*Christopher Walken, Walken making, making nervous, nervous Captain, Captain Hook, Hook love*$\}$, $h_u = \{\#PeterPanLive\}$, $h_b = m_u = m_b = \emptyset$.

A social data stream is defined as a sequence of social data objects:

Definition 2. *A social data stream (*sds*) is a continuous and temporal sequence of social objects* $so_1 ... so_r ...$ *, generated by social media users from an initial time* t_0.

The centroid CC of a cluster C has a representation similar to that of a tweet. However, it must be a representative of all the tweets examined so far and assigned to C. Thus it has to take into account not only the items occurring in the tweets, but also their frequencies.

Definition 3. *The centroid of a cluster* C *is a tuple* $CC = (c, t_0, t_c, sgn_C)$*, where c is the cluster label,* t_0 *is the creation time of the cluster,* t_c *is the time stamp of the last time a social object was added to C, and* $sgn_C = (sgn, ff)$*.* sgn *is the feature vector analogous to the tweet signature, while* $ff = (f_{w_u}, f_{w_b}, f_{h_u}, f_{h_b}, f_{m_u}, f_{m_b})$ *is the list of frequencies corresponding to the signature.*

Notice that the difference between the signature of a tweet and that of a centroid is that each feature in the former contains the list of items appearing in the tweet, while for the latter it is the list of items occurring in all the tweets assigned to C, without repetitions.

A social data stream is characterized by a high, and often rapid, variation of the contents posted by users. As time progresses, the arrival of a tweet reporting on a subject never appeared before is deemed as a *new topic*, thus a new cluster is created from this object.

A cluster that has continued to receive social objects in the recent past is said *active*. If it also keeps a growing rate over a temporal horizon it means that the topic of discussion is judged interesting and attracts many new users. Users, however, may loose interest on a particular topic. Thus, when no new object is assigned to a cluster C for a time period, the cluster becomes *inactive* and it is removed from the list of clusters obtained so far. The time period depends on the application domain, thus it must be determined by the user needs. To this end we define the *life span* of a cluster as the time period between the last centroid update and the time the cluster has been generated.

Definition 4. *The life-span of a cluster* C *is defined as* $lf = \delta + 2^h \times (t_c - t_0)$*, where* t_0 *is the creation time of the cluster and* t_c *is the time of the last object assigned to* C*.* δ *and* h *are user defined parameters.* δ *is the minimum time period a cluster is maintained even if no new tweet is added to it, while* h *determines the temporal horizon a cluster is considered active. The decay rate of a cluster is defined as* $\lambda = 1/lf$

Analogously to the concepts introduced by Aggarwal and Yu [1] for clustering data streams, a time-dependent *fading function* that diminishes the importance of a cluster, is defined as follows.

$$f(t_c) = 2^{-\lambda(t_{so} - t_c)} \tag{1}$$

where t_{so} is the publication time of the post tw that could be added to C and λ is the *decay rate* of a cluster. λ establishes the importance of the historical data in the social streams. The lower the value of λ, i.e. the higher the h value, the higher the importance of the historical information maintained in the cluster since the longer the life-span of a cluster.

A cluster C is considered *inactive*, and removed from the list of currently active clusters, if no new objects arrive during its life-span. This means that $t_{so} - t_c \geq lf$, that is $f(t_c) \leq 0.5$. In the next section, we will describe the online clustering method and how cluster centroids are updated.

IV. Online clustering algorithm

In this section, first the method is described, then the new similarity measure used for assigning a tweet to a cluster is defined.

A. Methods

The online clustering algorithm is an incremental algorithm that sequentially processes the incoming social data streams and groups similar tweets into the same cluster, each cluster characterizing a topic of discussion. The method is based on the algorithm proposed by Yang et al. [3], suitably modified for the social data context.

The pseudo-code of the algorithm is reported in Figure 1. The first time the algorithm receives a tweet tw_1 from the stream, it generates the first cluster C_1 from the social object so_1 representing tw_1, by storing in the centroid the signature of so_1 and setting the corresponding frequencies to 1. Moreover C_1 receives the time stamp of so_1. After that, while a new tweet tw_i arrives at time t_{so_i}, the algorithm builds the representation so_i of tw_i and computes the similarity between the tweet and the clusters active at the time stamp t_{so_i}. If a cluster is inactive, it is removed from the set of clusters. If C_c is the cluster whose centroid has maximum similarity with so_i, and this similarity value is higher than the fixed threshold ϵ, the content of the social object is added to C_c by updating the centroid. Otherwise a new cluster is generated from so_i. These steps are repeated until the algorithm receives new tweets.

Two important steps of the algorithm are the centroid maintenance and the similarity function. The pseudo-code of the method to update a centroid $CC = (c, t, sgn_C)$ when a new tweet tw is added to any cluster C is described in Figure 2. Let $so = (o, u, t_{so}, l, sgn)$ be the social object representing tw. First of all, the time stamp of C is updated with the time stamp t_{so} of so. As regards the signature, all the items of each feature must be checked if already present in the centroid signature. Thus the intersection I and the difference D between the signatures of so and CC are computed. Then, for each feature $so.sgn(i)$ of the signature, if an element of this feature already appears in the feature $sgn_C.sgn(i)$ of the centroid, the corresponding frequency must be incremented by 1, otherwise, it will be added to the centroid feature and its frequency is set to 1.

B. Similarity measure

The similarity measure used by a method to group objects in the same cluster plays a key role for any clustering algorithm. Since our aim is to group social data concerning the same topic, devising the proper similarity function is crucial for the effectiveness of the algorithm. To this purpose, we can make the following observations:

- Stream data objects discussing the same topic are usually temporally close, suggesting the use of a combined measure of lexical similarity and temporal proximity as a criterion for clustering.
- Words, hashtags and mentions appearing more frequently should have a higher weight when computing the similarity
- A significant change in the lexicon and in term frequency are reliable indicators that the incoming stream reports on a new topic.

According to these observations, we propose a novel measure based on the Jaccard similarity that takes into account the lexical similarity of terms, their frequency, and the time distance.

Algorithm Online clustering
Input: A continuos stream of tweets $\mathrm{tw_1}, \ldots, \mathrm{tw_n}, \ldots$
a similarity threshold ϵ
Output: The set of currently active clusters $\mathcal{C}$

begin
$\mathcal{C} \leftarrow \emptyset$;
i=1;
$\mathrm{so_1} \leftarrow$ GenerateSocialObject($\mathrm{tw_1}$);
$\mathrm{C_1} \leftarrow$ CreateCluster($\mathrm{so_1}$);
$\mathcal{C} \leftarrow \mathcal{C} \cup \mathrm{C_1}$;
while (not end of stream)
$\mathrm{i} \leftarrow \mathrm{i} + 1$;
Receive the next tweet $\mathrm{tw_i}$;
$\mathrm{so_i} \leftarrow$ GenerateSocialObject($\mathrm{tw_i}$);
for each cluster $\mathrm{C_j} \in \mathcal{C} = \{C_1, \ldots, C_k\}$
if isActive($\mathrm{C_j}$) **then**
$\mathrm{sim(so_i, C_j)} \leftarrow$ ComputeSimilarity($\mathrm{so_i}$,$\mathrm{C_j}$);
else
$\mathcal{C} \leftarrow \mathcal{C} - \mathrm{C_j}$;
end
end
c=argmax$_{j \in \{1,\ldots,k\}}$ sim(so_i, C_j)
if $\mathrm{sim(so_i, C_c)} >= \epsilon$ **then**
updateCentroid($\mathrm{C_c}, \mathrm{so_i}$);
else
$\mathrm{C_i} \leftarrow$ CreateCluster($\mathrm{so_i}$);
$\mathcal{C} \leftarrow \mathcal{C} \cup \mathrm{C_i}$;
end
end while;
end

Figure 1. Online clustering algorithm.

Algorithm UpdateCentroid
Input: a social object $\mathrm{so} = (\mathrm{o}, \mathrm{u}, \mathrm{t_{so}}, \mathrm{l}, \mathrm{sgn})$
the cluster centroid $\mathrm{CC} = (\mathrm{c}, \mathrm{t_0}, \mathrm{t_c}, \mathrm{sgn_C})$
Output: an updated cluster centroid CC

begin
$\mathrm{CC.t_c} = \mathrm{so.t_{so}}$
for i=1 to 6 do
let $\mathrm{I} = \mathrm{so.sgn(i)} \cap \mathrm{CC.sgn_C.sgn(i)}$
$\mathrm{D} = \mathrm{so.sgn(i)} - \mathrm{I}$
$\mathrm{CC.sgn_C.sgn(i)} = \mathrm{CC.sgn_C.sgn(i)} \cup \mathrm{D}$
for each item in I, add 1 to the corresponding $\mathrm{CC.sgn_C.ff(i)}$
for each item in D, set to 1 to the corresponding $\mathrm{CC.sgn_C.ff(i)}$
end
return CC;
end

Figure 2. Update centroid procedure.

Let $so = (o, u, t_{so}, l, sgn)$ and $CC = (c, t_0, t_c, sgn_C)$ be a social object and a centroid, respectively, with $sgn = (w_u, w_b, h_u, h_b, m_u, m_b)$ the signatures of so, while $sgn_C.sgn = (w_u^C, w_b^C, h_u^C, h_b^C, m_u^C, m_b^C)$ and $sgn_C.ff = (f_{w_u}, f_{w_b}, f_{h_u}, f_{h_b}, f_{m_u}, f_{m_b})$ that of CC with the corresponding list of frequencies. For each word, hashtag, mention, both unigram and bigram of so, the intersection with the terms in the centroid signature and their union are computed. Thus let $I^i = sgn(i) \cap sgn_C.sgn(i)$ be the intersection of a couple of features, and $U^i = sgn(i) \cup sgn_C.sgn(i)$ their union. Then

$$sim(so, CC) = \frac{\sum_{j=1}^{|I^i|} sgn_C.ff_I(j)}{\sum_{j=1}^{|U^i|} sgn_C.ff_U(j)} \times f(t_c) \qquad (2)$$

where $sgn_C.ff_I(j)$ and $sgn_C.ff_U(j)$ are the sum of frequencies of the terms appearing in the intersection and union, respectively, while $f(t_c)$ is the fading function (equation (1)) introduced in the precious section. $f(t_c)$ biases the similarity function towards clusters temporally closer to the tweet. The presence of this term allows to take into account not only the similarity between the signatures, but also their time distance, because, as observed above, the same events are usually generated in close temporal horizon.

Example 2. *Consider the social object of Example 1, and suppose to have a cluster with the following word unigrams of the centroid signature* $w_u^C = \{$*peter, pan, young, viral, pirates, love, Christopher, Walken, killed*$\}$*, and corresponding frequencies* $ff_{w_u} = \{$*10,10,3,3,3,5,8,8,1*$\}$*, then* $I^1 = I^{w_u} = \{$*Christopher, Walken, love*$\}$*,* $U^1 = U^{w_u} = \{$*peter, pan, young,viral,pirates,love, Christopher, Walken, killed, making, nervous, Captain, Hook*$\}$*. Then* $sgn_C.ff_I(1) = 8+8+5$*, while* $sgn_C.ff_U(1) = \{10+10+3+3+3+5+8+8+1+1+1+1+1\}$*. The other terms of equation (2) are computed in the same way.*

In the next section the results of the method are described.

V. Experimental Study

In this section, we report our experimental outcomes to demonstrate the effectiveness of the approach for detecting real-life interesting topics over tweet streams. The algorithm has been written in R [12].

A. Twitter Data Set

To collect tweets, we implemented a multi-threaded crawler to access the Twitter Streaming API. The data extracted in this work is a dataset of tweets tagged with GPS coordinates within the boundaries of the area of Manhattan in New York City. Specifically, we consider a Twitter dataset of 671,170 tweets issued by 91,356 mobile users, during the month of December 2014. Figure 3 shows the average number of tweets extracted per day. This number varies from a minimum value of $11,944$ tweets on December 3 and a peak of $28,565$ tweets on December 13. This peak is particularly significant since it reflects the outcomes of the topic detection algorithm. In fact, as will be illustrated in the following, one of the most relevant topics identified is related to an event that took place in New York on December 13, that is the march Million March NYC in solidarity with the families of those killed by law enforcement officers.

Because of the constraint on the length, tweet content could be very noisy: terms can be combined with punctuation and can contain abbreviations, typos, or conventional word variations. To reduce the amount of noise before the detection task, the raw data is preprocessed by applying tokenization and stemming. We used the packages tm and stringr of R [12] (e.g., removePunctuation, removeStopWord) to extract cleaned terms from the original messages

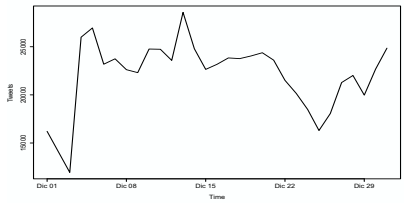


Figure 3. Tweet frequency.

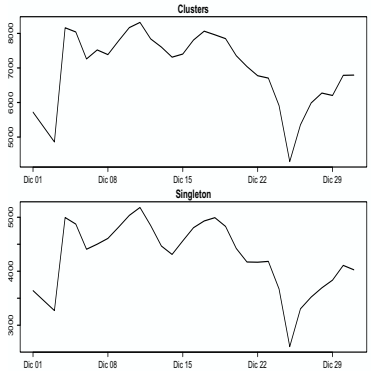


Figure 4. Daily numbers of clusters and singletons.

by removing stopwords and punctuation. We also used the package SnowballC of R for the stemming process to reduce inflected words to their root.

B. Results

We run the clustering algorithm over the collected tweets by simulating the online execution and the decay rate of historical information by fixing the activity period of a cluster to one day. Thus every 24 hours we discard the already generated clusters. As regards the threshold ϵ to compute the similarity between a tweet and a centroid, we tested different settings of this parameter on a subset of 5000 tweets to determine the configuration fitting better to the analyzed data. We considered the thresholds 0.001, 0.05, 0.1, and 0.5. The first two values returned a too low number of clusters, while $\epsilon = 0.5$ gave too many clusters, many of which containing only one tweet. Setting ϵ to 0.1 resulted to be a more appropriate value. Figure 4 shows the average number of clusters obtained per day, along with the number of singletons. The number of clusters varies from 5000 to 8000, but many are singletons, i.e. uninteresting tweets, and some of them of large size, i.e. corresponding to specific topics of interest in that day. Moreover, we computed the inter-cluster similarity to better understand the suitability of this value fixed for ϵ. The average similarity among pairs of clusters using Equation (2) is about 0.0014, thus clusters are well separated.

In the following we report the most significant results we obtained, where the significance is measured in terms

of topic popularity. We assumed that topic popularity can be expressed in terms of cluster dimension, that is the number of tweets talking about that topic. In Table I the top 8 identified topics are listed. Each detected topic is reported beside the corresponding story obtained from different newswire sources (e.g., Wikipedia, The New York Times, Daily News) and some representative tweets of the many retrieved by our algorithm for the corresponding topic. Topics vary greatly, ranging from arts and music to sports, from crimes and racism to cinema and finally, due to the period, converging toward Christmas-related argumentation. As can be noted, the detected topics are well aligned with the textual description of the real-word story.

Among the detected topics, particularly relevant are the ones about the Death of Erik Garner. Specifically, we have three topics related to this real-life story: (1) one topic, on the December 3, is associated with the grand jury decision to not indict NYPD officers involved in the chokehold case; (2) a second topic concerns the consequent protests that erupted in the city the day after, on December 4; (3) the third topic is related to the march Million March NYC of December 13 organized in solidarity with the families of those killed by law enforcement officers, streamed through the city streets in Washington, New York, Boston, Chicago and Oakland. The differences among the three related topics are also reflected by the slight shift in the semantics of the words in the corresponding tweets extracted by the algorithm. This can be noted in the clouds of the hashtag and word unigrams associated with the topics and shown in Figures 6. For example, one could see that while on December 3 the hashtag #erikgarner exhibits the highest frequency, on December 4 this hashtag is still the most frequent but with lower prevalence, and new hashtags appear, such as #blacklivesmatter and #ferguson, while others, for example #grandjurydecision, disappear. This semantics shifts is more evident in the word unigram set where there is a large predominance of the term protest and its variations. The top 8 topics include two on December 1: one, Banksy Documentary, refers to the art documentary about the street artist exhibition of Bansksy; the other, Knicks Match, to the NBA basketball match among New York Knicks and Miami Heat in the Madison Square Garden. On December 5, we detected other two topics: one concerns the musical Peter Pan Live!, a television special that was broadcast by NBC on December 4-8, 2014; the other topic is the Foo Fighter Concert at Irving Plaza. Finally, we noted that as Christmas was approaching the predominant topics were indeed Christmas-related. Accordingly, we obtained a number of clusters about Christmas. Table I reports only the cluster of December 21, as it was the larger among the Christmas-related ones.

Figures 5 to 7 show some representative word clouds of the hashtags and of the words of some of the topics listed in Table I. Such word clouds will be better explained after having introduced the *entropy* metrics, the measure we adopted to evaluate the quality of the produced clusters.

C. Evaluation

An important step to evaluate a clustering algorithm is defining a way to measure the quality of data partition. For clustering of numerical data, usually, measures based on geometrical distance are adopted. However, if, as in our case, the data is categorical, geometric approaches are inappropriate as there is no inherent distance measures between data values. Furthermore, since the proposed approach is a one pass clustering algorithm, after having processed a stream we loose it, therefore the only data content we maintain about a tweet is that summarized in the signature of the cluster centroid. Accordingly, to evaluate the quality of the clustering we refer to the concept of entropy [13].

The entropy of a cluster gives insights about cluster homogeneity. In other words, it tells us how people discuss about a topic, describing the distribution of the terms across the tweets: whether people tend to talk about a topic using similar terms. Precisely, the entropy decreases when tweets share a similar vocabulary, whereas it increases when the vocabulary varies among tweets grouped in the cluster. We computed the entropy for each feature f in the signature sng_C of the centroid CC of each cluster C. Specifically, we computed the variation of the feature throughout the tweets in the cluster. The higher the entropy, the more different the feature from tweet to tweet within the cluster. For this purpose we use the Shannon Entropy:

$$H(f)_C = -\sum_{i=1}^{|f|} p_i log\ p_i, \quad p_i = \frac{f_{f_i}}{N} \tag{3}$$

where $H(f)_C$ is the Shannon's entropy of feature f for the cluster C, f_{f_i} is the size of the value i of feature f (in other words its frequency), $|f|$ is the number of distinct values, N is the total size of feature f (the sum of the frequencies of the $|f|$ different values of the feature), and p_i is the observed probability of the value i. For example, if we consider the hashtag unigram feature h_u of the centroid of the cluster associated with the topic Deadth of Eric Garner on December 3, we have $|f| = |h_u| = 10$ different values for h_u and for the value $i =$ ericgarner of h_u we have a frequency $f_{f_i} = 236$ while $N = 353$. Thus, $p_i = \frac{f_{f_i}}{N} = 0.67$, and the entropy $H(h_u)_C = 0.88$.

In Table II the entropy values for the top 8 detected topics are shown. For each topic we report the entropy values for each of the features and the total average value. As can be noted from the table, the entropy values are in general rather small for all the features of the topics, ranging, on average, from 1.43 to 1.90. This means that the terms used in the tweets exhibit a certain regularity, in other words people use a similar vocabulary. For example, for the Peter Pan Live! topic we have an entropy of 0.60 for the feature h_u,

Topic	Day	Real-word Stories	Sample Tweets
Banksy Documentary	1	31 Days of Mystery: The "Banksy Does New York" Documentary (from Huffington Post) Banksy Does NY: HBO's Newest Doc Takes Us Back To Banksy's Residency in NYC (from Arts & Culture, New York)	In honor of taking this picture before I knew what it was and watching the #banksy
Knicks Match	1	New York Knicks vs Miami Heat: Betting Odds, Tips (from tipsterlabs.com) New York Knicks - Madison Square Garden, New York City (from The Garden)	#Knicks vs #Heat @ #MadisonSquareGarden #MSG #NYC @ Madison Square Garden Knicks vs Heat #melo @ Madison Square Garden
Death of Eric Garner	3	Eric Garner's son wants cop indicted for father's death, assures noFerguson-like riots in Staten Island (from DailyNews) Wave of protests after Grand Jury doesnt indict officer in Eric Garner chokehold case (from the New York Times)	Praying for no riots tonight in #NYC after #EricGarner grand jury decision Nyc March heading up 5th ave #EricGarner #BlackLivesMatter
Death of Eric Garner	4	Protests erupt in New York City after a grand jury decides not to indict NYPD officers Daniel Pantaleo and Justin Damico in the death of Garner (from Wikipedia) Eric Garner grand jury decision sparks protests which shut down West Side Highway, Brooklyn Bridge and Lincoln Tunnel (from DailyNews) Eric Garner protests erupt in New York (from The Guardian)	Chelsea, #NYC. Cops in undercover cars. Protesters unite with #reasonablecause. #stopracism #stophate #enoughisenough #justice #EricGarner Several scattered groups of protesters around Manhattan. #EricGarner
Peter Pan Live	5	Peter Pan Live! is a television special that was broadcast by NBC on December 4, 2014, starring Allison Williams in the title role and Christopher Walken as Captain Hook (from Wikipedia) Peter Pan Live! with Allison Williams and Christopher Walken on NBC (from the Newy Yor Times)	Christopher Walken is making me SO NERVOUS as Captain Hook...and I love it. #PeterPanLive. Christopher Walken tapping is all I need to see in life. #PeterPanLive
Foo Fighters Concert	5	Foo Fighters End Cross-Country Sonic Journey With Marathon New York Show (from The RolllingStones) Foo Fighters Setlist at Irving Plaza, New York, NY, USA (from NewYork.com)	@foofighters Please let Irving Plaza fans who've been standing in the cold for hours keep their place in line! We just wanna ROCK I guess that was it for @foofighters - BS on not waiting in line before 3pm at @IrvingPlaza - tickets were sold out at 1pm #IrvingPlaza #Foo.
Million March NYC	13	Justice for All and Million March NYC police brutality protests ? how the day unfolded (from The Guardian) 25,000 March in New York to Protest Police Violence (from the New York Times) Two NYPD cops assaulted as Manhattan march over Eric Garner case turns ugly (from Daily News)	Amidst the #HandsUpDontShoot chant at #MillionsMarchNYC, hearing a few yell Fight back! All white, in case you were wondering Given that cops kill 500+ people a year, this is an indictment rate of 1%. #MillionsMarchNYC
Christmas Time	21	Christmas Time	Merry Christmas; Happy Holidays to all! #Rockefeller #christmas in #nyc @ Rockefeller Center http #Rockefeller #Christmas tree- pretty even in the dreary, drizzly weather! #NYC @ Rockefeller Center

Table I
LIST OF THE TOP 8 DETECTED TOPICS.

which is really a low entropy value: this means that people twitted about that topic using almost the unique hashtag peterpanlive (see the word cloud of Figure 5(a)) which actually has a frequency of 93% all over the hashtags used for this topic. As regards Eric Garner of December 3, the feature h_u has an entropy of 0.88, which again is a very low value with a prevalence of the hashtag ericgarner that has a frequency of 67%. The entropy of word unigram w_u is instead 2.60 in fact, as we can see from Figure 6(b), in the tweets 10 word unigrams are used, all related to the topic, but with an average frequency of 10%, meaning that people use this variety of words with equal distribution.

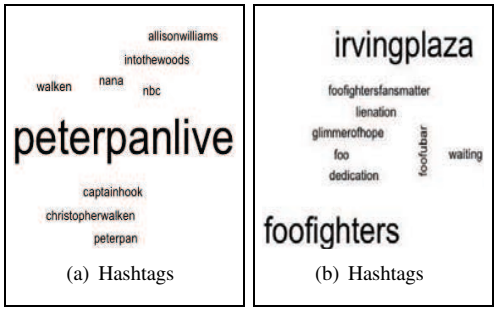
We can conclude that the achieved results are really good in terms of entropy, showing the homogeneity of the produced clusters.

Topic	h_u	h_b	m_u	m_b	w_u	w_b	avg
Banksy Documentary	1.52	2.75	1.34	3.11	1.63	2.73	2.18
Knicks Match	1.67	1.78	0.87	0.60	1.71	1.96	1.43
Death of Eric Garner	0.88	1.52	0.91	0.60	2.60	2.79	1.55
Death of Eric Garner	1.11	2.00	1.60	0.90	2.75	3.07	1.90
Peter Pan Live	0.60	1.82	1.40	0.78	2.77	3.08	1.74
Foo Fighters Concert	0.76	083	0.48	0.51	2.21	2.35	1.19
Million March NYC	0.92	1.74	1.03	0.60	2.41	2.57	1.55
Christmas Time	1.64	1.95	0.46	NA	1.82	2.23	1.62

Table II
ENTROPY VALUES OF THE TOP 8 DETECTED TOPICS.

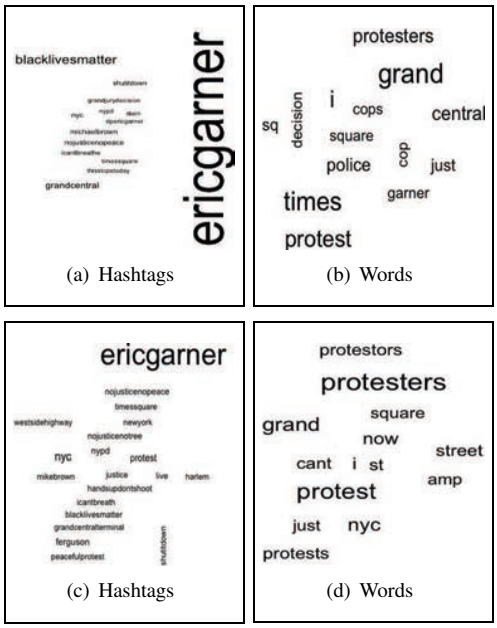
VI. CONCLUSIONS

The paper presented an online algorithm that incrementally groups tweets dealing with event-specific topics. The main novelties of the approach are the tweets and centroids representations that allow to summarize the contents of tweets examined so far, and a Jaccard based similarity measure between a tweet and a centroid which takes into account both term frequency and temporal closeness. The experimentation on text messages broadcast in the Manhattan area showed that the method is capable to detect topics related to events effectively occurring in the examined period. Several aspects, however, need to be investigated. First of all the setting of the similarity threshold, that sensibly influences the results. We are studying a mechanism that dynamically changes its value when many small or singleton clusters are generated. Another problem comes from the concept of



(a) Hashtags (b) Hashtags

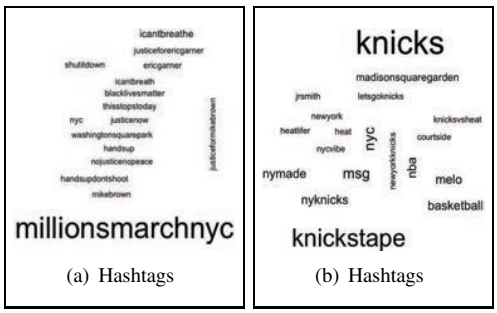
Figure 5. Wordcloud about Peter Pan Live (a) and Foo Fighters concert (b).



(a) Hashtags (b) Words

(c) Hashtags (d) Words

Figure 6. Wordcloud about Erik Garner events of December 3 (a) (b) and December 4 (c) (d).



(a) Hashtags (b) Hashtags

Figure 7. Wordcloud about Million March NYC (a) and Knicks Basketball Match (b).

life-span which, though application dependent, determines the importance of historical data. Moreover the approach has to be enriched with a technique for burst detection of unexpected events. However, the method we proposed can be considered as a valuable contribution to the emerging area of event detection from social media.

VII. Acknowledgement

This work has been partially supported by MIUR D.D. n. 0001542, under the project $BA2KNOW-PON03PE\_00001\_1$

References

[1] C. C. Aggarwal and P. S. Yu, "On clustering massive text and categorical data streams," *Knowl. Inf. Syst.*, vol. 24, no. 2, pp. 171–196, 2010.

[2] C. C. Aggarwal, "Mining text and social streams: A review," *SIGKDD Explor. Newsl.*, vol. 15, no. 2, pp. 9–19, Jun. 2014.

[3] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proc. of the 21st ACM SIGIR*, ser. SIGIR '98. ACM, 1998, pp. 28–36.

[4] Q. He, K. Chang, E. Lim, and J. Zhang, "Bursty feature representation for clustering text streams," in *Proceedings of the Seventh SIAM International Conference on Data Mining*, 2007, pp. 491–496.

[5] J. Lin, R. Snow, and W. Morgan, "Smoothing techniques for adaptive online language models: Topic tracking in tweet streams," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. ACM, 2011, pp. 422–429.

[6] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015.

[7] C. C. Aggarwal and K. Subbian, "Event detection in social streams," in *Proc. of the Twelfth SIAM Int. Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.*, 2012, pp. 624–635.

[8] J. Yin, A. Lampert, M. A. Cameron, B. Robinson, and R. Power, "Using social media to enhance emergency situation awareness," *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 52–59, 2012.

[9] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. ACM, 2010, pp. 851–860.

[10] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang, "STREAMCUBE: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream," in *31st IEEE International Conference on Data Engineering, ICDE 2015*, 2015, pp. 1561–1572.

[11] C. I. Muntean, G. A. Morar, and D. Moldovan, "Exploring the meaning behind twitter hashtags through clustering," in *Business Information Systems Workshops - BIS 2012 International Workshops and Future Internet Symposium*, 2012, pp. 231–242.

[12] I. Feinerer, K. Hornik, and D. Meyer, "Text mining infrastructure in r," *Journal of Statistical Software*, vol. 25, no. 5, pp. 1–54, 3 2008.

[13] T. Li, S. Ma, and M. Ogihara, "Entropy-based criterion in categorical clustering," in *ICML*, ser. ACM International Conference Proceeding Series, vol. 69. ACM, 2004.