Event Detection in Twitter Microblogging

Nikolaos D. Doulamis, *Member, IEEE*, Anastasios D. Doulamis, *Member, IEEE*, Panagiotis Kokkinos, and Emmanouel (Manos) Varvarigos

***Abstract*—The millions of tweets submitted daily overwhelm users who find it difficult to identify content of interest revealing the need for event detection algorithms in Twitter. Such algorithms are proposed in this paper covering both short (identifying what is currently happening) and long term periods (reviewing the most salient recently submitted events). For both scenarios, we propose fuzzy represented and timely evolved tweet-based theoretic information metrics to model Twitter dynamics. The Riemannian distance is also exploited with respect to words' signatures to minimize temporal effects due to submission delays. Events are detected through a multiassignment graph partitioning algorithm that: 1) optimally retains maximum coherence within a cluster and 2) while allowing a word to belong to several clusters (events). Experimental results on real-life data demonstrate that our approach outperforms other methods.**

***Index Terms*—Clustering, document and text processing, fuzzy representation, pattern analysis, tweet characterization.**

I. Introduction

EVENT detection algorithms that identify what is really being discussed in Twitter are necessary for structuring micro-blogging content [1]–[3]. The underlying idea of such algorithms is to extract a set of keywords that show an increasing usage at about the time an event is happening. Two main steps are required to construct an efficient tweet event detection algorithm: 1) we need to textually characterize the tweet content and 2) we need to apply learning strategies to retrieve events from tweets by analyzing the time evolution of the appearance count of certain words.

Current information theoretic metrics for document characterization, e.g., the term frequency–inverse document frequency (TF–IDF) [4] or distributional features [5], are not suitable for Twitter. This is because tweets: 1) are short messages (no more than 140 characters) leading to statistical inaccuracies when applying traditional document metrics on them; 2) present dynamic behavior with large volumes of posts being published during short time periods; and 3) exhibits temporal shifts at the times a particular event is posted. Twitter is also characterized by some key features: 1) users may subscribe to other users' tweets (followers) and 2) Twitter has the ability to forward a tweet to their followers (retweets).

All these reasons make processing of the tweets' content very different than that in traditional document analysis. Particularly, tweets present different word distributions from one time period to another, as new trends appear for the discussed topics. This implies time varying document frequency metrics. Additionally, tweets are generated from different authors having different target audiences and/or writing styles, and they contain a number of extra symbols, misspelled or abbreviated words, resulting in a noisy estimation of the term frequency metric. Finally, Twitter's following and retweeting are important and they should be taken into account in the analysis.

After text characterization, the second step is to extract events (equivalently sets of keywords) from the tweet posts by detecting common temporal similarities in their words' time series signals; the words of an event present a synchronized behavior in their appearance count. The research challenges at this stage are the following.

1) Tweet messages are often of unstructured meanings and the words of an event do not appear under a synchronized manner. This requires new forms of representation to compensate for the vagueness introduced by these temporal variations.
2) In contrast to conventional one-class assignment clustering methods, multiassignment clustering approaches are needed, since one word may belong to several events.
3) As words may belong to several events, clustering is not well-separable requiring the use of advanced methods, like graph partitioning.

A. Contribution

In this paper, we examine two scenarios. The short-term event detection (scenario 1) aims at detecting the most salient events that are currently being posted by the tweets. The long-term event detection (scenario 2) reviews events that have occurred over a long time period and aims at synopsizing what has mostly happened over that period.

For both scenarios, we propose new information metrics that exploit the dynamic nature of Twitter, and combine a number of techniques. First, we redefine the IDF score so as to make it a time varying metric that has the ability to sense "trending" topics. Second, we introduce conditional scores in order to address short message inaccuracies. Third, we apply processing over several time intervals so as to create "feature trajectories." Fourth, we propose a fuzzy representation

Manuscript received May 30, 2015; revised September 2, 2015; accepted September 27, 2015. Date of publication November 2, 2015; date of current version November 15, 2016. This work was supported by the European Union funded Project 4DCHWORLD through FP7 People Program under Grant 324523. This paper was recommended by Associate Editor J. Liu.

N. D. Doulamis and A. D. Doulamis are with the National Technical University of Athens, Athens 157 73, Greece (e-mail: ndoulam@cs.ntua.gr; adoulam@cs.ntua.gr).

P. Kokkinos and E. Varvarigos are with the Computer Engineering and Informatics Department, University of Patras, Patras 265 04, Greece (e-mail: kokkinop@ceid.upatras.gr; manos@ceid.upatras.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2489841

in order to compensate for temporal shifts in tweets' posting. Fifth, we model the clustering problem as a multi (instead of a single) assignment spectral graph partitioning problem.

This paper is organized as follows. Section II introduces metrics for tweet characterization. Section III discusses the multiassignment graph partitioning problem, Section IV defines the similarity measures. and Section V provides experimental results. Previous works are given in Section VI. Finally, Section VII concludes the paper.

II. Modeling Dynamic Nature of Tweets

A. Problem Formulation

We assume that the event detection algorithm is activated at the end of the kth time interval (period) $(t_{k-1}, t_k] = (t_k - \beta, t_k]$, $k = 1, 2, \ldots$, where $\beta > 0$ denotes the duration of the interval. We let $N(k)$ be the number of tweets that are posted during the kth time interval. In our setting, M events can be detected within the kth time interval. Each event is defined as a set of $W(m) = \{w_i^m, i = 1, 2, \ldots, L^m\}$ of L^m words, where $m = 1, \ldots, M$, stands for the event index. Two scenarios are supported.

Scenario 1 (Short-Term Event Detection): Scenario 1 extracts the most important events currently being posted in Twitter. In this scenario, we need to find out the synchronized words' behavior, i.e., which of the words posted by the tweets present similar temporal patterns.

Scenario 2 (Long-Term Event Detection): Scenario 2 reviews the events that have occurred over a long time interval to synopsize what has mostly happened during that interval. To detect the most important events in this scenario, we need to find out similar words' behavior being invariant to time shifts and for this reason new similarity metrics are needed.

To capture the temporal dynamicity of tweets' content, we form feature trajectories that model words' distribution over a time period. Let us denote by $\mathbf{s}_w(k)$ the feature trajectory of a word w. In scenario 1, the purpose is to cluster words together that present similar burst patterns in their distribution. Thus, we consider the similarity $D(\mathbf{s}_{wi}(k), \mathbf{s}_{wj}(k))$ of two words' feature trajectories. Instead, scenario 2 synopsizes what has mostly happened in long-time periods and therefore the words' distributions are clustered together independently of whether they are shifted in time, thus optimizing $D(\mathbf{s}_{wi}(k + t), \mathbf{s}_{wj}(k))$ for every time shift t of the feature time signal.

Table I presents indicative stories extracted from tweets posted on specific time instances. These tweets were retrieved by querying Twitter using the keywords of "BBC sports," "BBC auto," and "BBC news." The purpose of our algorithm for scenario 1 is to identify the most important events posted on Twitter at the specific time, by clustering the keywords these events are composed of. Table II presents indicative keywords related to events occurring over longer time periods (e.g., half a month). Again, these tweets have been retrieved by querying Twitter using the keyword of BBC news. Since scenario 2 synopsizes what has mostly happened during longer time periods, in Table II, we present only the relevant keywords of the main stories. From a technical point of view, the main difficulty in the scenario 2 case is that the relevant keywords are not absolutely synchronized and they are noisy. To cope with this, the Riemannian similarity metric is used to compensate micro-variations in words' signals.

TABLE I
Indicative Stories for Scenario 1 Case

Topic	Date	Main Story	Relative Keywords
Sports	4/06/14	Ecuador 2-2 England: Friendly Match; Wayne Rooney and Roy Hodgson react	England, Ecuador, Friendly
Sports	4/06/14	Diego Costa passes a medical at Chelsea	Medical, Diego, Costa, Chelsea, passes
Sports	4/06/14	Tennis: Andy Murray beat Gael Monfils and will face Rafael Nadal in Roland Garros semi-finals.	Andy, Murray, Monfils, Gael, Defeat, tennis
Sports	4/06/14	Netherlands 1-0 Wales, Friendly Match	Netherlands, Wales, Friendly
Auto	4/06/14	Masterpiece from McLaren 650S	McLaren 650S
Auto	4/06/14	Jyrobike: First auto-balancing bicycle unveiled	Jyrobike, auto-balancing bicycle
News	19/6/14	Harrison Ford broke left leg in accident	Harrison Ford leg broke
News	19/6/14	Iraq crisis: US to send 'military advisers	Iraq, crisis, US military, advisers

TABLE II
Indicative Stories for Scenario 2 Case

Topic	Date	Keywords from the Main Story
News	1-15/7/14	Israel Gaza Conflict: Crisis, death, war, kill, people
News	1-15/7/14	Ebola battle at west Africa.
News	1-15/7/14	World Cup: Germany wins, Argentina, final

B. Tweet-Based Information Theoretic Metrics

The TF–IDF [4] metric is not suitable to characterize tweets content. First, tweets are very short messages which lead to statistical inaccuracies in calculating TF scores. One way to compensate this difficulty is to consider as a document a collection of tweets gathered over a time interval k, instead of a single tweet. However, since tweets are posted by different authors of different target objectives, or writing styles, conditional probability distributions need to be incorporated. Additionally, we propose metrics that also consider Twitter-specific features such as retweeting and number of followers. Second, Twitter is very dynamic. This implies that words that are not statistically frequent in previous time intervals may be common in the current ones, since users' trends evolve through time. This transforms the IDF score to a time varying variable measuring the inverse trend frequency (ITF) of a word proportionally to IDF that measures the IDF. Third, there are usually temporal delays when posting tweets. Thus, event detection is very sensitive to those time shifts (temporal variations). Consequently, it is not proper to estimate the tweets metrics independently from one time interval to another, but instead the score of one time interval should affect the scores of adjacent intervals. To address this, fuzzy theoretic information metrics are adopted.

1) Conditional Word Tweet Frequency–Inverse Trend Word Tweet Frequency: We denote by $N^{(w)}(k)$ the number of

tweets containing the word w over all the $N(k)$ tweets collected at time interval k. We define the conditional word tweet frequency (CWTF) at time instance k and for a given word w as

$$\text{CWTF}(k, w) = N^{(w)}(k)/N(k). \tag{1}$$

The main difference of CWTF from the classical description of TF is that, here we count the number of tweets that contain a specific word within the current examined time interval k instead of counting the number of times that a word appears within a document. That is, all tweets that contain the specific word contribute the same to the calculation of CWTF. Thus, CWTF models a conditional distribution of tweets frequency, i.e., tweets under the condition that they contain the word w.

We define the inverse trend word tweet frequency (ITWTF) as a metric that assesses how frequently tweet posts contain the specific word w over p previous time intervals, $(t_{k-1} - \beta, t_{k-1}], \ldots, (t_{k-p} - \beta, t_{k-p}]$. In particular, we have that

$$\text{ITWTF}(k, p, w) = \log \frac{\sum_{i=1}^{p} N(k-i)}{\sum_{i=1}^{p} N^{(w)}(k-i)}. \tag{2}$$

In contrast to the conventional IDF score, ITWTF is a time varying metric that evolves as new time intervals are taken into account. A word that is rarely frequent up to the current examined time interval k will receive high values of ITWTF. However, if this word becomes trendy at the current time interval k, the CWTF score will take high values, forcing the product CWTF*ITWTF to be high. As long as this word remains trendy, in the forthcoming time intervals the ITWTF score will start to decay forcing the product CWTF*ITWTF to start decreasing as well. This means that, events that have been extracted as salient at previous stages will start to have less impact in the forthcoming stages. The product CWTF*ITWTF is the first tweet-based information theoretic metric $\vartheta_1(k, w)$

$$\vartheta_1(k, w) = \frac{N^{(w)}(k)}{N(k)} \cdot \log \frac{\sum_{i=1}^{p} N(k-i)}{\sum_{i=1}^{p} N^{(w)}(k-i)}. \tag{3}$$

2) Word Frequency–Inverse Trend Word Frequency: The second metric $\vartheta_2(k, w)$ considers the frequency of appearance of w in the tweets within the kth interval, denoted by $C^{(w)}(k)$. We also denote by $C(k)$ the total number of words that appear within the $N(k)$ tweets. Then, metric $\vartheta_2(k, w)$ is defined as

$$\vartheta_2(k, w) = \frac{C^{(w)}(k)}{C(k)} \cdot \log \frac{\sum_{i=1}^{p} C(k-i)}{\sum_{i=1}^{p} C^{(w)}(k-i)}. \tag{4}$$

The first term of (4) is designed to measure word frequency (WF) appearance at the current kth time interval, while the second term expresses the ITWF score, making $\vartheta_2(k, w)$ also a time varying signal. The main difference between the metrics $\vartheta_1(k, w)$ and $\vartheta_2(k, w)$ is that in $\vartheta_1(k, w)$ the significance of a word over the corpus of tweets at time interval k is independent of the number of words a tweet has, with tweets of few or many words contributing equally to the metric. The opposite holds for metric $\vartheta_2(k, w)$ of (4).

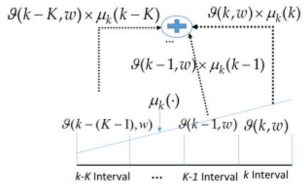

Fig. 1. Operation of the proposed fuzzy representation.

3) Weighted Conditional Word Tweet Frequency–Inverse Trend Weighted Conditional Word Tweet Frequency: The third metric, $\vartheta_3(k, w)$, considers Twitter specific parameters, such as the number of followers and retweets. The number of followers indicates authors' credibility. The number of retweets is a metric for ranking the importance of the textual content. In particular, we denote by $f_m(k)$, $m = 1, \ldots, N(k)$, the number of followers for the mth tweet at time k, and as $r_m(k)$ the number of retweets. Then, $p_m^f(k) = f_m(k)/\sum_{m=1}^{N(k)} f_m(k)$ and $p_m^r(k) = r_m(k)/\sum_{m=1}^{N(k)} r_m(k)$ are their normalized values. Then

$$\begin{aligned}\vartheta_3(k, w) &= \frac{\sum_{m=1}^{N(k)} p_m^f(k) \cdot p_m^r(k) \cdot i_m(w, k)}{\sum_{m=1}^{N(k)} p_m^f(k) \cdot p_m^r(k)} \\ &\quad \times \log \frac{\sum_{j=1}^{p} \sum_{m=1}^{N(k-j)} p_m^f(k-j) \cdot p_m^r(k-j)}{\sum_{j=1}^{p} \sum_{m=1}^{N(k-j)} p_m^f(k-j) \cdot p_m^r(k-j) \cdot i_m(w, k-j)}\end{aligned} \tag{5}$$

where $i_m(w, k)$ is an indicator function that equals one if the mth tweet contains the word w, and zero otherwise.

C. Fuzzy Tweet-Based Representation

We form a time series signal, denoted as $\mathbf{x}_w(k)$, that contains the tweet-based information theoretic metrics of (3)–(5) over a time period of time intervals

$$\mathbf{x}_w(k) = [\vartheta(k, w)\vartheta(k-1, w) \cdots]^T. \tag{6}$$

In (6), variable $\vartheta(k, w)$ refers to one of the three metrics defined in (3)–(5). Each element $\vartheta(k, w)$ of the time series signal $\mathbf{x}_w(k)$ expresses the degree of importance of word w at the kth time interval and in a nonfuzzy representation is calculated independently of each other. However, in our proposed fuzzy representation, metric $\vartheta(k, w)$ for the kth interval is diffused over K previous intervals but with a different degree of membership for each interval

$$\vartheta_f(k, w) = \sum_{i=0}^{K-1} \vartheta(k-i, w)^* \mu_k(k-i) \tag{7}$$

where subscript f denotes the fuzzy representation of the respective metric and $\mu_k(k-i)$ is the fuzzy membership degree for the $(k-i)$th time interval. The μ_k takes values in the range [0, 1]. Usually triangular functions are used to obtain values of μ_k but any other fuzzy function can be also adopted. Values of μ_k near unity (zero) indicate high (low) degree of membership of the metric. Other types of diffusion methods can also

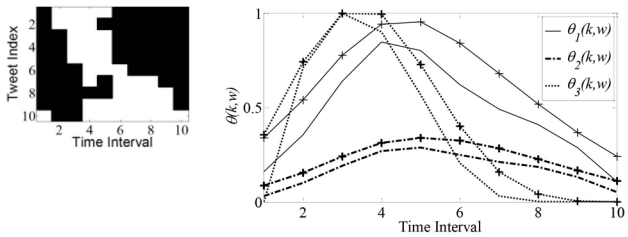

Fig. 2. (a) Word distribution over ten time intervals. (b) Time series of the three proposed metrics. The curves with "+" corresponds to fuzzy representations. The values have been normalized in [0, 1].

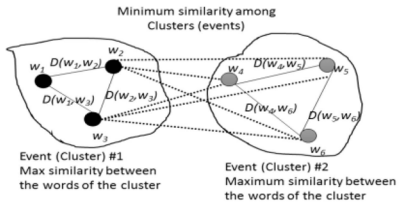

Fig. 3. Schematic depicting the modeling of the event tweet detection problem through graph partitioning.

be adopted, such as trapezoid, Gaussian, generalized bell, sigmoid, etc. [6]. Fig. 1 illustrates a graphical representation of the proposed fuzzy model. Each value $\vartheta(k, w)$ is weighed by membership $\mu_k()$ and the respective product is diffused over K time intervals. Then, the fuzzy time series signal is

$$\mathbf{x}_{f,w}(k) = \left[\vartheta_f(k, w)\vartheta_f(k-1, w)\cdots\right]^T. \tag{8}$$

D. Scale-Time Modeling

To further compensate for the sensitivity of the tweets to time shifts (temporal variations), we apply the discrete wavelet transform (DWT) on the signal of (8). The reason we chose the DWT instead of discrete Fourier transformation (DFT) is that, unlike the sine/cosine functions used in DFT, which are localized in frequency but extend infinitely in time, wavelets are localized both in time and in frequency domain. In particular, let us denote by $\mathbf{s}_w(k)$ the wavelet transformed signal of $\mathbf{x}_{f,w}(k)$, $\mathbf{s}_w(k) \equiv \psi(\mathbf{x}_{f,w}(k))$, where $\psi(\cdot)$ refers to the discrete wavelet operator. Then

$$\mathbf{s}_w(k) = \left[\cdots c_w^{(i,j)}(k)\cdots\right]^T \equiv \left[\cdots s_w^{(i)}(k)\cdots\right]^T \tag{9}$$

where we denote by $c_w^{(i,j)}(k)$ the jth coefficient of signal $\mathbf{s}_w(k)$ at the ith scale and by $s_w^{(i)}(k)$ the ith element of the transformed wavelet version of signal $\mathbf{s}_w(k)$ derived by simply renumbering the wavelet coefficients $c_w^{(i,j)}(k)$. In our approach, we keep the q wavelet coefficients, preferring the ones derived from the low pass scale.

E. Discussion on Example

To better understand the metrics defined above, we consider a simple synthetic example. Assume we are given $N(k) = 10$ tweets over ten time intervals, $k = 1, \ldots, 10$. We also assume eight words per tweet. Then, we construct three time series $\mathbf{x}_w(k)$, each for the three metrics $\vartheta_{\{1,2,3\}}(k, w)$. The tweets are assumed to be organized into three groups. The first three tweets (1–3) are assumed to have the majority of followers (most significant tweets), the second group (4–6) is of medium importance, while the third group (7–10) is of the lowest importance. We assume 1000 followers for the first group, 100 followers for the second group, and ten followers for the third group.

We visualize the appearances of a specific word w over the ten time intervals as in Fig. 2(a). A white block indicates that word w appears in the respective tweet. At the first time interval, word w is posted only by one unimportant user. At the second time interval, two out of the three most important users tweet word w. At following time intervals, all the important users followed by the users of intermediate importance submit word w. Finally, at the last time intervals, the unimportant users submit the word in their tweets. The meaning of the example of Fig. 2(a) is that word w initially appears in one tweet as a noise. Then at the second time interval word w becomes trendy and therefore starts to be posted by the most important users, that is, users with many followers. At subsequent time intervals (from the 6th and on), the word w is not posted by the important users but instead it is propagated through the tweets of users of medium and low importance.

Fig. 2(b) depicts the time series of the three metrics for this specific word over the ten different time intervals. We have assumed an initial probability of appearance of 10^{-6} for word w, i.e., w rarely appears in the previous time intervals. This initial probability is used for the calculation of ITF score for the first time interval. For the remaining nine intervals, ITF term is updated according to the conditional frequency score computed at a previous time interval [see the first terms of (3)–(5)]. The values of $\vartheta(k, w)$ have been normalized to fit the range of [0, 1] for better comparisons. As is observed from Fig. 2(b), the third metric $\vartheta_3(k, w)$ (5) presents the highest discrimination regarding modeling of w. In particular, $\vartheta_3(k, w)$ takes high values only at the time intervals where the important users post the word. In contrast, metrics $\vartheta_1(k, w)$ and $\vartheta_2(k, w)$ spread the word's significance across all intervals since they handle all tweets equally. Metric $\vartheta_2(k, w)$ yields the lowest discrimination accuracy. This is mainly because in $\vartheta_{\{1,3\}}(k, w)$, a tweet fully contributes to the overall measure if it contains the specific word (0 or 1 contribution), while in $\vartheta_2(k, w)$ a tweet contributes proportionally to the number of words within this tweet. In Fig. 2(b), we have also depicted with the "+" mark the fuzzy representations of the respective metrics. We observe that the fuzzy mapping improves the discriminatory performance.

III. Event Detection as Multiassignment Graph Partitioning Problem

Following the analysis of Section II, each word is represented as a time series signal $\mathbf{s}_w(k)$. Then, our goal is to create events consisting of a set of L words by examining the similarity between the word's distributions. The value of L is proportional to the number of the most frequent words (see Section V-A3). Toward this end, a graph $G = \{V, E\}$

is created, whose vertex set $V = \{w_1, w_2, \ldots, w_L\}$ corresponds to the set of L different words we examine (this is the reason we use symbol w to notate the elements of V). Each edge $e_{i,j} = (w_i, w_j)$ in E carries a non-negative weight equal to a distance $D(w_i, w_j)$ defined between the corresponding words w_i and w_j. The distance metrics adopted will be discussed in Section IV. Fig. 3 presents the concept of modeling the event tweet detection problem as a graph partitioning problem. In this figure, we have assumed six words and two clusters (events). The vertices of the graph are the words, while the edges are weighed by the respective similarity metric (see Section IV). Our goal is to decompose the graph into M partitions (sub-graphs), each of which corresponds to an event.

In conventional graph partitioning problems, a graph is divided into M mutual exclusive sub-graphs, implying that each datum belongs only to one cluster. Such a partitioning, however, is not valid in our event detection setting where it is possible for one word to belong to more than one events, but with different degrees of membership. For this reason, in this paper, we adopt a multiassignment partitioning scheme, as explained in the following section.

A. Multiassignment Graph Partitioning

Let us define an Lx1 membership vector, $\mathbf{u}_r = [\cdots u_{i,r} \cdots]^T$, each element $u_{i,r}$ $i = 1, 2, \ldots, L$, of which indicates the membership degree of the ith word to the rth partition. Please note that $\mathbf{u}_r$ is different than μ_k used in (7). Variable μ_k refers to the fuzzy membership degree of the information-theoretic metrics to respective time intervals using a fuzzy function, like the triangular ones. In other words, μ_k compensates temporal shifts in postings words. Vector $\mathbf{u}_r$, on the other hand, expresses the membership degree of words to an event (cluster).

The membership values $u_{i,r}$ are allowed to take continuous values expressing the level that a vertex i (that is, word w_i) belongs to the rth partition. Elements $u_{i,r}$ do not express probabilities, but the matching degree of a word to a particular event set. This means that one word can belong to more than one events (clusters) with membership degree of one. Let us now denote by $\mathbf{D} = [D(w_i, w_j)]$ a matrix containing the similarity measures for all $L \times L$ pairs of words. Let us also denote as $\mathbf{\Lambda} = \text{diag}(\cdots l_i \cdots)$ a diagonal matrix, whose elements l_i, $i = 1, 2, \ldots, L$, express the cumulative similarity degree of one word w_i with the remaining words, that is

$$l_i = \sum_j D(w_i, w_j). \tag{10}$$

Actually, $\mathbf{\Lambda}$ expresses the degree matrix of the graph. The goal of a multiassignment graph partitioning algorithm is to minimize the normalized similarity degree of the rth partition with respect to the others. Normalization factors have been added to avoid the creation of small partitions, deteriorating clustering performance [7]. Therefore, we have that

$$\hat{\mathbf{u}}_r, \forall r: \ \min P = \min \sum_{r=1}^{M} \frac{\mathbf{u}_r^T \cdot (\mathbf{\Lambda} - \mathbf{D}) \cdot \mathbf{u}_r}{\mathbf{u}_r^T \cdot \mathbf{\Lambda} \cdot \mathbf{u}_r}. \tag{11}$$

Minimization of (11) is equivalent to estimate the optimal membership vectors $\hat{\mathbf{u}}_r$ $r = 1, 2, \ldots, M$, where we recall that M stands for the number of clusters we want to create.

B. Membership Vectors Estimation

Let us denote in the following as $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_M]$ the membership matrix, the columns of which are the membership vectors $\mathbf{u}_r$, $r = 1, \ldots, M$. Then, (11) can be written as

$$P = M - \text{trace}\Big(\big(\mathbf{U}^T \cdot \mathbf{D} \cdot \mathbf{U}\big) \cdot \big(\mathbf{U}^T \cdot \mathbf{\Lambda} \cdot \mathbf{U}\big)^{-1}\Big). \tag{12}$$

Using matrix calculations the trace of (12) is expressed

$$P = M - \text{trace}\Big(\tilde{\mathbf{U}}^T \cdot \mathbf{\Lambda}^{-1/2} \cdot \mathbf{D} \cdot \mathbf{\Lambda}^{-1/2} \cdot \tilde{\mathbf{U}}\Big) \tag{13}$$

where matrix $\tilde{\mathbf{U}}$ is in fact a "orthonormal" version of the membership matrix $\mathbf{U}$, that is $\tilde{\mathbf{U}} = \mathbf{U} \cdot (\mathbf{U}^T \cdot \mathbf{U})^{-1/2}$.

To minimize (13), we can use the Ky-Fan theorem [8], which states that optimization of (12) subject to $\tilde{\mathbf{U}}^T \cdot \tilde{\mathbf{U}} = I$ is obtained as the sum of the $M(M < L)$ largest eigenvalues of matrix $\mathbf{\Lambda}^{-1/2} \cdot \mathbf{D} \cdot \mathbf{\Lambda}^{-1/2}$, where λ_i refers to the ith largest eigenvalue of matrix $\mathbf{\Lambda}^{-1/2} \cdot \mathbf{D} \cdot \mathbf{\Lambda}^{-1/2}$, that is

$$\max_{\text{subject to } \tilde{\mathbf{U}}^T \cdot \tilde{\mathbf{U}} = \mathbf{I}} \left\{ \text{trace}\Big(\tilde{\mathbf{U}}^T \cdot \mathbf{\Lambda}^{1/2} \cdot \mathbf{D} \cdot \mathbf{\Lambda}^{-1/2} \cdot \tilde{\mathbf{U}}\Big) \right\} = \sum_{i=1}^{M} \lambda_i. \tag{14}$$

However, the maximization of (14) leads to the minimization of (13), and the minimum value of P is given as $M - \sum_{i=1}^{M} \lambda_i$. This min value $\tilde{\mathbf{U}}_{\text{opt}}$ is obtained for matrix $\tilde{\mathbf{U}}$ at

$$\tilde{\mathbf{U}}_{\text{opt}} = \mathbf{U}^e \cdot \mathbf{R} \tag{15}$$

where $\mathbf{U}^e$ is a $L \times M$ matrix whose columns are the eigenvectors corresponding to the M largest eigenvalues of matrix $\mathbf{\Lambda}^{-1/2} \cdot \mathbf{D} \cdot \mathbf{\Lambda}^{-1/2}$ and $\mathbf{R}$ is an arbitrary rotation matrix and thus $\mathbf{R}^T = \mathbf{R}^{-1}$ and $\det(\mathbf{R}) = 1$. This optimal value satisfies the constraint of $\tilde{\mathbf{U}}$, that is, $\tilde{\mathbf{U}}^T \cdot \tilde{\mathbf{U}} = \mathbf{I}$ [9].

Equation (15) gives the optimal solution for the membership matrix but in the continuous domain, i.e., the elements of $\tilde{\mathbf{U}}_{\text{opt}}$ can take any arbitrary continuous value. However, as we have stated above, the membership vectors $\mathbf{u}_r$ in matrix $\mathbf{U}$ express the degree of membership of a vertex (i.e., word) of G to each of the M available clusters (events). In a real-life tweet driven event detection problem, one word belongs to a limited number of events. To address this, we need to approximate the solution of (15) under a multiassignment clustering framework.

1) Optimal Rotation Matrix Estimation: More specifically, let us define by $\mathbf{U}^a = [\mathbf{u}_1^a \mathbf{u}_2^a \cdots \mathbf{u}_M^a]$ an $L \times M$ approximate matrix each row (i.e., a word) of which contains values equal to one for the columns (representing events) at which the respective word is assigned to, and values equal to zero, otherwise

$$u_{i,j}^a = \begin{cases} 1 & \text{when } w_j \in i \text{ partition} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where $u_{i,j}^a$ is the jth element of vector $\mathbf{u}_i^a$ with $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, L$. The $L(M)$ is the total number of words (events). Actually, matrix $\mathbf{U}^a$ is a discrete approximate

solution. In contrast to conventional one-cluster assignment problems, a row of $\mathbf{U}^a$ may have more than one unit elements.

In this paper, we propose an iterative methodology that recursively estimates an optimal rotation matrix $\hat{\mathbf{R}}$ in a way to update the continuous solution $\mathbf{U}^e \cdot \mathbf{R}$ to be closest to the discrete one. In particular, matrix $\hat{\mathbf{R}}$ is estimated through the following constrained minimization problem [10]:

$$\min \hat{\mathbf{R}} = \arg\min \left\| \mathbf{U}^a - \mathbf{U}^e \cdot \mathbf{R} \right\| \quad \text{subject to } \mathbf{R}^T = \mathbf{R}^{-1}. \quad (17)$$

The solution of (17) can be expressed as an singular value decomposition problem

$$\hat{\mathbf{R}} = \mathbf{\Xi} \cdot \mathbf{\Pi}^T \text{ and } \mathbf{U}^{a^T} \cdot \mathbf{U}^e = \mathbf{\Pi} \cdot \mathbf{\Omega} \cdot \mathbf{\Xi}^T \quad (18)$$

where $(\mathbf{\Pi}, \mathbf{\Omega}, \mathbf{\Xi})$ is the singular value decomposition of $\mathbf{U}^{a^T} \cdot \mathbf{U}^e$ with $\mathbf{\Pi}^T \cdot \mathbf{\Pi} = \mathbf{I}$ and $\mathbf{\Xi}^T \cdot \mathbf{\Xi} = \mathbf{I}$.

C. Multiassignment Clustering Approximation

A simple rounding mechanism is to set the maximum value of each row of the continuous solution equal to 1 and the other to 0. This approach has the drawback that it is valid only for a single-assignment method (which is not our case), and its performance is not satisfactory when there is no clear dominant maximum value. In this paper, we adopt as rounding process a novel method, appropriate for the multiassignment clustering problem. Initially, the algorithm starts assuming a hard one-class assignment, and then the problem is relaxed. In particular, we treat the L rows of the continuous solution as an M-dimensional feature vector and then we apply the k-means algorithm to assign the L rows (words) to one of the M available clusters (events). The main concept is to assign each word to only one cluster that fits best. Using this step we create M mutual exclusive sets (that correspond to the M events) $\tilde{W}(m)$, $m = 1, \ldots, M$. Let us denote as $u_m^{\min}$ the minimum membership value over all words $w_i \in \tilde{W}(m)$ for the event $\tilde{W}(m)$

$$u_m^{\min} = \min u_{i,m} \quad \forall w_i \in W(m) \quad (19)$$

where $u_{i,m}$ is the membership value of the ith word to the mth event. Using (19), we are able to extend sets $\tilde{W}(m)$ to fit the multiassignment clustering problem as

$$W(m) = \tilde{W}(m) \cup \left\{ w_i : u_{i,m} \geq u_m^{\min} \right\}. \quad (20)$$

Equation (20) means that we append to the initially estimated sets $\tilde{W}(m)$ additional words whose membership values to this particular event is greater than or equal to $u_m^{\min}$, even though these words have been assigned to other events.

D. Dynamically Updating Rotation Matrix

The aforementioned process describes how we can obtain from the continuous an approximate discrete solution. In this section, we describe an iterative methodology for updating the rotation matrix $\mathbf{R}$ of (17) to yield a new solution that can provide a better discrete approximation.

We denote by $\hat{\mathbf{R}}(n)$ the optimal rotation matrix at the nth iteration of the algorithm. Using matrix $\hat{\mathbf{R}}(n)$, we estimate a new continuous solution $\tilde{\mathbf{U}}_{\text{opt}}(n) = \mathbf{U}^e \cdot \hat{\mathbf{R}}(n)$ that can provide a better discrete approximation solution. We then discretize $\tilde{\mathbf{U}}_{\text{opt}}(n)$ to get $\mathbf{U}^a(n)$ as above. Then, using $\mathbf{U}^a(n)$ and $\mathbf{U}^e$, we can derive the new optimal matrix $\hat{\mathbf{R}}(n+1)$ at the $(n+1)$th iteration by singularly decomposing the matrices $\mathbf{U}^{a^T}(n) \cdot \mathbf{U}^e$.

E. Estimation of Number of Clusters

As shown in [11], in an ideal case of clean datasets, the highest magnitude eigenvalue of $\mathbf{\Lambda}^{-1/2} \cdot \mathbf{D} \cdot \mathbf{\Lambda}^{-1/2}$ will be a repeated eigenvalue of magnitude equal to 1 and multiplicity equal to M. Thus, one could estimate M by counting the number of eigenvalues that are equal to 1. However, if the clusters are not clearly separated, the magnitudes of the highest eigenvalues start to deviate from 1, making such a criterion unreliable [12]. An alternative approach is to search for a drop in the magnitude of the ordered eigenvalues [13]. Since the eigenvalues depend on the structure of the clusters, no assumptions can be made on their values, implying that the eigen-gap can be either small or large [12]. To overcome this difficulty, [12] introduces a cost function J that relates to the degree of separability of the clusters, and the optimal number of clusters is derived as the one that optimizes this cost function. However, the analysis in [12] assumes a single-cluster assignment problem, where each sample is assigned to one cluster. Since, in our case, we have a multiassignment problem at hand, we need to modify the cost function of [12] to fit our constraints. In order to do so, recall that $\tilde{\mathbf{U}}_{\text{opt}}$ is an $L \times M$ matrix obtained after rotating the eigenvector matrix $\mathbf{U}^e$ (15). Then, if we denote by $U_{i,j}$ the (i,j) element of $\tilde{\mathbf{U}}_{\text{opt}}$ and let $U_{\max} = \max_j U_{i,j}$, the optimal number of clusters M_{opt} is given as the argument that minimizes the cost function J

$$M_{\text{opt}} = \arg\min_M J = \frac{1}{L \cdot \tilde{M}} \sum_{i=1}^{L} \sum_{j=1}^{M} \frac{U_{i,j}^2}{U_{\max}^2} \quad (21)$$

where $\tilde{M}$ is the average number of clusters to which a sample can belong to. Equation (21) means that samples that clearly belong to a cluster contribute with one to the cost function J, instead of the samples that are far away from a cluster.

Variable $\tilde{M}$ is estimated by sorting the values of $U_{i,j}^2/U_{\max}^2$ in a descending order and then computing the differences $d_j = U_{i,j}^2/U_{\max}^2 - U_{i,j+1}^2/U_{\max}^2$ between two adjacent values, for $j = 1, \ldots, M-1$. We also estimate the cumulative values of these differences. Variable $\tilde{M}$ is computed at the index where a drop of the cumulative values below a threshold (e.g., 0.4 in our case) is noticed. We calculate J for different number of clusters (starting from a minimum value of 2 up to a maximum one) and select as the best number of clusters the one that minimizes J.

IV. Similarity Measures

In this section, we define the similarity metrics among the words' time series signals. In particular, we use the cross-correlation and the Riemannian distance as the similarity metric for scenarios 1 and 2. An essential difference between the two is that cross-correlation expresses the similarity degree of two signals (takes value one when we have absolute matching), while Riemannian metric is a distance

(inverse function of the similarity) that takes value zero in the matching case. For this reason, in our case, we linearly normalize the Riemannian metric so that its maximum value is set to 0 (high dissimilarity) and its minimum is set 1 (high similarity).

A. Cross-Correlation as for Similarity for Scenario 1

The main limitation of the most commonly used Euclidean distance is that it is sensitive to scaling and/or translation [14], [15]. For this reason, the normalized cross-correlation D_{cc} is adopted in this paper for the scenario 1 case

$$D_{\text{cc}}(w_i, w_j) = \frac{\mathbf{s}_{w_i}^T \cdot \mathbf{s}_{w_j}}{\sqrt{\mathbf{s}_{w_i}^T \cdot \mathbf{s}_{w_i}}\sqrt{\mathbf{s}_{w_j}^T \cdot \mathbf{s}_{w_j}}}. \tag{22}$$

B. Riemannian Metric as for Similarity for Scenario 2

The normalized cross-correlation metric is still an "averaging operator." For instance, using the cross-correlation criterion, it is quite probable for words' distributions that exhibit similar behavior but are shifted in time to yield low cross-correlation values. To address this difficulty, we introduce the Riemannian geodesic metric [16].

In Section II-D we constructed a wavelet signal $\mathbf{s}_w^{(k)}$ of size q for a given time interval k and word w (q stands for the number of wavelet coefficients), and defined an $q \times q$ autocovariance matrix $\mathbf{C}_{\boldsymbol{w}}$ with elements

$$c_w(l) = E\left\{\left(\mathbf{s}_w^{(k)} - E\left\{\mathbf{s}_w^{(k)}\right\}\right)^T \cdot \left(\mathbf{s}_w^{(k-l)} - E\left\{\mathbf{s}_w^{(k-l)}\right\}\right)\right\} \tag{23}$$

where $E\{\cdot\}$ denotes the expectation operator. In (23), we have not added the conventional row/column indices on matrix elements $c_w(l)$ since these elements in fact depend on $k - l$ distance due to the autocovariance matrix properties.

The $q \times q$ symmetric positive definite matrices (nonsingular covariance matrices) can be formulated as connected Riemannian manifolds. It has been shown in geodesy science that the distance between the two arbitrary covariance matrices $\mathbf{A}$ and $\mathbf{B}$ is given by the following equation [16]:

$$D_R(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(\mathbf{A}, \mathbf{B})}. \tag{24}$$

In (24), $\lambda_i(\mathbf{A}, \mathbf{B})$ refers to the ith generalized eigenvalue of matrices $\mathbf{A}$ and $\mathbf{B}$, i.e., $\mathbf{A} \cdot \mathbf{x} = \lambda \cdot \mathbf{B} \cdot \mathbf{x}$, while n is the number of generalized eigenvalues. In our case, matrices $\mathbf{A}$ and $\mathbf{B}$ are the autocovariance matrices of two words' distributions, that is, $\mathbf{A} \equiv \mathbf{C}_{w_i}$ and $\mathbf{A} \equiv \mathbf{C}_{w_j}$ for two arbitrary words w_i, w_j.

C. Discussion on Example

1) Cross-Correlation and Wavelets: We generate eight different word distributions in a similar way to that used for generating the two signals used in Fig. 2 and we calculate the three metrics $\vartheta_{\{1,2,3\}}(k, w)$ for each of them. Then, we compute the cross-correlation distance among the time series signals of all the eight words, resulting in an 8×8 cross-correlation matrix for each of the three adopted metrics $\vartheta_{\{1,2,3\}}(k, w)$.

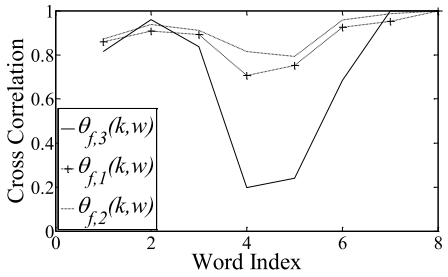

Fig. 4. Discriminatory performance of the cross-correlation distance of one word against other eight word indices under the fuzzy feature trajectories case; the similar (dissimilar) word pairs are closer to 1 (0).

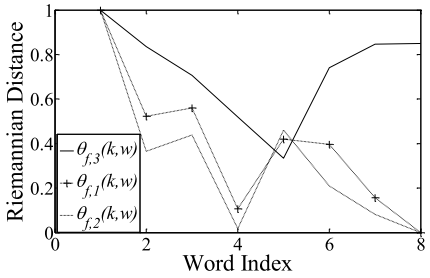

Fig. 5. Discriminatory performance of different tweet post characterization metrics as regards Riemannian similarity measure. The fuzzy three metrics have been used.

Fig. 4 presents the cross-correlation similarity distance of one word with respect to all the eight ones, examining the discriminatory performance of the three tweet-based metrics under the fuzzy representation (this is indicated by subscript f). A better discriminatory performance means that similar word pairs must have cross-correlation distance close to one, whereas dissimilar ones close to zero. As is observed, the time series signal [see (8)] derived from metric $\vartheta_{f,3}(k, w)$ exhibits higher discriminatory performance than the time series derived from metrics $\vartheta_{f,1}(k, w)$ and $\vartheta_{f,2}(k, w)$.

2) Riemannian Metric: Fig. 5 presents the behavior of the three metrics using fuzzy representation and Riemannian metric. Again, eight words have been generated and the first one (randomly selected) is compared against all the others. In the examined controlled environment, words 1–3, 7, and 8 have been generated so as to present similar behavior. It is clear that $\vartheta_{f,3}(k, w)$ metric gives high discriminatory performance compared to the other metrics. Particularly, it presents high values for the words 2, 3, 7, and 8 instead of the other two metrics that are not so robust especially regarding the words 7 and 8.

In the following we discuss the effect of the wavelet transform on the Riemannian metric, which is suitable for scenario 2. In this case, we use metric $\vartheta_{f,3}(k, w)$ since it yields better performance than $\vartheta_{f,1}(k, w)$ and $\vartheta_{f,2}(k, w)$ as found above. Fig. 6 depicts the discriminatory performance using both a wavelet and a non-wavelet representation and the Riemannian metric. As is observed, the wavelet representation provides better discrimination than the non-wavelet one, since again the wavelet transform smoothens the temporal noise (synchronicity in the posts). In the same figure, we have depicted the performance of the cross-correlation similarity metric defined as in (21). It is clear that cross-correlation

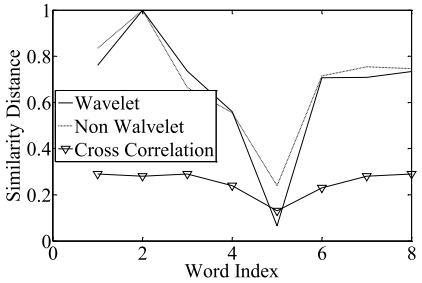

Fig. 6. Discriminatory performance of the fuzzy represented weighted conditional tweet WF metric on cross-correlation and Riemannian distance.

fails to provide sufficient discrimination for detecting the most important events as required by scenario 2.

V. Simulation Results

A. Experimental Set-Up

1) Dataset Creation: We extracted real-life tweet data using the publicly available application programming interface (API) of Twitter, which can be found at https://dev.twitter.com/docs/streaming-apis. Using this API, we downloaded tweets at different time intervals. For scenario 1, the time intervals were defined to have a 6-h duration and the selected time horizon was one month, for a total of 120 time intervals assessed. For scenario 2, the time intervals were defined every two days and the selected horizon was six months, for a total of 90 time intervals examined. As the extracted tweets were highly heterogeneous in their content, we also filtered them out by using topic categories when querying Twitter, such as "sports," "auto," "news," and major media agencies/broadcasters. For example, for the topic category sports, we constrained the query on the Twitter search engine API using the keywords BBC sports, Reuters sports, Euronews sport, and Fifacom. It should be mentioned that these keywords were not applied in the account field (the "from" query operator) of the Twitter search engine but were handled as text query operator. In this way, we collected tweets not only from the accounts of major media agencies but also from simple users who were referring to these agencies. Simple users' tweet posts were actually the majority of our data. The number of tweets extracted highly depends on the respective category and keywords used for querying. For example, within a 6-h time interval of scenario 1, the tweets retrieved from the keywords of BBC sports were about 250, while for the BBC news they were more than 3000. Proportional numbers are concluded for the scenario 2 case.

2) Ground-Truth Creation: Clearly, since the extracted tweets are highly heterogeneous, an overwhelming amount of effort is required to manually detect the topics and extract relevant keywords associated to each of them. To address this difficulty, a semi-automatic process was adopted in this paper, different for each of the two scenarios examined, to reduce the textual annotation time. Regarding scenario 1, for each six-hour time interval, we used the Natural Language Toolkit of Python to count the words' frequencies for all the retrieved tweets. We identified in this way the 20 words that appeared most frequently in the tweets and delivered them to the three experts who were responsible for the annotation. The experts were specialized in communication and media studies.

TABLE III
Experts Annotation, Ground Truth, and Produced Events for Some Story Examples of Scenario 1

Main Story: Ecuador 2-2 England: Friendly Match; Wayne Rooney and Roy Hodgson react England, Ecuador, Friendly on 4/6/2014 (see Table I)				
Expert 1# annotation:	**Expert 2# annotation:**	**Expert 3# annotation:**	**Union**	**Produced**
Ecuador, England, Friendly	Ecuador, England 2-2	Ecuador, Ecuador, friendly, Rooney	Ecuador, England 2-2, friendly, Rooney	Ecuador, England, 2-2, world-cup, Honduras, wales
Main Story: Harrison Ford broke left leg in accident on 19/6/2014 (see Table I)				
Expert 1# annotation:	**Expert 2# annotation:**	**Expert 3# annotation:**	**Union**	**Produced**
Harrison, Ford, leg, broke	Harrison, ford, leg, broke, accident	Harrison, Ford, leg, broke	Harrison, Ford, leg, broke, accident	Harrison, ford, broke leg, publicist confirms

TABLE IV
Experts Annotation, Ground Truth, and Produced Events for Some Story Examples of Scenario 2

Main Story: Israel Gaza Conflict on 1-15 /7/2014 (see Table II)				
Expert 1# annotation:	**Expert 2# annotation:**	**Expert 3# annotation:**	**Union**	**Produced**
Ebola Virus Africa	Ebola battle West Africa	Ebola Africa Death	Ebola Virus battle West Africa death	Ebola West Africa
Main Story: Ebola battle at west Africa on 1-15 /7/2014 (see Table II)				
Expert 1# annotation:	**Expert 2# annotation:**	**Expert 3# annotation:**	**Union**	**Produced**
Gaza Conflict people kills	Gaza Israel Conflict	Gaza Israel Conflict deaths	Gaza Israel Conflict, deaths, people kills	Gaza Israel Conflict Air Attacks kills children mothers

The experts examined all the time intervals, each performing 120 annotations (number of six-hour intervals in a one month period). The number of stories that each expert extracted depended on the topic category (sports, news), the time interval and the expert's interpretation, and it varied from two to six stories per category, expert, and time interval. The three experts produced near duplicate stories. We allowed the experts to remove noisy words and/or add new words in order to construct reliable sets of events (clusters). The union of all events generated by the three experts is considered as the ground-truth dataset. Note that by depicting to the experts the 20 most frequent words, we significantly reduced the manual effort required for the annotation. The dataset has been collected under the framework of a research European Union project and the availability of the data undergone consensus of the consortium and ethics issues.

Table III shows the annotation results generated by each expert for some of the stories presented in Table I for scenario 1. In the same table, we depict the union of all experts as ground truth data. For the scenario 2 case, we automatically brought before the experts the 20 words that most frequently appeared within a 2-day time period (time interval for scenario 2), using again the WF counts (see Table IV). Then, the experts filtered out these events every 15 days to define the most salient topics that occurred within half a month. Thus, we first extracted 90 sets of most frequent words

(a two day time interval for six months) and then we created 12 higher topic activities (each topic activity per 15 days).

3) Preprocessing Steps: Within a time interval, only the most frequent words are considered. As for the ground truth case, we count words' frequencies for all the retrieved tweets using the Natural Language Toolkit of Python. The toolkit has the option of handling synonyms and abbreviations, using the lexical database of WordNet. To overcome the problem that certain types of words, such as articles, Internet slangs, common verbs/nouns, and abbreviations do not contribute to the frequency of an event, we filtered out these words using the http://www.noslang.com/dictionary, taking into consideration only verbs and nouns. Words with very low appearance frequencies will eventually exhibit low values in tweet characterization metric and have less influence on the event detection process. In our experimental set-up, the words under examination were three times more than the ones extracted in the annotation phase. Thus, only a small set of words were delivered to the system for processing, significantly reducing the size of the graph used for keywords extraction and consequently the computational processing requirements. To reach a fair comparison among our proposed algorithm and the methods presented in the literature (see Section V-D), only the tweets that remain after the preprocessing step (e.g., tweets that contain any of the selected most frequent words) are considered in the comparative study.

For scenario 1, four previous time intervals were examined to evaluate the fuzzy metrics using triangular membership functions. The time signal was of size 20 intervals (five days). Thus, the size of the wavelet signal was equal to $q = 20$. Similarly, we set $p = 40$ as a period to evaluate the ITF. Regarding scenario 2, wavelet signals were generated every 15-day nonoverlapping time periods, i.e., intervals on which the most prominent events were defined. Then, the event detection algorithm synopsized what had mostly happened during that six month period.

B. Objective Evaluation Metrics

We used the objective criteria of precision and recall to evaluate the efficiency of the proposed algorithm and compare it to that of other methods. In this paper, an event is considered as a collection of keywords and therefore precision/recall of an event is evaluated over the keywords this event is composed of (keyword precision/recall).

To define these criteria, we need to match the produced events to the ground truth ones. This is achieved by utilizing the Hungarian algorithm to avoid matching of a produced event to more than one ground truth sets and to solve the assignment combinatorial problem in polynomial time [17].

Let us denote by $W(m), m = 1, 2, \ldots, M$, the set of produced events, where the number M is optimized as described in Section III-E. We also denote by $W_{\text{gt}}(m), m = 1, 2, \ldots, M_{\text{gt}}$, the M_{gt} ground truth events generated by the experts, as described in Section V-A. The cost used for the Hungarian algorithm to relate the mth ground truth set $W_{\text{gt}}(m)$ with the lth produced event $W(l)$ is through the F_1 score $F_1(m, l)$ that

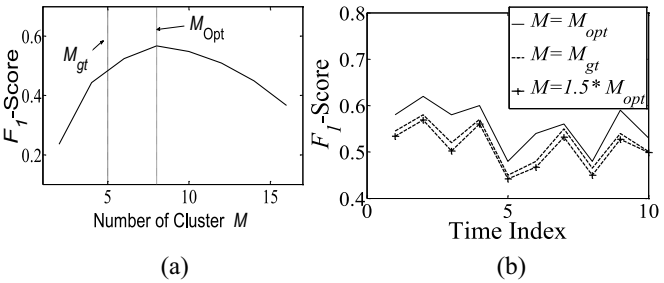

Fig. 7. (a) Effect of M on the F_1-score using training data of the scenario 1 case. Vertical lines correspond to M_{opt} and M_{gt}. (b) F_1-score versus ten different time index intervals of the validation dataset, for $M = M_{\text{opt}}$, $M = M_{\text{gt}}$, and $M = 1, 5^* M_{\text{opt}}$.

compensates both precision $\text{PR}(m, l)$ and recall $\text{RE}(m, l)$ values

$$F_1(m, l) = 2 \cdot \frac{\text{PR}(m, l) \cdot \text{RE}(m, l)}{\text{PR}(m, l) + \text{RE}(m, l)} \tag{25}$$

with

$$\text{PR}(m, l) = \frac{W(l) \bigcap W_{\text{gt}}(m)}{\|W(l)\|} \text{RE}(m, l) = \frac{W(l) \bigcap W_{\text{gt}}(m)}{\left\|W_{\text{gt}}(m)\right\|} \tag{26}$$

where $\|\cdot\|$ denotes set cardinality. The optimum assignment is performed so as to minimize the cost $1\text{-}F_1(m, l)$ among the lth produced and the mth ground truth event.

C. Parameter Selection

The performance of the proposed algorithm is affected by the choice of a number of parameters involved. We experimentally tune these parameters in order to maximize the algorithm's performance. For both the scenario 1 and the scenario 2 cases, we initially split the ground truth dataset into a training set and a validation set. Then, we concentrate on estimating the number M of produced events. Formally, M is determined by optimizing the cost function J of (21). In the following, we experimentally validate the theoretical approach using data of the training set. Since parameter M affects both precision and recall, the F_1-score is used for the validation. Fig. 7(a) depicts the F_1-score for different choices of M for the scenario 1 case. As is observed, the F_1-score is maximized at the optimal value M_{opt} of (21), which is also depicted in Fig. 7(a). In the same figure, we also depict the average number of ground truth events M_{gt}, which is found to be greater than M_{opt}. This is mainly because there are several words that do not belong to any of the ground truth annotated events. By setting $M = M_{\text{gt}}$, these additional keywords are classified as ground truth events, decreasing the F_1-score. In contrast, by setting M slightly greater than M_{gt}, these nonannotated keywords are assigned to the additional created clusters, improving classification accuracy. However, as M becomes large the ground truth events are also split into more than one clusters, decreasing F_1-score.

Fig. 7(b) justifies the M_{opt} value, depicting the F_1-score for ten different time index intervals of data of the validation set. The results are compared with $M = M_{\text{gt}}$ and $M = 1.5^* M_{\text{opt}}$. We observe that the F_1-score is maximized when $M = M_{\text{opt}}$. Similar conclusions are drawn for the

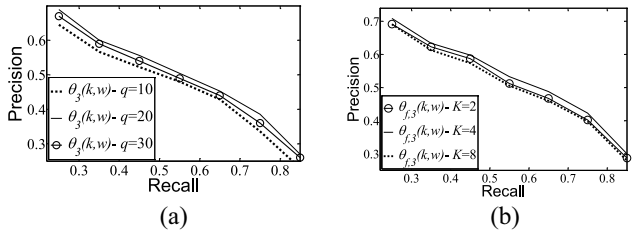

Fig. 8. (a) Precision–recall curve for different scale-space (wavelet) parametric values q, under $\vartheta_3(k, w)$ metric. (b) Precision–recall curve for different fuzzy parameters K, at $q = 20$ and under $\vartheta_3(k, w)$ metric.

TABLE V
EFFECT OF DIFFERENT PARAMETERS FOR BOTH SCENARIOS

Scenario 1			Scenario 2		
q=10	**q=20**	q=30	q=5	**q=7**	q=10
0.517	**0.534**	0.525	0.469	**0.486**	0.473
K=2 & q=20	**K=4 & q=20**	K=8 & q=20	K=2 & q=7	**K=4 & q=7**	K=8 & q=7
0.544	**0.558**	0.539	0.477	**0.505**	0.488

scenario 2 case, where $M = M_{\text{opt}}$ maximizes classification performance.

The second parameter that has to be tuned is the length q of the wavelet time signal. Table V shows the effect on the F_1-score of different values of q, when $M = M_{\text{opt}}$. The results for both scenarios 1 and 2 were obtained using data from the training set. For the wavelet length optimization, no fuzzy representation was adopted. We see that the optimal value for scenario 1 is $q = 20$, while for scenario 2 it is $q = 7$. In general, small values of q fail to model the temporal behavior of a word's distribution, while for large values of q (long time horizon signals) the same word corresponds to different actual events, deteriorating again classification accuracy. In Table V, we also depict the effect of the parameter K used in the fuzzy representation. We use, in both scenarios, the best values of q as determined above. Again, small values of K fail to compensate for the temporal delays encountered in a word's posting for the same event, while large values of K result in smooth signals, deteriorating classification performance.

In Fig. 8(a), we validate the efficiency of selecting $q = 20$ for the scenario 1 case by curving the precision versus recall for $q = 10$, $q = 20$, and $q = 30$. The results have been obtained on data from the validation set. We see that even for different recall values (which also affects the number M of events), $q = 20$ yields the highest precision verifying the selection for that type of data. The results have been obtained using no fuzzy representation. In Fig. 8(b), we plot the same curve but under different values of K used in the fuzzy scheme. Again, $K = 4$ yields the highest performance for all recall values. In that case, we select $q = 20$. We should note here that all the previous results were obtained using $\vartheta_3(k, w)$, since this metric provides better characterization than the other two metrics (see Section V-D).

Similar, validation conclusions are derived for the scenario 2 case that verifies the parameters selected in Table V. We should stress that the exact values of the parameters depend on the types of topics used and the characteristics of the tweet messages. Different optimization may be concluded for a different dataset but the values will be more or less within the predefined ranges.

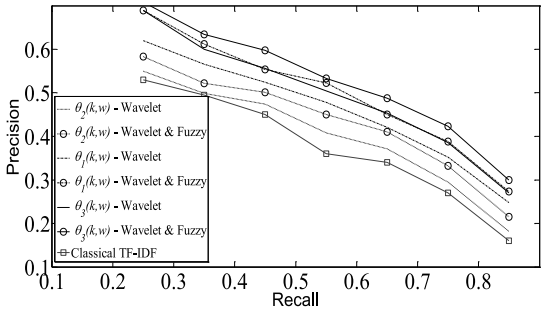

Fig. 9. Precision–recall curve for the three proposed metrics depicting the effect of fuzzy representation and scale-space (wavelet) transformation for the scenario 1 case.

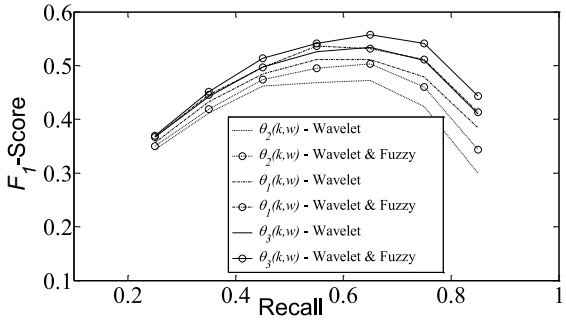

Fig. 10. F_1-score versus recall for the three proposed metrics depicting the effect of fuzzy representation and scale-space (wavelet) transformation for the scenario 1 case.

D. Evaluation for Scenario 1

Fig. 9 presents the precision–recall results for the three metrics $\vartheta_i(k, w)$ $i = 1, 2$, and 3 when the fuzzy and wavelet representation described in Section II is applied under scenario 1. Precision/recall is estimated in this case for the most salient event (cluster), by maximizing $\sum_{w_i \in W(m)} \vartheta(k, w_i)$. We observe that $\vartheta_3(k, w)$ metric yields higher precision values for the same recall than the other two metrics, while the lowest precision values are achieved for metric $\vartheta_2(k, w)$. An improvement in the precision values is also noticed for the case of both fuzzy and scale-space representation due to better modeling of the dynamic nature of Twitter. In this figure, we also included the classical TF–IDF score. This score resembles $\vartheta_2(k, w)$ with the difference that inverse frequency is now consider constant. Instead, in $\vartheta_2(k, w)$ the inverse frequency is time varying signal to better approximate Twitter dynamicity.

In Fig. 10, the F_1-score versus the recall value is depicted for the real data experiments. We observe that the highest F_1-scores are obtained for recall RE $= 0.65$, meaning that at these recall values, cluster elements contain not only more relevant data but also much of them (many relevant words). Instead, for low recall, F_1 is low even though precision is high. This is because a low recall value corresponds to a large number of clusters, each containing few elements (words).

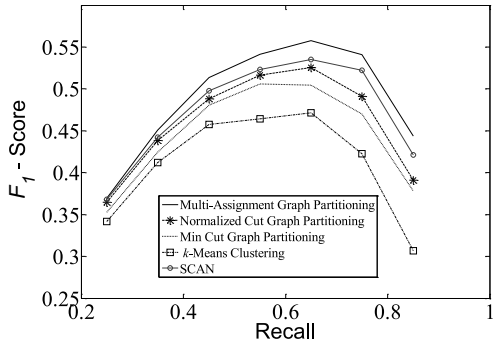

Fig. 11. F_1-score versus recall for different clustering algorithms for scenario 1. For all comparisons, a time series signal of fuzzy elements of $\vartheta_{f,3}(k, w)$ tweet post characterization metric has been used.

Most of the words in these clusters are relevant but they are few compared to the actual number of words needed to represent this particular event. Similarly, at high recall values, few clusters are constructed, and thus each event contains most of the relevant words. However, in this case other words are also assigned to the same cluster decreasing precision and thus the F_1-score. Since in tweet posts the number of events is actually unknown, F_1-score can be seen as a proper metric to estimate the more suitable number of clusters, that is, through the recall value. Again, the combination of fuzzy and scale-space representation improves event detection performance.

The effect of different clustering algorithms on the F_1-score versus recall curve is shown in Fig. 11. In this case, the results have been obtained using a time series of metric $\vartheta_{f,3}(k, w)$ after being fuzzy represented and scale-space transformed, since this representation provided the highest precision accuracy for a given recall value. In Fig. 11, we compare our multiassignment graph partitioning approach of Section III, with three other clustering methods: 1) the conventional *k*-means; 2) the un-normalized min cut graph partitioning; and 3) the normalized cut graph partitioning [7] and the structural clustering algorithm for networks (SCAN) [18]. We can see that the proposed clustering method outperforms the other three methods. This is because our clustering methodology allows one word to belong to multiple clusters, as is the actual case, while simultaneously finding the discrete optimum solution that better approximates the continuous one. The SCAN also assigns works to multiple clusters since hubs are considered as keywords that belong to several events.

Fig. 12(a) illustrates the comparative performance between the proposed method and three other techniques: 1) the graph centrality method [19]; 2) the latent Dirichlet allocation (LDA) approach; and 3) the method of [20], where event detection in Twitter is based on a combination of LDA with the PageRank algorithm. For all the compared methods, appropriate tuning of the parameters involved was carried out to optimize their performance. Particularly, as for the graph centrality approach [19] two heuristics are examined: 1) the all neighbors and 2) the nearest neighbors edging approach. The first approach gives higher performance than the second one, as is also verified in [19]. However, the main difficulty in

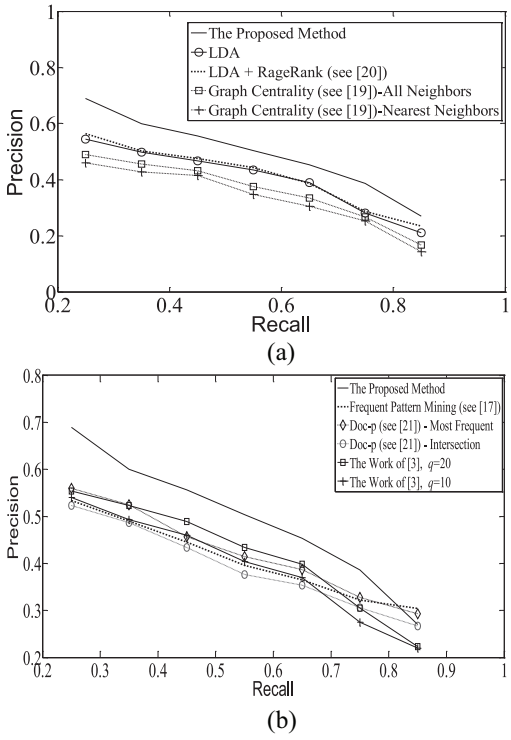

Fig. 12. Comparisons of the proposed method using other methodologies regarding the precision–recall curve for the scenario 1 configuration.

implementing the algorithm of [19] is that it actually ranks keywords instead of creating groups of clusters (events) as in our case. To cluster keywords into different groups a simple nearest neighbors approach is adopted, as in [21]. In Fig. 12(a), however, we vary the recall value, which also affects the choice of *M*, to validate precision robustness against different number of clusters. Thus, in our implementation we set the same *M* for all compared methods for fairness, as the one that yields the specific recall value.

Regarding LDA, the number of clusters *M* should be known *a priori*. Again, in Fig. 12(a), *M* is appropriately regulated so in order to yield the specific recall value. Finally, [20] combines LDA with PageRank criterion.

Fig. 12(b) also compares our method with: 1) the document-pivot approach [21]; 2) the wavelet-based method [3]; and 3) frequent pattern mining [22]. The document-pivot method clusters tweet messages without supervision, mainly exploiting a threshold-based nearest neighboring technique. The threshold parameter affects the number of clusters and consequently the recall value of Fig. 12(b). The main difficulty of [21] is that it groups tweets together instead of extracting a set of keywords that represent an event. To handle this, we can use a number of different heuristics. The first is to take the union of all tweet words in a cluster. In this way, the recall value increases but we obtain low precision. The second heuristic takes into account the intersection of all tweet words of the same cluster. In this way, the precision is improved but the recall value decreases. Another heuristic is to exploit the $\vartheta_3(k, w)$ score to rank words within a cluster. Then, keywords are extracted at the index where a drop in cumulative ranked

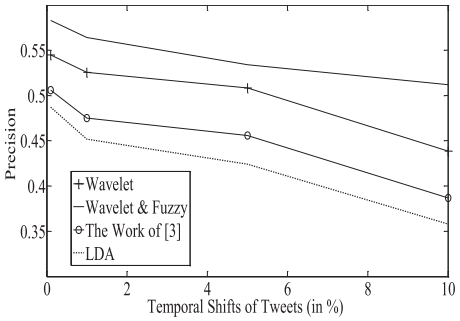

Fig. 13. Precision versus temporal tweets shifts with the wavelet and the fuzzy–wavelet representation using $\vartheta_{f,3}(k,w)$ and comparisons with other methods for recall RE = 0.4.

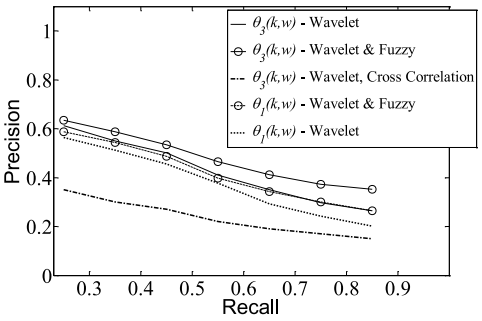

Fig. 14. Precision–recall curve for the scenario 2 case where the Riemannian distance is exploited. The results has been illustrated when we apply the fuzzy–wavelet representation and the only wavelet representation for the two metrics $\vartheta_1(k,w)$ and $\vartheta_3(k,w)$.

distribution of $\vartheta_3(k,w)$ is noticed. We call this heuristic "most-frequent approach." Fig. 12(b) verifies that the third heuristic of the document pivot method provides better results than the other two.

Regarding the wavelet-based method of [3], the main parameter involved is the wavelet length q, whose proper selection was an issue for our algorithm also. In all comparisons, we selected $q = 20$ since this value optimizes clustering performance. Again, in Fig. 12(b), we present results for $q = 10$ and $q = 20$ to show how the wavelet length parameter affects the clustering performance. Finally, the performance of frequent pattern mining technique [22] depends on a threshold through which the most frequent co-occurring words are marked. This threshold affects the number of clusters and is set, in our case, to a certain value to yield the target recall value.

As is observed in both Fig. 12(a) and (b), our method outperforms the compared ones, since it better models the dynamic behavior of Twitter data. The graph centrality method and the RageRank-based algorithm present the limitation that words exhibiting a high centrality metric often belong to different event clusters. The LDA approach accurately models narrow topic scopes. The scale-space wavelet representation [3] seems to present the most robust performance, but as the recall increases its efficiency tends to decay more rapidly. The frequent pattern mining technique [22] behaves robustly for higher recall values.

In Fig. 13, we plot precision versus the degree of temporal shifts (variations) of the tweets, for a given recall value RE = 0.4. In particular, the experiment performed was the following. We shifted from one time interval to another a percentage $a\%$ of the tweets that contained the most frequent words. In this way we contaminate a time interval with tweets containing salient keywords posted at previous time intervals. This aims at simulating in a realistic way the dynamic nature of the tweets in order to examine the robustness of our algorithm under asynchronous postings. We notice that as the percentage of tweet temporal shifts increases, precision decreases for all metrics, since more tweets are posted asynchronously (more time noise). We can see from Fig. 13 that the use of fuzzy–wavelet representation makes the proposed event detection algorithm more robust to such noise. In this figure, we also compare the robustness of precision values with respect to temporal shifts for two other methods, namely the wavelet-based algorithm of [3] and LDA. We observe

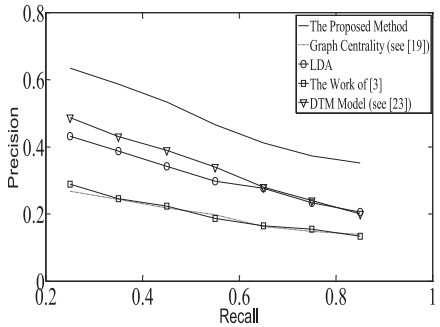

Fig. 15. Comparisons of the proposed method with others regarding precision–recall curve for scenario 2; (a) wavelet-based method of [3] and (c) DTM model of [23].

that our proposed fuzzy–wavelet representation yields more robust results in terms of precision accuracy, for all values of the temporal shift parameter a, than the other algorithms examined.

E. Evaluation for Scenario 2

Fig. 14 shows the performance of the combined fuzzy–wavelet presentation using the two most prominent information theoretic metrics, that is, $\vartheta_1(k,w)$ and $\vartheta_3(k,w)$ with and without fuzzy representation. We can see that fuzzy provides a slight improvement of the precision values for all recalls. Again, the highest performance is given for the $\vartheta_3(k,w)$ metric.

In the same figure, we present the effect of cross-correlation distance on precision–recall performance as opposed to the Riemannian one, using $\vartheta_3(k,w)$ metric. It is clear that the cross-correlation gives much lower precision than the Riemannian distance due to its sensitivity to words' temporal shifts.

In Fig. 15, we compare our Riemannian-distance based algorithm with four other approaches: 1) the graph centrality method presented in [19]; 2) the LDA approach for modeling tweet content; 3) the scale-space representation of [3]; and 4) the dynamic topic model (DTM) of [23] which is closer to the scenario 2 case. We observe that our scheme significantly improves precision for a given recall value when compared to the other methods examined, thanks to the use of a more appropriate distance, namely the Riemannian one. Again, we have optimized the parameters of all the algorithms.

Table III presents examples of produced events from our algorithm for both scenarios 1 and 2. The results indicate that most of the keywords of the ground truth set have been extracted. The performance of the compared algorithms depends on the diversity of the tweet content. For instance, the document-pivot method performs well in case we retrieve many tweets of quite similar content, but its performance decays for events consisting of tweets of diverse content. This observation implies that document pivot is not suitable for scenario 2, splitting for example the Gaza–Israel conflict event into multiple clusters. Instead, the Harrison Ford's broken leg event of scenario 1 is well captured by this algorithm.

Regarding the computational complexity of our proposed algorithm, the main bottleneck corresponds to the multiassignment clustering and particularly to the eigenvalue decomposition. The fastest implementation for the eigenvalue decomposition is through the Lanczos method whose complexity is $O(M \cdot L^2 \cdot \tau)$, where M is the number of the number of events, L is the number of words, and τ is the number of iterations of the algorithm. Since M is usually several times smaller than L the complexity is of order $O(L^2)$. In our implementation, we keep L small by ignoring the words that exhibit low values in WF appearance making the actual computational time quite affordable.

VI. Previous Works

News content aggregators represent one of the first approaches in structuring tweet content. The approach in [24] geographically classifies tweets, while [25] introduces a demo that allows users to submit textual queries. A users/topics statistical analysis is given in [26]. Other methods statistically rank tweets, either using retweets/followers' properties [27], or URL links [28], or social scores [29]. In this context, the work of [30] classifies similar micro-bloggers into topical clusters by detecting representative authoritative authors, [31] and [32] exploit link-based methodologies (the followers' PageRank [31], or the topical sensitive PageRank [32]), while [29] uses a popularity score defined on the retweeting count. A social voting advice application has been proposed in [33]. However, all these approaches are in fact indirect event detection tools (e.g., search engines, or ranking mechanisms).

Regarding event-detection algorithms for Twitter, two main approaches exist [3], [22]: 1) the documents-based and 2) the features-based approach. The first detects events by clustering tweet posts based on their similarity, while the second attempts to detect events by clustering features. Document-based approaches, such as those proposed in [34]–[37], first perform clustering and then proceed to extract the most salient terms. However, they suffer from cluster fragmentation and thresholding issues, especially for time varying signals [22].

Feature-based approaches extract sets of terms that model trending topics occurring in Twitter. We focus mostly on these algorithms, since our approach belongs to this category. One of the most commonly used methods is based on LDA [38]. The works of [20]–[42] combine LDA with graph models and appropriate features (e.g., Twitter's followers). Geographic-based topic analyses using LDA models have been presented in [43] and [44]. Sentiment variations on Twitter are detected in [45] by introducing a foreground/background LDA model, while [46] presents on-line adaptive models. LDA-based methods perform well in cases of narrow topic stories but they fail in capturing broader events [22], [47]. To address these difficulties, modifications of LDA models have been proposed, such as the DTM [48] for applying sequential summarization of trending stories [23] and the location-time constrained topic for capturing the spatial and temporal content properties [49].

Other approaches exploit burst patterns of a word's distribution that create a co-occurrence graph and then applies graph centrality criteria for extracting the keywords [19] or use the SCAN [18] for identifying trending events [22]. Graph-based methods have been used for sentiment analysis [51], performing poorly when detecting closely interconnected events. For this reason, Aiello *et al.* [22] introduced a frequent pattern mining method, while [52] analyzes temporal feature trajectories for event detection and then applies the DFT on the time word signals. The events are detected as spikes in the frequency domain. However, when using DFT, time information is lost. This problem is also verified in [52], where Gaussian mixture models are proposed to address it or the wavelet transformation [3].

VII. Conclusion

In this paper, we proposed a new event detection algorithm suitable for tweets. To address the dynamic nature of tweets messages, first we construct fuzzy signals, obtained from information theoretic metrics, appropriately modified to capture tweets characteristics. Experimental results on real-life data indicate that the proposed fuzzy time signals better model tweet dynamic behavior than the conventional TF–IDF scores since they are able to capture Twitter dynamics. In addition, the words' clustering was modeled as a multiassignment graph partitioning problem that allows one word to belong to several clusters (events). The conclusions are that multiassignment clustering outperforms conventional graph partitioning methods. Finally, the paper exploits Riemannian distance metrics between word signatures in order to compensate tweet submission delays. Experimental evidence indicates the superior performance of our approach.

In case that we have more unstructured forms of datasets, we yield closely interconnected events that share a large number of common keywords, reducing precision, and recall. However, the proposed approach is more robust toward unstructured datasets than other methods, since pairwise similarities are exploited and words are intercorrelated based on fuzzy time feature series.

References

[1] A. Sun and M. Hu, "Query-guided event detection from news and blog streams," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 5, pp. 834–839, Sep. 2011.

[2] L. B. Jabeur, L. Tamine, and M. Boughanem, "Uprising microblogs: A Bayesian network retrieval model for tweet search," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Trento, Italy, 2012, pp. 943–948.

[3] J. Weng and B.-S. Lee, "Event detection in Twitter," in *Proc. AAAI Conf. Weblogs Soc. Media (ICWSM)*, Barcelona, Spain, 2011, pp. 401–408.

[4] A. Aizawa, "The feature quantity: An information theoretic perspective of Tfidf-like measures," in *Proc. ACM SIGIR*, Athens, Greece, 2000, pp. 104–111.

[5] X.-B. Xue and Z.-H. Zhou, "Distributional features for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 3, pp. 428–442, Mar. 2009.

[6] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "A fuzzy video content representation for video summarization and content-based retrieval," *Signal Process.*, vol. 80, no. 6, pp. 1049–1067, Jun. 2000.

[7] J. Shi and J. Malik, "Normalized cut and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[8] K. Fan, "Maximum properties and inequalities for the eigenvalues of completely continuous operators," *Proc. Nat. Acad. Sci.*, vol. 37, no. 11, pp. 760–766, 1951.

[9] N. D. Doulamis, P. Kokkinos, and E. Varvarigos, "Resource selection for tasks with time requirements using spectral clustering," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 461–474, Feb. 2014.

[10] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. IEEE ICCV*, vol. 1. Nice, France, 2003, pp. 313–319.

[11] A. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2001, pp. 849–856.

[12] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2004, pp. 1601–1608.

[13] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2002, pp. 1255–1262.

[14] A. Doulamis and N. Doulamis, "Performance evaluation of Euclidean/correlation-based relevance feedback algorithms in content-based image retrieval systems," in *Proc. IEEE ICIP*, vol. 1. Barcelona, Spain, Sep. 2003, pp. 737–740.

[15] N. Doulamis and A. Doulamis, "Evaluation of relevance feedback schemes in content-based in retrieval systems," *Signal Process. Image Commun.*, vol. 21, no. 4, pp. 334–357, Apr. 2006.

[16] W. Forstner and B. Moonen, "A metric for covariance matrices," *Geodesy—The Challenge of the 3rd Millennium*. Berlin, Germany: Springer, pp. 299–309, 2003.

[17] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

[18] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "SCAN: A structural clustering algorithm for networks," in *Proc. KDD 13th ACM Int. Conf. Knowl. Disc. Data Mining*, San Jose, CA, USA, 2007, pp. 824–833.

[19] W. D. Abilhoa and L. N. De Castro, "A keyword extraction method from Twitter messages represented as graphs," *Appl. Math. Comput.*, vol. 240, pp. 308–325, Aug. 2014.

[20] A. Bellaachia and M. Al-Dhelaan, "Learning from Twitter hashtags: Leveraging proximate tags to enhance graph-based keyphrase extraction," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, Besancon, France, 2012, pp. 348–357.

[21] S. Petrovic, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to Twitter," in *Proc. HLT Annu. Conf. North Amer. Ch. Assoc. Comput. Linguist.*, Los Angeles, CA, USA, 2010, pp. 181–189.

[22] L. M. Aiello *et al.*, "Sensing trending topics in Twitter," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1268–1282, Oct. 2013.

[23] D. Gao, W. Li, X. Cai, R. Zhang, and Y. Ouyang, "Sequential summarization: A full view of Twitter trending topics," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 2, pp. 293–302, Feb. 2014.

[24] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitterstand: News in tweets," in *Proc. 17th ACM Int. Conf. Adv. Geogr. Inf. Syst.*, Seattle, WA, USA, 2009, pp. 42–51.

[25] M. Grinev *et al.*, "Sifting micro-blogging stream for events of user interest," in *Proc. 32nd ACM SIGIR Res. Develop. Inf. Retriev.*, Boston, MA, USA, 2009, p. 837.

[26] M. Cha, F. Benevenuto, H. Haddadi, and K. P. Gummadi, "The world of connections and information flow in Twitter," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 4, pp. 991–998, Jul. 2012.

[27] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in Twitter: The million follower fallacy," in *Proc. 4th Int. AAAI Conf. Weblogs Soc. Media (ICWSM)*, Washington, DC, USA, 2010, pp. 10–17.

[28] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum, "An empirical study on learning to rank of tweets," in *Proc. 23rd Int. Conf. Comput. Linguist. (COLING)*, Stroudsburg, PA, USA, 2010, pp. 295–303.

[29] R. Nagmoti, A. Teredesai, and M. De Cock, "Ranking approaches for microblog search," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Tech.*, vol. 1. Toronto, ON, Canada, 2010, pp. 153–157.

[30] A. Pal and S. Counts, "Identifying topical authorities in microblogs," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, Hong Kong, 2011, pp. 45–54.

[31] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proc. 19th ACM Int. Conf. World Wide Web (WWW)*, Raleigh, NC, USA, 2010, pp. 591–600.

[32] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential Twitterer," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, 2010, pp. 261–270.

[33] I. Katakis, N. Tsapatsoulis, F. Mendez, V. Triga, and C. Djouvas, "Social voting advice applications-definitions, challenges, datasets and evaluation," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1039–1052, Jul. 2014.

[34] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in Twitter," in *Proc. IEEE/ACM Web Intell. Intell. Agent Tech. Conf.*, vol. 3. Toronto, ON, Canada, 2010, pp. 120–123.

[35] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitter stand: News in tweets," in *Proc. GIS 17th ACM Int. Conf. Adv. Geogr. Inf. Syst.*, New York, NY, USA, 2009, pp. 42–51.

[36] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory search and topic summarization for Twitter," in *Proc. 4th Int. AAAI Conf. Weblogs Soc. Media*, Washington, DC, USA, 2010, pp. 384–385.

[37] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on Twitter," in *Proc. 5th Int. AAAI Conf. Weblogs Soc. Media*, Barcelona, Spain, 2011, pp. 438–441.

[38] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[39] A. Bellaachia and M. Al-Dhelaan, "NE-Rank: A novel graph-based keyphrase extraction in Twitter," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Tech.*, vol. 1. Macau, China, 2012, pp. 372–379.

[40] Y. Kim and K. Shim, "TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation," *Inf. Syst.*, vol. 42, pp. 59–77, Jun. 2014.

[41] M.-C. Yang and H.-C. Rim, "Identifying interesting Twitter contents using topical analysis," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4330–4336, Jul. 2014.

[42] S. Yamamoto and T. Satoh, *Two Phase Extraction Method for Extracting Real Life Tweets Using LDA* (LNCS), Berlin, Germany: Springer, 2013, pp. 340–347.

[43] S. Sizov, "GeoFolk: Latent spatial semantics in Web 2.0 social media," in *Proc. WSDM*, New York, NY, USA, 2010, pp. 281–290.

[44] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical topic discovery and comparison," in *Proc. WWW*, Hyderabad, India, 2011, pp. 247–256.

[45] S. Tan *et al.*, "Interpreting the public sentiment variations on Twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1158–1170, May 2014.

[46] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *Proc. ICDM*, Pisa, Italy, 2008, pp. 3–12.

[47] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving LDA topic models for microblogs via tweet pooling and automatic labeling," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retriev.*, 2013, pp. 889–892.

[48] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. ICML*, Pittsburgh, PA, USA, 2006, pp. 113–120.

[49] X. Zhou and L. Chen, "Event detection over Twitter social media streams," *Int. J. Very Large Data Bases (VLDB)*, vol. 23, no. 3, pp. 381–400, 2014.

[50] B. Di Eugenio, N. Green, and R. Subba, "Detecting life events in feeds from Twitter," in *Proc. IEEE 7th Int. Conf. Semant. Comput.*, Irvine, CA, USA, 2013, pp. 274–277.

[51] A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Urena-López, "Ranked WordNet graph for sentiment polarity classification in Twitter," *Comput. Speech Language*, vol. 28, no. 1, pp. 93–107, 2014.

[52] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retriev.*, Amsterdam, The Netherlands, 2007, pp. 207–214.

Nikolaos D. Doulamis (S'96–M'00) received the Diploma (Hons.) and Ph.D. (Hons.) degrees in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1995 and 2001, respectively.

He is currently an Assistant Professor with the NTUA. He has authored over 50 (150) journals (conference) papers in the field of video analysis and learning. He has over 2200 citations and is involved in large scale European projects.

Dr. Doulamis was a recipient of several awards, such as the Best Greek Engineer Student, the Graduate Thesis Award, the NTUAs Best Young Medal, and best paper awards in IEEE conferences. He has served as an Organizer and/or Program Committee Member of major IEEE conferences.

Anastasios D. Doulamis (S'96–M'00) received the Diploma (Hons.) and Ph.D. (Hons.) degrees in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1995 and 2001, respectively.

Until 2014, he was an Associate Professor with the Technical University of Crete, Chania, Greece. He is currently a Faculty Member with the NTUA. He has authored over 200 papers in leading journals and conferences, receiving over 2000 citations.

Prof. Doulamis was a recipient of several awards in his studies, including the Best Greek Student Engineer, the Best Graduate Thesis Award, and the National Scholarship Foundation Prize. He has also served on the Program Committee of several major conferences of IEEE and ACM.

Panagiotis Kokkinos received the Diploma degree in computer engineering and informatics and the M.S. degree in integrated software and hardware systems from the University of Patras, Patras, Greece, in 2003 and 2006, respectively, where he is currently pursuing the Ph.D. degree in computer engineering and informatics.

Emmanouel (Manos) Varvarigos received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1988, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1990 and 1992, respectively.

He has held faculty positions with the University of California at Santa Barbara, Santa Barbara, CA, USA, from 1992 to 1998, and the Delft University of Technology, Delft, The Netherlands, from 1998 to 2000. Since 2000, he has been a Professor of Computer Engineering and Informatics, with the University of Patras, Patras, Greece, being involved in pioneering research and development projects at European level.

Prof. Varvarigos has served on the organizing and program committees of several IEEE conferences.