Mining Temporal Evolution of Entities in a Stream of Textual Documents

Gianvito Pio, Pasqua Fabiana Lanotte, Michelangelo Ceci, and Donato Malerba

University of Bari A. Moro
Dept. of Computer Science - Via Orabona, 4 - I-70125 Bari, Italy
{name.surname}@uniba.it

Abstract. One of the recently addressed research directions focuses on the problem of mining topic evolutions from textual documents. Following this main stream of research, in this paper we face the different, but related, problem of mining the topic evolution of entities (persons, companies, etc.) mentioned in the documents. To this aim, we incrementally analyze streams of time-stamped documents in order to identify clusters of similar entities and represent their evolution over time. The proposed solution is based on the concept of temporal profiles of entities extracted at periodic instants in time. Experiments performed both on synthetic and real world datasets prove that the proposed framework is a valuable tool to discover underlying evolutions of entities and results show significant improvements over the considered baseline methods.

1 Introduction

Topic Detection and Tracking (TDT) [3,22,5,10] is an important research area which applies data mining algorithms in order to find and follow topics in streams of news or, in general, in streams of textual documents. According to the classification suggested in [7], there are three main research lines in TDT: *i) Segmentation* - documents coming from a stream are clustered according to their topic. Each cluster represents the same topic across the time dimension. *ii) New topic detection* - new clusters are identified in the stream. *iii) Topic tracking* - evolutions of clusters are tracked. In this case, new documents can be associated to existing clusters, causing changes in clusters' properties.

By focusing our attention on topic tracking, in this paper, we argue that it is possible to use such techniques to discover evolutions of entities over time. We focus on entities (e.g. people, organizations) having particular roles (e.g. perpetrator, victim, in the risk identification and analysis domain) in particular types of domain-dependent relationships (e.g. kill, steal). These entities are considered as units of analysis. In this respect, the proposed framework identifies such entities and incrementally analyzes streams of documents in order to discovery clusters of "similar" entities and represent their evolution over time. To this aim, we apply a time-slice density estimation method [2] that allows us to represent the profile of each entity. Moreover, it allows us to analyze profiles evolution by measuring the rate of change of properties and peculiarities of entities activities'

T. Andreasen et al. (Eds.): ISMIS 2014, LNAI 8502, pp. 50–60, 2014.

over a given time horizon. At this purpose, we apply a time-slice density estimation method [2] that allows us to represent the profile of each entity. Moreover, it allows us to analyze profiles evolution by measuring the rate of change of properties and peculiarities of entities activities' over a given time horizon.

In the literature, several papers have faced the problem of mining evolutions in streams of documents and, in particular, the problem of tracking topics, ideas and "memes" [14,24]. However, most of the work considers single keywords or short phrases in the documents as units of analysis. On the contrary, we consider as units of analysis the entities that can be associated with (identified in) the documents. This means that we identify the evolution of entities by analyzing documents they are associated with. Evolutions are expressed according to relevant terms that allow us to represent and characterize entities. From the methodological viewpoint, we do not identify evolutions by evaluating whether a particular data mining model has become stale because of a change in the underlying data distribution [13,2], but we provide the user with an understanding of the changes, according to a content-based representation of the entities' profiles (entities are represented according to terms occurring in the textual documents).

The proposed framework could be profitably exploited in different application domains. For instance, in the analysis of papers belonging to the medical domain, it could support researchers in the identification of evolutions about the recognized role of biological entities (e.g. genes) over time. Another example is represented by the risk identification and analysis domain, which is considered in this paper. In this case, using publicly available news (e.g. daily police reports, public court records, legal instruments) about criminals, and assimilating the concept of *topic* with the concept of *crime typology* represented by a group of "similar" criminals, the proposed method can be considered as a valuable tool for law enforcement officers in risk and threat assessment.

The contribution of this paper is manifold: on the basis of entities identified in the documents, *i)* we define an unsupervised feature selection algorithm which overcomes limitations of existing unsupervised methods *ii)* we represent the entities' profile and on-line modify it according to more recent documents; *iii)* we generate clusters of similar entities and represent and analyze their evolution.

2 Related Work

In the literature, a variety of approaches to deal with evolving clusters from textual data streams can be found. For example, in [23] the authors propose an incremental and neural-network-based version of the "spherical" k-means which, according to an appropriate rule for updating the weights of the neural network, incrementally modifies the closest cluster center, given a new document. In [1] the authors cluster blogs by considering their labels and generating a "collective wisdom". In [15], stories, built from blogs, are clustered. After a set of initial clusters is built, a dynamic clustering algorithm incrementally updates clusters on the basis of the distance between new stories and clusters' stories.

Despite the clear relationship, there are differences between these researches and ours. In the former, clusters represent the same topic across the time dimension

whereas we associate clusters to a single time interval. Consequently, we do not apply incremental clustering approaches, but we identify clusters for each time interval and compare them with those previously identified. Moreover, in topic tracking, clusters group documents on the same topic, i.e. the unit of analysis is the document, while in our case the unit of analysis is the entity. This means that we cluster entities on the basis of documents associated to them.

A similar approach to ours is proposed in [4], where clusters of keywords extracted from messages published in blogs are identified for each time interval. Clusters associated to consecutive time intervals are pairwise compared in order to identify pairs with the highest affinity. By combining affinity relationships over several time intervals, it is possible to identify the top-k paths that express the most significant evolutions of the initial clusters over time. The main difference with respect to our approach is that the considered unit of analysis is the "keyword", on the basis of the assumption that clusters of keywords characterize topics. Similarly, in [21] the authors propose an approach for defining and monitoring topics by clustering, for each time interval, blogs on the basis of their content. However, in this case, clustering is performed on the pairs $(class, similarity)$ obtained by a centroid-based classifier. This means that clustering significantly depends on a preliminary supervised learning phase.

In [18], the authors propose the identification of evolutions of clusters over time, by considering the application of either batch or incremental clustering approaches for each time interval. Evolutions are represented through an *Evolution Graph*, where nodes are clusters and edges connect clusters belonging to adjacent time intervals, and are summarized through the so-called *fingerprints*. Also in this case, the units of analysis are the keywords of textual documents.

Finally, we mention the work presented in [12] where the authors propose to learn, from news, a generative model of terms which takes as input the topic and the mentioned entities. Although this work does not exploit the time dimension, it considers, similarly to our work, the possible correlation of news with other entities such as people, organizations, locations.

3 TB-EEDIS

In this section we define the framework ***TB-EEDIS*** *(Time-Based Entity Evolution DIScoverer)* that allows us to discover *evolutions* of entities from a stream of textual documents. In this respect, an *evolution* is defined as a relevant change of entities' properties in different time windows. All the necessary information are extracted from time-stamped textual documents which are implicitly associated to a single time window. In this work, *time windows* are defined as adjacent and disjunct time intervals, obtained by partitioning the entire period we intend to analyze into intervals of the same size. Evolutions are discovered by analyzing the changes identified among distinct time windows.

The textual content of each document d_j is represented according to the classical *Vector Space Model (VSM) with TF-IDF weighting.* Each entity is represented in the same space of terms used for representing documents (an example

for news about criminals, that we will show in the experiments, is: [attack; fire; claim; suspect; report; injur; islam]), which better represents the profile of the entity in a given time window. Since the terms space can be very large, we select relevant features through an unsupervised feature selection algorithm.

Summarizing, the framework *TB-EEDIS* consists of the following phases: *i)* identification of entities from each document; *ii) VSM* representation of the documents (after applying classical pre-processing techniques), *iii)* feature selection, *iv)* identification of the position of each entity for each time window, *v)* clustering of entities for each time window and *vi)* evolution discovery and analysis. Entity identification is performed by applying two natural language processing techniques, that is, Named Entity Extraction [19] and Dependencies Analysis [16]. The adopted strategy considers the logical structure of the sentences and, starting from relationships, it identifies the involved entities. Since this task is not the main subject of the paper, for space constraints, we do not report further details about this phase. In the following, we explain the methods we use for selecting relevant features, representing entities and studying their evolution.

3.1 Feature Selection

We present two distinct unsupervised feature selection algorithms. The use of unsupervised approaches is motivated by the task we consider (i.e. clustering) and the consequent absence of any target (class) attribute to guide the selection.

Variance-based Feature Selection. The most straightforward way to perform feature selection is by computing the variance of the relative term-frequency of each term in the entire document collection and keeping the k terms with the highest variance. Intuitively, a term with high variance will better discriminate documents, whereas a term with low or zero variance will substantially describe the documents in the same way. This feature selection algorithm has the advantage of a linear time complexity, at the price of some disadvantages: *a)* It selects the terms which best discriminate between documents, disregarding their real similarity. In fact, it does not take into account the case in which similar documents share the same terms with similar relative term-frequency. This can lead to lose terms that characterize entire classes of documents. *b)* It does not consider the correlation between terms. In fact, two strongly correlated terms will be both selected if they are in the set of the top k terms with the highest variance. This leads to select redundant terms.

MIGRAL-CP. In [9], the authors propose to use the Laplacian Score to identify the features which better preserve samples similarity. However, the Laplacian Score rewards features for which similar samples show a small variation in the feature values, but does not reward those that show a large variation for dissimilar samples. Inspired by this work, we define a different method called *MInimum GRAph Loss with Correlation Penalty (MIGRAL-CP)*, which *i)* selects k terms to represent the whole collection of documents, showing both great variation for

dissimilar documents and low variation for similar ones and *ii)* discards features correlated with already selected features.

Formally, given the set of documents $D = \{d_1, d_2, \ldots, d_n\}$ and the set of terms $T = \{t_1, t_2, \ldots, t_m\}$, we build the (fully-connected undirected) graph G, where each node represents a document and each edge between two documents d_i and d_j is labeled with their similarity computed as: $v_{i,j} = e^{-||w_{d_i} - w_{d_j}||^2}$, where w_{d_i} (w_{d_j}) is the relative term-frequency vector associated to the document d_i (d_j), defined according to the set of terms T. The similarity measure we use is defined in [9] but, obviously, any other similarity measure might be considered.

We define an iterative method to select a subset of k terms which satisfy the above requirements. The first term is selected in order to maximize the score:

$$Score_1(t_r) = \frac{1}{2}\Big(1 - \frac{1}{n}\sum_{j=1}^{n} \rho(V_j, F_{r,j})\Big) \tag{1}$$

where:

- $V_j = [v_{j,1}, v_{j,2}, \ldots, v_{j,n}]$ are the similarity values between the document j and all the other documents, using all the terms.
- $F_{r,j} = \big[(s_{r,j} - s_{r,1})^2, (s_{r,j} - s_{r,2})^2, \ldots, (s_{r,j} - s_{r,n})^2\big]$ are the dissimilarities between the document j and all the other documents, using the term t_r only. In this formula, $s_{r,j}$ is the relative term frequency of the term t_r in d_j.
- $\rho(\cdot, \cdot)$ is the Pearson correlation coefficient.

The first selected term ($\hat{t}_1 = \arg\max_{t_r \in T} Score_1(t_r)$) will be the one which determines the highest inverse linear correlation between the documents' similarities computed with all the terms and documents' dissimilarities computed with only that term. The remaining $k - 1$ terms are selected according to:

$$Score_i(t_r) = Score_{i-1}(t_r) \times (1 - penalty(t_r, \hat{t}_{i-1})) \tag{2}$$

where, at the iteration i, $penalty(t_r, \hat{t}_{i-1})$ reduces the score of each term t_r according to its correlation to the term selected during the previous iteration, preventing the selection of redundant terms. Coherently with the correlation coefficient introduced before, we define $penalty(t_r, \hat{t}_{i-1}) = \max\left(0, \left|\rho(t_r, \hat{t}_{i-1})\right| - \gamma\right)$, where $0 \leq \gamma \leq 1$. The rationale of this choice is that a correlation value of $\left|\rho(t_r, \hat{t}_{i-1})\right| \leq \gamma$ is considered too small to result in a penalty.

3.2 Representing Entities

For each time window τ_z, the profile of each entity is represented in the same k-dimensional terms space identified in the feature selection phase. In the following we describe two possible alternatives.

Time-Weighted Centroid. In this case, the profile of the entity c in τ_z is:

$$X(c, \tau_z, h) = \frac{\sum_{<d_j, \tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h) \times w_{d_j}}{\sum_{<d_j, \tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h)}, \tag{3}$$

where $S_{c,\tau_z,h}$ is the set of documents associated to c and belonging to τ_z or to one of the previous $h-1$ time windows, and $p_{\tau_z,\tau_j}(h) = 1 - \frac{z-j+1}{h}$ is the time fading-factor which reduces the effect of the document d_j according to the distance between τ_z and the time window τ_j (i.e. the time window of d_j).

Max Density Point. In this solution, inspired to the work in [6], each document d_j is represented as a k-dimensional Gaussian function $d'_j(\cdot)$: $[0,1]^k \rightarrow \mathbb{R}^+$:

$$d'_j(x) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - s_{i,j})^2}{2\sigma^2}} \tag{4}$$

where $\sigma \in [0,1]$ is a parameter that defines the width of the Gaussian function.

The position of c in τ_z is the point with the highest sum of contributions:

$$X(c,\tau_z,h) = \arg\max_{x \in [0,1]^k} \sum_{<d_j,\tau_j> \in S_{c,\tau_z,h}} p_{\tau_z,\tau_j}(h) \times d'_j(x) \tag{5}$$

where the time fading-factor $p_{\tau_z,\tau_j}(h)$ reduces the value of the Gaussian function.

For computational reasons, we search $X(c,\tau_z,h)$ in the discrete space Φ^k, where $\Phi = \left\{0, \frac{1}{\beta}, \ldots, \frac{\beta-1}{\beta}, 1\right\}$ and $\beta+1$ is the number of desired distinct values.

Moreover, we adopt two further optimizations: *i)* we limit the search to the areas interested by at least one document belonging to the time window τ_z, and to the position $X(c,\tau_{z-1},h)$, assumed in the previous time window (*incrementality*); *ii)* we adopt a greedy search that works only around the points for which the $d'_j(\cdot)$ functions, contributing to $X(c,\tau_z,h)$, reach the highest values. In particular, we focus (for each dimension) on a smaller area around the point for which $d'_j(\cdot)$ assumes the maximum value. Formally, let y be the value assumed on a given dimension by a document. Given the applied discretization, we analyze only $\beta\sigma$ values on both sides of y, leading to a total of $2\beta\sigma+1$ values[1], instead of $\beta+1$, thus covering all the available values in $[y-\sigma; y+\sigma]$ (see Figure 1).

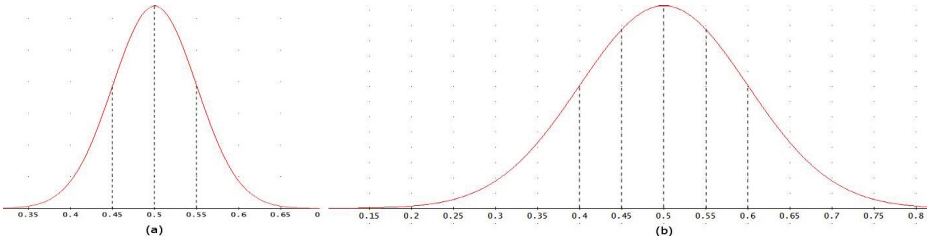

Fig. 1. Gaussian function defined on a single dimension with $y = 0.5$, $\beta = 20$, $\sigma = 0.05$ (a) and $\sigma = 0.10$ (b). In (a) it is enough to analyze only the values 0.45, 0.50 and 0.55, whereas in (b) it is necessary to analyze also the values 0.40 and 0.60.

[1] Reasonable values of σ are ≤ 0.1. In the experiments we use $\sigma \in \{0.05, 0.1\}$.

3.3 Clusters Evolution Discovery

The last necessary step, before analyzing the evolutions of entities, consists of searching for clusters of entities for each time window. Although we perform clustering for each time window independently, i.e. without considering the temporal neighborhood, it is noteworthy that the influence of documents belonging to previous time windows is caught by the proposed strategy for the identification of entities' profile, as already described in Section 3.2.

We use a variant of the K-means algorithm. Obviously, in TB-EEDIS, any clustering algorithm (also density based, e.g. DBSCAN [8]) can be plugged in.

Our improvement to the standard K-means algorithm consists in the automatic identification of the reasonable number of clusters to be extracted, which is necessary in the task at hand, since the number of clusters is not known a-apriori. The solution we adopt is that of exploiting the *Principal Component Analysis (PCA)*, which identifies a new (smaller) set of prototype features such that a given percentage of the variance in the data is explained [11]. By inverting the roles of features and examples, it is possible to identify a set of (orthogonal) prototype examples, according to which other examples can be aggregated. In our solution we use the number of prototype examples as an indication of the appropriate number of clusters, according to the underlying data distribution.

Once clustering is performed for each time window, it is possible to identify:

- the *position* of the cluster in the k-dimensional terms space. This can be identified by analyzing the terms with the highest weights in the cluster (e.g. of its centroid), which gives an idea about the entity typology it represents.
- a *matching* between clusters of different time windows by maximizing the similarity between the centroids of matched clusters, which are still represented in the same terms space.
- the *number of entities* which have evolved from the entity typology represented by C_i to that represented by C_j, where C_i and C_j are two generic clusters extracted for the time windows τ_{z-1} and τ_z, respectively (Figure 2).

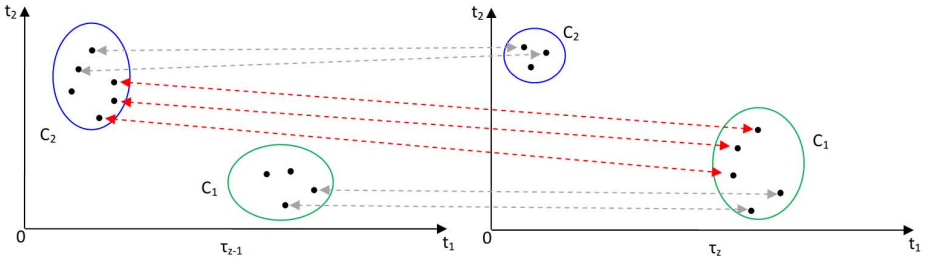

Fig. 2. An evolution: Three entities moved from C_2 in τ_{z-1} to C_1 in τ_z

4 Empirical Evaluation

The evaluation is performed on a set of synthetically generated datasets and on a real dataset. The synthetic datasets consist of documents about 50 entities, generated for 30 time windows. For each time window and entity, a set of 10 documents is generated by considering 7 specific vocabularies (representing different topics) and a generic English vocabulary, used to introduce "noise terms". For each time window and entity, the probability of changing the topic is set to 0.2. Three different datasets are generated, setting the number of time windows necessary to complete each evolution (which defines the "speed" of changes) to 4, 10 and 20. In order to reproduce realistic situations, when simulating an evolution from a topic A to a topic B, we gradually decrease the frequency of terms representing A and increase the frequency of terms representing B in the generated documents. During an evolution, no additional evolutions can start.

As real dataset, we consider the Global Terrorism Database (GTD)[2] for the risk identification and analysis domain. GTD consists of information on terrorist events (more than 104,000 cases) around the world from 1970 through 2011, including systematic data (such as the textual content of a news, involved criminals, publication date) on domestic as well as international terrorist incidents.

In the evaluation performed on the synthetic datasets, the feature selection is executed with both the proposed methods (variance-based, which is considered as baseline, and MIGRAL-CP), with $k = 10$ (number of features to select). For MIGRAL-CP, results are obtained with $\gamma = 0.5$ (γ is the minimum threshold on the Pearson coefficient for applying a penalty), which after preliminary experiments (not reported in this paper for space constraints), resulted in the best trade off between relevancy and allowed redundancy of selected terms. For MaxDensity, the discretization parameter β is set to 20 and σ of the Gaussian functions $d'_j(\cdot)$ is set to 0.05[3].

As regards GTD, we consider 13 annual time windows (from 1998 to 2010), for a total of 11,225 news concerning 82 criminals/criminal organizations. The feature selection is executed with both the proposed methods, with $k = 15$ and, for MIGRAL-CP, $\gamma = 0.5$. For MaxDensity, β is set to 20 and σ is set to 0.05[4]3.

Both synthetic and real datasets are analyzed with two different values for the PCA variance (90% and 95%) and with different values of h (2, 5 and 10). In particular, each synthetic dataset is analyzed with the corresponding value of h such that the number of time windows used to perform an evolution is $2h$. This solution is motivated by the fact that, in general, the system should be able to detect the change of the topic in the middle of the evolution. On the other hand, GTD is analyzed with all the considered values of h, since we do not know a priori the speed of evolutions in the dataset.

Results are collected in terms of running times (hh:mm) required to complete the whole evolution discovery process and in terms of a variant of the

[2] `http://www.start.umd.edu/gtd/`

[3] We also performed experiments with $\sigma = 0.10$. Since there was no significant difference in the results, for space constraints, we report only results with $\sigma = 0.05$.

Table 1. Results obtained on the synthetic datasets and on the GTD dataset. Italic indicates a better result with respect to the strategy for computing the entity position, while bold indicates a better result with respect to the feature selection strategy.

			Synthetic Dataset						**GTD Dataset**			
			Variance			**MIGRAL-CP**			**Variance**		**MIGRAL-CP**	
h	**Position**	**Var**	**time**	**q-mod**	**NMI**	**time**	**q-mod**	**NMI**	**time**	**q-mod**	**time**	**q-mod**
2	Centroid	90%	07:38	*0.581*	0.652	15:21	**0.613**	**0.757**	00:09	**0.294**	39:54	0.245
2	Centroid	95%	07:39	*0.606*	0.692	15:22	**0.659**	**0.799**	00:09	**0.319**	39:54	0.270
2	MaxDensity	90%	07:43	0.570	*0.689*	15:26	***0.672***	***0.770***	110:41	*0.322*	100:17	***0.447***
2	MaxDensity	95%	07:43	0.577	*0.698*	15:27	***0.710***	***0.800***	110:41	***0.509***	100:17	*0.479*
5	Centroid	90%	08:06	0.559	0.615	21:03	**0.616**	**0.726**	00:09	**0.297**	39:54	0.224
5	Centroid	95%	08:07	0.633	0.678	21:04	**0.634**	**0.739**	00:09	**0.316**	39:54	0.249
5	MaxDensity	90%	08:16	*0.614*	*0.679*	21:11	***0.654***	***0.773***	137:41	*0.325*	118:59	***0.454***
5	MaxDensity	95%	08:17	*0.665*	*0.713*	21:12	***0.690***	***0.797***	137:41	*0.521*	118:59	***0.487***
10	Centroid	90%	08:31	*0.534*	0.548	25:51	**0.547**	**0.698**	00:09	**0.304**	39:54	0.232
10	Centroid	95%	08:32	*0.565*	0.575	25:52	**0.587**	**0.731**	00:09	**0.322**	39:54	0.245
10	MaxDensity	90%	08:45	0.491	*0.603*	26:08	***0.607***	***0.724***	144:20	*0.400*	126:06	***0.452***
10	MaxDensity	95%	08:46	0.522	*0.623*	26:09	***0.660***	***0.762***	144:20	***0.524***	126:06	*0.479*

Q-Modularity measure [17], which allows us to evaluate the quality of the clustering with respect to a random clustering. This variant is described in the following. Let $e_{ij} = \frac{2}{r(r-1)} \sum_{c' \in C_i, c'' \in C_j} sim(X(c', \tau_z, h), X(c'', \tau_z, h))$ be a measure of the strength of the interconnections between entities in the cluster C_i and entities in the cluster C_j. In this formula, r represents the total number of entities and $sim(\cdot, \cdot) \in [0, 1]$ is the cosine similarity. Intuitively, we want clusters for which e_{ii} values are generally large and $e_{ij} (i \neq j)$ values are generally small. Formally: $Q = \sum_{i=1}^{k} \left(e_{ii} - a_i^2\right)$, where $a_i = \sum_j e_{ij} = \sum_j e_{ji}$.

Moreover, for the synthetic datasets, we also evaluate the results in terms of the average Normalized Mutual Information (NMI) [20]. In particular, NMI is computed between the set of extracted clusters and the set of true clusters representing topics imposed during the generation of the datasets, in order evaluate the ability of TB-EEDIS to correctly catch the underlying evolutions.

As it can be observed in Table 1, for feature selection, the MIGRAL-CP algorithm always leads to better Q-Modularity and NMI results in the synthetic datasets, with respect to the variance-based method (which we consider as a baseline). The disadvantage is that better results are obtained at the price of significantly higher running times. These observations do not hold for the GTD dataset, where there is no clear advantage of MIGRAL-CP over the variance-based method in the case of MaxDensity (where we have better results). The motivation can be found in the fact that in the synthetic datasets we explicitly introduced redundancy in the text, while in GTD this phenomenon is not under control and the two methods almost equally perform.

As regards the method for computing the entities' position, the MaxDensity method always significantly outperforms the centroid-based method (which we consider as baseline) on GTD, at the price of a slightly higher running times, whereas on the synthetic datasets it shows betters results only in terms of NMI.

Observing the influence of the variance (for the PCA-based estimation of the number of clusters), we have that, for $Var = 95\%$, extracted clusters better adapt to the underlying topics models (see NMI in Table 1), without incurring

in overfitting issues. This phenomenon is reflected on Q-Modularity values, also for GTD. However, it is noteworthy that the quality of results is less dependent on such parameter when the MIGRAL-CP feature selection method is adopted.

From a qualitative viewpoint, it is interesting to identify a description of the clusters, in order to deeply understand the evolutions in which they are involved. A possibility consists in the analysis of the terms describing the entities belonging to the cluster. For example, analyzing the centroid of a cluster identified from GTD (Var=90%, MaxDensity-MIGRAL-CP, $h = 5$), that is: [attack: 0.593; fire: 0.371; claim: 0.271; suspect: 0.1; report: 0.057; injur: 0.057; islam: 0.05] allows us to identify a specific type of crime (terrorist attack). For future work, we will investigate the possibility of performing an extensive qualitative analysis of the evolutions discovered from real datasets.

5 Conclusions

In this paper, we propose the framework TB-EEDIS to incrementally extract knowledge from time-stamped documents. In particular, it: identifies entities with domain-specific roles; represents documents by exploiting unsupervised feature selection algorithms; represents the entities' profile and identifies clusters of entities in order to represent and analyze their evolution.

Results show that the algorithms proposed for the unsupervised feature selection (MIGRAL-CP) and for the identification of the position of entities (Max Density-based) generally provide better results when compared to baseline approaches. Moreover, results obtained in terms of Normalized Mutual Information on synthetic datasets prove the ability of TB-EEDIS to catch the underlying evolutions of entities, making it applicable in additional domains (e.g. biological).

For future work, we intend to analytically identify the value of σ, with respect to h, such that the global optimum is guaranteed. Moreover, we will qualitatively evaluate the evolutions discovered on real datasets and we will analyze how different sizes of time windows can influence results.

Acknowledgements. We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References

1. Agarwal, N., Galan, M., Liu, H., Subramanya, S.: Wiscoll: Collective wisdom based blog clustering. Inf. Sci. 180, 39–61 (2010)
2. Aggarwal, C.C.: On change diagnosis in evolving data streams. IEEE Trans. Knowl. Data Eng. 17(5), 587–600 (2005)
3. Allan, J. (ed.): Topic Detection and Tracking: Event-based Information Organization. Kluwer International Series on Information Retrieval, Kluwer (2002)
4. Bansal, N., Chiang, F., Koudas, N., Tompa, F.W.: Seeking stable clusters in the blogosphere. In: VLDB, pp. 806–817. ACM (2007)

5. Brants, T., Chen, F., Farahat, A.: A system for new event detection. In: ACM SIGIR, pp. 330–337. SIGIR 2003. ACM (2003)
6. Ceci, M., Appice, A., Malerba, D.: Time-slice density estimation for semantic-based tourist destination suggestion. In: ECAI (2010)
7. Chung, S., McLeod, D.: Dynamic pattern mining: An incremental data clustering approach. In: Spaccapietra, S., Bertino, E., Jajodia, S., King, R., McLeod, D., Orlowska, M.E., Strous, L. (eds.) Journal on Data Semantics II. LNCS, vol. 3360, pp. 85–112. Springer, Heidelberg (2005)
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: ICDM, pp. 226–231 (1996)
9. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: NIPS (2005)
10. Jameel, S., Lam, W.: An n-gram topic model for time-stamped documents. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 292–304. Springer, Heidelberg (2013)
11. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer (2002)
12. Kim, H., Sun, Y., Hockenmaier, J., Han, J.: Etm: Entity topic models for mining documents associated with entities. In: ICDM, pp. 349–358. IEEE (2012)
13. Kleinberg, J.: Bursty and hierarchical structure in streams. In: ACM SIGKDD, KDD 2002, pp. 91–101. ACM, New York (2002)
14. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: KDD 2009, pp. 497–506. ACM, New York (2009)
15. Li, X., Yan, J., Fan, W., Liu, N., Yan, S., Chen, Z.: An online blog reading system by topic clustering and personalized ranking. ACM Trans. Internet Technol. 9, 9:1–9:26 (2009)
16. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure trees. In: LREC (2006)
17. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103(23), 8577–8582 (2006)
18. Ntoutsi, E., Spiliopoulou, M., Theodoridis, Y.: Fingerprint: Summarizing cluster evolution in dynamic environments. IJDWM 8(3), 27–44 (2012)
19. Sarawagi, S.: Information extraction. Foundations and Trends in Databases 1(3), 261–377 (2008)
20. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617 (2003)
21. Varlamis, I., Vassalos, V., Palaios, A.: Monitoring the evolution of interests in the blogosphere. In: ICDEW, pp. 513–518 (2008)
22. Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B., Liu, X.: Learning approaches for detecting and tracking news events. IEEE Intelligent Systems and their Applications 14(4), 32–43 (1999)
23. Zhong, S.: Efficient streaming text clustering. Neural Networks 18(5-6) (2005)
24. Zhu, Y., Shasha, D.: Efficient elastic burst detection in data streams. In: ACM SIGKDD, KDD 2003, pp. 336–345. ACM, New York (2003)