TRACES: Generating Twitter Stories via Shared Subspace and Temporal Smoothness

Xuchao Zhang[1], Zhiqian Chen[1], Liang Zhao[2], Arnold P. Boedihardjo[3], Chang-Tien Lu[1]
[1]Virginia Tech, Falls Church, VA, USA
[2]George Mason University, Fairfax, VA, USA
[3]U. S. Army Corps of Engineers, Alexandria, VA, USA
[1]{xuczhang, czq, liangz8, ctlu}@vt.edu, [2]zhao9@gmu.edu, [3]arnold.p.boedihardjo@usace.army.mil

***Abstract*—In the era of information overload, people are struggling to make sense of complex story events in massive social media data. Most existing approaches are designed to address event extraction in news reports, documents and abstracts, but such approaches are not suitable for Twitter data streams due to their unstructured language, short-length messages, and heterogeneous features; few existing approach generates a story by considering both the shared topics throughout the story and the smooth connection between successive nodes simultaneously. In this paper, a novel Twitter stoRy generation framework via shAred subspaCe and tEmporal Smoothness called TRACES is proposed. Given a query of an ongoing event, a novel multi-task clustering method integrated with shared subspace and temporal smoothness (STMTC) is proposed to generate the event stories. Extensive experimental evaluations of data sets for different events demonstrate the effectiveness of this new approach.**

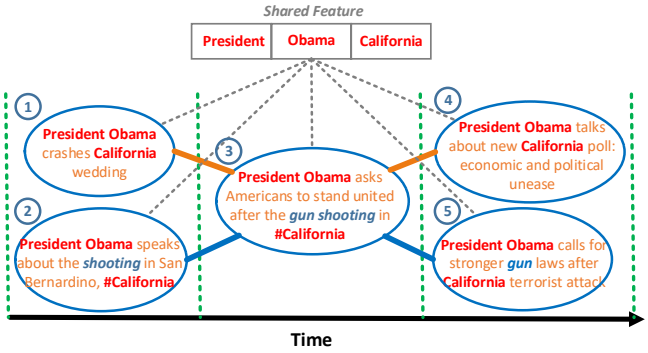

Figure 1. Stories of president Obama with state California.

I. Introduction

Social media such as Twitter is rapidly becoming a real-time "news press" for spreading information at both a global and local community scales. Hundreds of millions of users post tweets every minute, discussing everything from their opinions about world events to incidents they observe on the street. Compared to traditional media, people are attracted to microblog sites such as Twitter because they provide instant first-hand reports on real-life events. Every day, millions of Twitter users around the world broadcast their observations and comments on a variety of topics such as crime, sports, and politics. In contrast to the censorship often imposed on traditional media, tweets can more freely express idiosyncratic views and inconvenient facts. It is helpful for industry, academia, and end-users if a story event can be automatically generated from this huge volume of tweets. For example, story lines related to the November 2015 Paris Attack in its first four days are shown in Figure 2. The horizontal location of each node indicates the time sequence of the events as they unfolded. The story provides a basic overview of the main events, such as the attack in Bataclan theater, the search for accomplices, and the international aid provided by the European Union and others as well as spreading useful information via Twitter, such as helplines for victims.

Ways to generate event stories have been well studied [1] [2][3], but most of these methods assume that the textual contents are robust and well presented, which is not always the case for social media. Furthermore, few existing method consider the two properties of event story simultaneously: 1) a common topic shared throughout all the nodes in a story, and 2) smooth connection between the consecutive nodes. For example, the story ② → ③ → ⑤ in Figure 1 considers both shared features: "*President*", "*Obama*", "*California*", and transition terms: "*shooting*" and "*gun*" between ② → ③ and ③ → ⑤. It is a more compelling result than story ① → ③ → ④ considering shared topic only, in which the three nodes refer to different events: *California wedding*, *California shooting* and *California poll*. Thus, the challenges facing microblog story generation arise from the consideration of both shared features and temporal smoothness. The example shown in Figure 1 demonstrates that the two properties are key factors to generate a compelling story.

In this paper, we focus on resolving the above challenges. The major contributions of this research are summarized as follows: 1) *Developing a system to generate event stories in Twitter.* A novel unsupervised approach is proposed for event story generation in Twitter. Our method extracts related tweets and features for a given user query, and connects related events as a story line by considering both the shared features throughout the story and the smoothness between successive events. 2) *Proposing an innovative story generation method with a multi-task clustering algorithm.* Based on the extracted tweet features from dynamic query expansion, a multi-task clustering algorithm that jointly

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

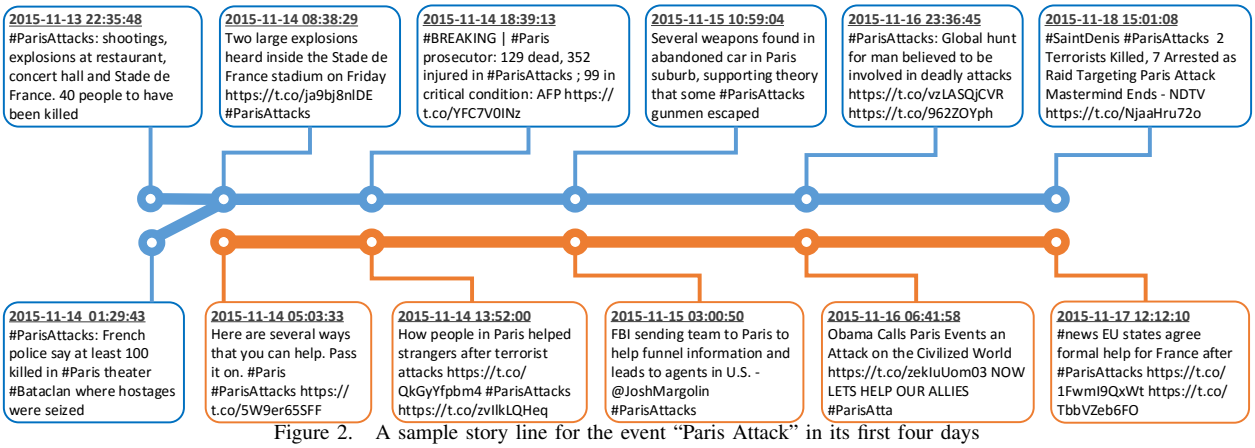

Figure 2. A sample story line for the event "Paris Attack" in its first four days

maximizes the shared subspace and temporal smoothness (STMTC) is proposed to distinguish stories by taking into account both their global shared topics and the temporal smoothness between consecutive events. 3) *Conducting extensive experimental performance evaluations.* Our method has been extensively evaluated on Twitter data covering more than 4 countries in Latin America, with multiple topics and languages. Comparisons with baselines and state-of-the-art methods have demonstrated its effectiveness.

The remainder of this paper is organized as follows. Section II describes the related work on story construction and multi-task clustering. Section III presents the multi-task clustering model with shared subspace and temporal smoothness. In Section IV, experimental results are analyzed and a case study is presented. We conclude our work in Section V.

II. Related Work

Several research directions are related to our work, including storyline construction, and multi-task clustering.

A. Timeline and Storyline Construction

Only a limited number of studies have looked at document summarization with time stamps, with most focusing on news articles. Mei et al. [4] proposed an HMM style probabilistic method to discover and summarize the evolutionary patterns of themes in text streams, while Lappas et al. [5] defined a term burstness model to discover the temporal trend of terms in news article streams. Wang et al. [6] took this further, developing an evolutionary document summarization system to generate an evolution skeleton along the timeline. The Evolutionary Timeline Summarization (ETS) [7] method has also been proposed to return an evolution trajectory along the timeline by emphasizing a theme's relevance, coverage, coherence and diversity. However it is difficult to apply these timeline generation methods to Twitter datasets due to their heterogeneous features, as most only consider the evolution between time periods rather than treating a story as having integrated shared features.

Some researchers have focused on storyline generation. For example, Lin et al. [3] proposed a language model with dynamic pseudo relevance feedback to obtain relevant tweets, and then generated story lines via graph optimization. The approach assumes that only one story exists in the tweets extracted from a user query, however, and their Steiner Tree algorithm does not consider the shared features in the story. Shahaf et al. [2] proposed a metro-map format story generation framework with three separate steps: BigClam [8], which is a community detection method to detect event clusters in each time period; the NMF method, which groups communities related to the story; and a sub-modularity function to optimize the story. The relationship between communities in different time periods is not considered integrally in the clustering phrase, which is detrimental to the clustering result, as demonstrated below in Section IV.

B. Multi-Task Clustering

While multi-task learning methods tracking classification [9][10] have received a lot of attention in recent years, a relatively few studies have been devoted to multi-task learning for the clustering problem. Gu et al. [11] performed multiple related clustering tasks together and utilized the relationships between these tasks in terms of their shared subspace. Later, the same group went on to a method to learn nonparametric and spectral kernel [12] for multi-task clustering. Zhang et al. [13] designed a method based on general Bregman divergences and defined two task regularizations to encourage coherence among tasks. A domain adaptation approach has also been proposed to perform multiple related clustering tasks [14]. However, although most of these methods integrate tasks with a shared subspace, but none consider the temporal smoothness between consecutive tasks, which is important when finding coherent events in story construction.

III. Proposed Model

In this section, a multi-task clustering method integrated with shared subspace and temporal smoothness is proposed

to generate story lines. We will begin by introducing the objective of clustering, after which the optimization of multi-task clustering objective is presented. Finally, a Kullback-Leibler divergence based method is used to connect clusters.

A. Clustering Objective

Given a set of relevant tweets $\mathcal{T}_p$ which is divided into m time steps $\{\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(m)}\}$, clustering task is defined as tweet clustering in one time step. Considering the case of single-task clustering, our purpose is to partition the k-th data set into c clusters. The semi-NMF clustering [15] algorithm achieves this goal by minimizing the following objective:

$$\begin{aligned} J_{st} &= \|\mathbf{X}^{(k)} - \mathbf{U}^{(k)}[\mathbf{P}^{(k)}]^T\|_F^2 \\ s.t.\ \ &\mathbf{P}^{(k)} \geq 0 \end{aligned} \tag{1}$$

where $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}]$, $1 \leq k \leq m$. $\mathbf{X}^{(k)}$ represents the feature and tweet relationship matrix extracted from dynamic query expansion method [16], and m is the number of tasks. $\|\cdot\|_F$ is the Frobenius norm, $\mathbf{U}^{(k)} \in \mathbb{R}^{d\times c}$ represents the feature assignment in clusters, $\mathbf{P}^{(k)} \in \mathbb{R}^{n_k\times c}$ is the partition matrix that represents the clustering assignment, d is the number of features, c is the number of clusters, and n_k is the number of tweets in task k. Unlike hard clustering methods such as k-means, a tweet can be assigned to different clusters when two topics are discussed in the same tweet.

When it comes to multi-task clustering setting, we introduce both shared subspace [11] and temporal smoothness [17] to represent the coherence between tasks. The shared subspace is obtained by an orthonormal projection of features across all the related tasks and represents the terms, hashtags, users, hyperlinks and mentions shared in each story.

Our shared subspace and temporal smoothness multi-task clustering (STMTC) method is formulated by minimizing the following objective function:

$$\begin{aligned} J_{mt} = &\sum_{k=1}^{m}\|\mathbf{X}^{(k)} - \mathbf{U}\mathbf{G}^{(k)}[\mathbf{P}^{(k)}]^T\|_F^2 + \lambda\sum_{k=1}^{m}\|\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}[\mathbf{P}^{(k)}]^T\|_F^2 \\ &+ \theta_1\sum_{k=1}^{m}\|\mathbf{U}\mathbf{H}^{(k)}\|_F^2 + \theta_2\|\mathbf{U}\|_F^2 \qquad s.t.\ \ \mathbf{W}^T\mathbf{W} = \mathbf{I} \quad \mathbf{P}^{(k)} \geq 0 \end{aligned} \tag{2}$$

where $\lambda, \theta_1, \theta_2 \in [0, 1]$ are regularization parameters that balance any clustering in the within-task input space, shared subspace and temporal smoothness. $\mathbf{U} = [\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(k)}]$ refers to the feature assignment in the clusters of all the tasks. $\mathbf{G}^{(k)} \in \mathbb{R}^{mc\times c}$ is defined as follows:

$$\mathbf{G}_{ij}^{(k)} = \begin{cases} 1, & \text{if } i = c(k-1) + j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$\mathbf{U}\mathbf{G}^{(k)}$ represents the feature assignment of clusters in task k. As with single-task clustering, $\mathbf{P}^{(k)} \in \mathbb{R}^{n_k\times c}$ is the partition matrix in task k. $\mathbf{W} \in \mathbb{R}^{d\times l}$ is the orthogonal projection of features into the shared subspace, in which l is the feature number in that subspace. $\mathbf{M} \in \mathbb{R}^{l\times c}$ is the subspace feature partition of the clusters in all the tasks. $\mathbf{H}^{(k)} \in \mathbb{R}^{mc\times 1}$ is defined as follows:

$$\mathbf{H}_i^{(k)} = \begin{cases} -1, & \text{if } (k-2)c + 1 \leq i \leq c(k-1) \\ 1, & \text{if } (k-1)c + 1 \leq i \leq kc \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Specifically, $\mathbf{H}^{(1)}$ is a zero matrix. The variance between one task and its neighbor can be represented as $\mathbf{U}\mathbf{H}^{(k)}$.

The objective in Eq (2) consists of three terms. The first term addresses within-task clustering, which contains m independent clustering tasks in the Twitter input space. The second term deals with shared subspace clustering, which not only learns the shared features but clusters the data for all the tasks together in the shared subspace. The third term consists of both the temporal smoothness $\mathbf{U}\mathbf{H}^{(k)}$ of neighboring tasks and the Frobenius norm of $\mathbf{U}$ for its sparsity.

B. Multi-Task Clustering Optimization

Observe that minimizing Eq (2) is performed with respect to $\mathbf{U}$, $\mathbf{P}^{(k)}$, $\mathbf{W}$ and $\mathbf{M}$. An alternating minimization algorithm is proposed to optimize the objective, where the objective is optimized with respect to one variable when fixing the others. As the space limitation, the computation of $\mathbf{U}$ and $\mathbf{M}$ can be found in the supplementary document [1].

1) Computation of $\mathbf{P}^{(k)}$: Given $\mathbf{U}, \mathbf{M}, \mathbf{W}$, optimizing Eq (2) with respect to $\mathbf{P}^{(k)}$ is equivalent to optimizing

$$\begin{aligned} J_{P^{(k)}} = &\|\mathbf{X}^{(k)} - \mathbf{U}\mathbf{G}^{(k)}[\mathbf{P}^{(k)}]^T\|_F^2 \\ &+ \lambda\|\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}[\mathbf{P}^{(k)}]^T\|_F^2 \\ s.t.\ \ &\mathbf{P}^{(k)} \geq 0 \end{aligned} \tag{5}$$

For the constraint $\mathbf{P}^{(k)} \geq 0$, an iterative solution will be presented. We introduce the Lagrangian multiplier $\gamma \in \mathbb{R}^{n_k\times c}$ into the equivalent objective function $J_{P^{(k)}}$:

$$\begin{aligned} L(\mathbf{P}^{(k)}) = &\|\mathbf{X}^{(k)} - \mathbf{U}\mathbf{G}^{(k)}[\mathbf{P}^{(k)}]^T\|_F^2 \\ &+ \lambda\|\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}[\mathbf{P}^{(k)}]^T\|_F^2 - \gamma{\mathbf{P}^{(k)}}^T \end{aligned} \tag{6}$$

Setting $\frac{\partial L(P^{(k)})}{\partial P^{(k)}} = 0$, we obtain

$$\gamma = -2\mathbf{A} + 2\mathbf{P}^{(k)}\mathbf{B} \tag{7}$$

where

$$\begin{aligned} \mathbf{A} &= [\mathbf{X}^{(k)}]^T\mathbf{U}\mathbf{G}^{(k)} + \lambda[\mathbf{X}^{(k)}]^T\mathbf{W}\mathbf{M} \\ \mathbf{B} &= [\mathbf{G}^{(k)}]^T\mathbf{U}^T\mathbf{U}\mathbf{G}^{(k)} + \lambda\mathbf{W}^T\mathbf{M} \end{aligned} \tag{8}$$

With the Karush-Kuhn-Tucker condition [18] $\gamma_{ij}\mathbf{P}_{ij}^{(k)} = 0$, we get

$$[-2\mathbf{A} + 2\mathbf{P}^{(k)}\mathbf{B}]_{ij}\mathbf{P}_{ij}^{(k)} = 0 \tag{9}$$

As with the Semi-NMF algorithm in [15], we obtain the updating formula:

$$\mathbf{P}_{ij}^{(k)} \leftarrow \mathbf{P}_{ij}^{(k)}\sqrt{\frac{[\mathbf{A}^+ + \mathbf{P}^{(k)}\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{P}^{(k)}\mathbf{B}^+]_{ij}}}. \tag{10}$$

[1] https://goo.gl/hE7chW

Algorithm 1: LEARNING SHARED SUBSPACE AND TEMPORAL SMOOTHNESS FOR MULTI-TASK CLUSTERING(STMTC)

Input: m tasks, $\{X^{(k)}\}_{k=1}^{m}$, the dimensionality of the shared subspace l, maximum number of iterations T, convergence threshold δ;
Output: Cluster Partition $\mathbf{P}^{(k)} \in \mathbb{R}^{n_k \times c}$, 1 ≤ k ≤ m; Feature Cluster Partition $\mathbf{U}$

Initialize t = 1, $J_t = 0$
Initialize $\mathbf{P}^{(k)}$ with positive random values
Initialize $\mathbf{W} \in \mathbb{R}^{d \times l}$ with random orthonormal matrix
while $\Delta J > \delta$ ***and*** $t \leq T$ **do**
  $\mathbf{U} = (\sum_{k=1}^{m} \mathbf{X}^{(k)}\mathbf{P}^{(k)}[\mathbf{G}^{(k)}]^T)\mathbf{D}^{-1}$
  $\mathbf{M} = \mathbf{W}^T\mathbf{X}\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$
  for *k=1 to m* **do**
    Update $\mathbf{P}_{ij}^{(k)} \leftarrow \mathbf{P}_{ij}^{(k)} \sqrt{\frac{[\mathbf{A}^+ + \mathbf{P}^{(k)}\mathbf{B}^-]_{ij}}{[\mathbf{A}^- + \mathbf{P}^{(k)}\mathbf{B}^+]_{ij}}}$
  Compute $\mathbf{W}_{ij}$ by eigen-decomposition of $\mathbf{X}(\mathbf{I} - \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T)\mathbf{X}^T$
  $J_t = J(\mathbf{U}, \mathbf{M}, \mathbf{P}^{(k)}, \mathbf{W})$
  $\Delta J \leftarrow J_t - J_{t-1}$
  $t \leftarrow t + 1$

where $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$ and $\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^-$, in which $\mathbf{A}_{ij}^+ = (|\mathbf{A}_{ij}| + \mathbf{A}_{ij})/2$ and $\mathbf{A}_{ij}^- = (|\mathbf{A}_{ij}| - \mathbf{A}_{ij})/2$.

2) Computation of W: Given $\mathbf{U}$, $\mathbf{M}$, $\mathbf{P}^{(k)}$, optimizing Eq (2) with respect to W is equivalent to optimizing:

$$\begin{aligned} J_W &= \sum_{k=1}^{m} \|\mathbf{W}^T\mathbf{X}^{(k)} - \mathbf{M}[\mathbf{P}^{(k)}]^T\|_F^2 \\ &= \|\mathbf{W}^T\mathbf{X} - \mathbf{M}\mathbf{P}^T\|_F^2 \\ s.t.\ \ &\mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned} \tag{11}$$

where $\mathbf{X} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}] \in \mathbb{R}^{d \times n}$ and $\mathbf{P} = [[\mathbf{P}^{(1)}]^T, \ldots, [\mathbf{P}^{(m)}]^T] \in \mathbb{R}^{c \times n}$. After substituting $\mathbf{M} = \mathbf{W}^T\mathbf{X}\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}$, [11] shows that the optimal $\mathbf{W}$ minimizing Eq (11) is composed of the eigenvectors of $\mathbf{X}(\mathbf{I} - \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T)\mathbf{X}^T$.

We can now present the algorithm for optimizing Eq (2), namely Algorithm 1. The convergence of Algorithm 1 can be guaranteed by auxiliary function approach [19].

C. Cluster Connection

To find the connected clusters, we use Kullback-Leibler divergence [1] to measure the distance between two feature distributions in two successive clusters $C_m^{(t)}$ and $C_n^{(t+1)}$, where m, $n \in [0, c)$ are the index of clusters at times t and $t+1$, respectively, and c is the cluster number.

$$D_{KL}(C_m^{(t)} || C_n^{(t+1)}) = \sum_{d \in F} p(d|C_m^{(t)}) \ln \frac{p(d|C_m^{(t)})}{p(d|C_n^{(t+1)})} \tag{12}$$

where

$$\begin{aligned} p(d|C_m^{(t)}) &= \frac{\mathbf{U}_{d,(t \cdot c+m)}}{\sum_{d'} \mathbf{U}_{d',(t \cdot c+m)}} \\ p(d|C_n^{(t+1)}) &= \frac{\mathbf{U}_{d,((t+1) \cdot c+n)} + \epsilon}{\sum_{d'} (\mathbf{U}_{d',((t+1) \cdot c+n)} + \epsilon)} \end{aligned} \tag{13}$$

F is the feature set and $\mathbf{U}$ is the feature assignment matrix. ϵ is a small positive constant that is introduced to avoid zero values of $p(d|C_n^{(t+1)})$. For ease of comparison , we normalize the KL distance into intervals $[0, 1)$ using $D = 1 - e^{-D_{KL}}$. The cluster in t is connected with the maximum D value at the time $t+1$. The successive cluster nodes will be disconnected when $D > \tau$ and merged when $D < \phi$, where τ and ϕ are the decision thresholds.

IV. EXPERIMENT

This section presents the empirical evaluations of the performance of our proposed new approach, TRACES. By comparing the results with existing methods and baselines, the effectiveness of our method and its components are demonstrated.

A. Experiment Setup

1) Dataset and Labels: In order to evaluate both the STMTC clustering method and story, we chose both *LATAM* and Paris attack datasets purchased from Datasift Inc[2] and crawled via REST API, respectively. The LATAM dataset contains tweets gathered from civil unrest events in Latin America. It has totally 87,269 tweets including those related to Brazil's World Cup protest, Venezuelan protests, the Colombian presidential election and Chile's education march, covering events from June 2014 to August 2014. All these events are labeled as different types in the dataset for clustering evaluation purposes. The labels are collected from a SVM classifier trained by pre-labeled data and verified by human beings. The Paris attack dataset contains the tweets related to a series of coordinated terrorist attack that occurred in Paris on November 13th 2015, the deadliest attack in France since World War II. The dataset was crawled using the keyword "paris" for the period from November 13 to December 15 in 2015 for over ten different languages. After preprocessing, the resulting dataset contains 1.4 million tweets in English and French and is used to evaluate story quality in the case study in Section IV-B2.

2) Evaluation Metrics: The story quality is not only evaluated in terms of its clustering quality, but also qualitative story case studies. To evaluate the clustering results, we adopt the standard performance measures frequently used for clustering: 1) *Clustering Accuracy*: Clustering Accuracy [20] discovers the one-to-one relationship between clusters and labeled classes and measures the accuracy of clusters which contain data points from the corresponding class. Given a data point x_i, let r_i and s_i be the obtained cluster label and

[2]www.datasift.com

Table I
CLUSTERING RESULT FOR DATASET LATAM 1

	Task 1		Task 2		Task 3		Task 4	
	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
KM	0.440±0.000	0.613±0.017	0.680±0.007	0.513±0.001	0.696±0.006	0.471±0.003	0.492±0.003	0.425±0.010
LSAKM	0.574±0.000	0.351±0.000	0.528±0.000	0.632±0.000	0.769±0.000	0.520±0.000	0.369±0.000	0.357±0.000
ASI	0.523±0.004	0.346±0.017	0.479±0.006	0.393±0.001	0.590±0.017	0.464±0.050	0.655±0.002	0.431±0.012
BigClam	0.724±0.000	0.670±0.003	0.633±0.000	0.521±0.002	0.771±0.000	0.574±0.000	0.653±0.000	0.576±0.001
All KM	0.349±0.016	0.536±0.035	0.346±0.006	0.514±0.022	0.568±0.000	0.565±0.010	0.552±0.000	0.529±0.010
All LSAKM	0.440±0.000	0.382±0.000	0.402±0.000	0.351±0.000	0.367±0.000	0.346±0.000	0.370±0.000	0.362±0.000
All BigClam	0.644±0.000	**0.724±0.000**	0.596±0.000	0.684±0.000	0.575±0.000	0.569±0.000	0.634±0.000	0.615±0.000
LSSMTC	0.723±0.020	0.555±0.024	0.668±0.012	0.519±0.003	0.625±0.004	0.491±0.032	0.648±0.003	0.566±0.013
STMTC	**0.782±0.002**	0.675±0.000	**0.918±0.002**	**0.745±0.008**	**0.792±0.003**	**0.583±0.009**	**0.721±0.013**	**0.628±0.001**

	Task 5		Task 6		Task 7		Task 8	
	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
KM	0.525±0.000	0.635±0.000	0.494±0.003	0.588±0.012	0.732±0.013	0.574±0.002	0.792±0.026	0.631±0.037
LSA	0.477±0.000	0.360±0.002	0.587±0.000	0.545±0.000	0.752±0.000	0.512±0.000	0.766±0.000	0.521±0.000
ASI	0.540±0.028	0.270±0.069	0.603±0.004	0.384±0.025	0.602±0.016	0.488±0.022	0.592±0.003	0.347±0.015
BigClam	0.706±0.003	0.539±0.008	**0.750±0.001**	0.589±0.005	0.713±0.005	0.656±0.000	0.662±0.004	0.574±0.000
All KM	0.527±0.000	0.543±0.016	0.523±0.000	0.545±0.016	0.544±0.000	0.566±0.012	0.539±0.000	0.530±0.005
All LSA	0.524±0.000	0.385±0.000	0.339±0.000	0.371±0.000	0.335±0.000	0.327±0.000	0.337±0.000	0.320±0.000
All BigClam	0.606±0.000	0.615±0.000	0.698±0.000	0.703±0.000	0.646±0.000	0.676±0.000	0.619±0.000	0.582±0.000
LSSMTC	0.634±0.023	0.496±0.079	0.659±0.037	0.476±0.080	0.650±0.000	0.572±0.007	0.673±0.010	0.576±0.012
STMTC	**0.748±0.019**	**0.624±0.036**	0.689±0.012	**0.625±0.010**	**0.765±0.021**	**0.678±0.017**	**0.832±0.010**	**0.668±0.000**

the label provided by the corpus, respectively. The cluster accuracy is defined as follows:

$$Acc = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \tag{14}$$

where n is the total number of tweets, $\delta(x, y)$ is a delta function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [21]. 2) *Normalized Mutual Information*: Normalized Mutual Information (*NMI*) [11] is used to measure the quality of clusters. Let $\mathcal{L}$ denote the set of clusters obtained from the ground truth and $\mathcal{C}$ those obtained from our algorithm. The normalized mutual information metric is then defined as follows:

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} \frac{n_{i,j}}{n} \log \frac{n \cdot n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i \log \frac{n_i}{n})(\sum_{i=1}^{c} \hat{n}_j \log \frac{\hat{n}_j}{n})}} \tag{15}$$

where n_i denotes the number of tweets contained in the cluster $\mathcal{C}_i (1 \leq i \leq c)$, $\hat{n}_j$ is the number of tweets belonging to class $\mathcal{L}_i (1 \leq j \leq c)$, and $n_{i,j}$ is the number of tweets that are in the intersection between the cluster $\mathcal{C}_i$ and the ground truth class $\mathcal{L}_j$; the larger the NMI, the better the clustering result.

3) Methods for Comparison: To evaluate the clustering results, we compare the proposed STMTC methods with typical single-task and multi-task clustering methods, including Kmeans (KM), Latent semantic analysis (LSA)+Kmeans (LSAKM), adaptive subspace iteration (ASI) [22], Cluster Affiliation Model for Big Networks (BigClam) [8], Shared Subspace Multi-Task Learning (LSSMTC) [11], which considers shared subspace. We also present the experimental results for the clustering data for all the tasks together using KM, LSAKM, and BigClam. Note that clustering the data via LSSMTC corresponds to the proposed method with $\theta_1, \theta_2 = 0$.

B. Evaluation of Story

The evaluation of the new STMTC method is performed based on the clustering quality and case study analysis.

1) Quality Analysis: Accuracy and NMI metrics are used to evaluate the clustering results for the STMTC method. Each experiment is repeated 5 times, and the average results are shown in Table I for the LATAM dataset. The experimental results show that the proposed STMTC method achieves the best overall performance: 11.48% and 18.81% higher than BigClam and LSSMTC in cluster accuracy, which are the two methods outperform the others. Similarly, the NMI of STMTC is 34.32% higher than other methods in average. Although for some individual tasks, BigClam and All BigClam achieve better results. This is because our method may trade off the results of individual tasks for overall performance. As these results demonstrate, simply clustering the data for all the tasks together does not necessarily improve the clustering result: All KM, All LSAKM, All BigClam actually achieve worse result than their original algorithms, such as All KM is 19.74% and 9.9% lower than KM method in clustering accuracy and NMI. Because the data distribution for each task is different, the direct combination of different tasks violates the i.i.d assumption in single task clustering.

2) Case Study: To help people make sense of complex stories in massive Twitter datasets, we chose to use *Paris Attack* event for our case study. The most representative tweet from the top 10 ranked tweets in the cluster were selected in the story line. As shown in Figure 2, the main structure of the story is as follows: the blue line describes the

story of the terror attack itself and the arrests of the suspects, while the orange line focuses on the local and international assistance provided for the victims. Along the blue line, some key facts are shown: 1) the Bataclan theater attack, 2) the Stade de France bombing, 3) the weapon found, and 4) the suspects' arrest. The orange line presents the local activities directed towards helping the victims after the attack and the international assistance provided by the U.S and the European Union. Key terms such as *explosion*, *weapon*, *US*, and *ally*, contribute to the smoothness between successive nodes, while *attack* and *help* are the shared subspace of the story. Since the multi-language and semantic enhancement of feature extraction is lacking in the baselines, we found some key facts such as "Stade de France bombing" are missing in their results.

V. Conclusions

This paper presents a novel approach, TRACES, to generate story lines in massive Twitter datasets that produce fresh insights into complex stories. A novel multi-task clustering algorithm, STMTC, is proposed to generate the story lines and integrate them with shared subspace and temporal smoothness. The extensive experimental results for the diverse datasets clearly demonstrate the high quality of multi-task clustering and the story lines it generates through a comparison with existing state-of-the-art methods.

References

[1] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. Sumblr: continuous summarization of evolving tweet streams. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 533–542. ACM, 2013.

[2] Dafna Shahaf, Jaewon Yang, Caroline Suen, Jeff Jacobs, Heidi Wang, and Jure Leskovec. Information cartography: Creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1097–1105, New York, NY, USA, 2013. ACM.

[3] Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. Generating event storylines from microblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 175–184, New York, NY, USA, 2012. ACM.

[4] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: An exploration of temporal text mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 198–207, New York, NY, USA, 2005. ACM.

[5] Theodoros Lappas, Benjamin Arai, Manolis Platakis, Dimitrios Kotsakos, and Dimitrios Gunopulos. On burstiness-aware search for document sequences. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 477–486, New York, NY, USA, 2009. ACM.

[6] Dingding Wang, Li Zheng, Tao Li, and Yi Deng. Evolutionary document summarization for disaster management. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 680–681, New York, NY, USA, 2009. ACM.

[7] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 745–754, New York, NY, USA, 2011. ACM.

[8] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 587–596, New York, NY, USA, 2013. ACM.

[9] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 109–117, New York, NY, USA, 2004. ACM.

[10] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. In *Journal of Machine Learning Research*, pages 615–637, 2005.

[11] Quanquan Gu and Jie Zhou. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on*, pages 159–168, Dec 2009.

[12] Quanquan Gu, Zhenhui Li, and Jiawei Han. Learning a kernel for multi-task clustering. In *AAAI*, 2011.

[13] Jianwen Zhang and Changshui Zhang. Multitask bregman clustering. *Neurocomputing*, 74(10):1720 – 1734, 2011.

[14] Zhihao Zhang and Jie Zhou. Multi-task clustering via domain adaptation. *Pattern Recognition*, 45(1):465 – 473, 2012.

[15] Chris Ding, Tao Li, Michael Jordan, et al. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45–55, 2010.

[16] Liang Zhao, Feng Chen, Jing Dai, Ting Hua, Chang-Tien Lu, and Naren Ramakrishnan. Unsupervised spatial event detection in targeted domains with applications to civil unrest modeling. *PLoS ONE*, 9(10):e110206, 10 2014.

[17] Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 814–822, New York, NY, USA, 2011. ACM.

[18] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[19] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *In NIPS*, pages 556–562. MIT Press, 2000.

[20] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 63–72. IEEE, 2008.

[21] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[22] Tao Li, Sheng Ma, and Mitsunori Ogihara. Document clustering via adaptive subspace iteration. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 218–225, New York, NY, USA, 2004. ACM.