SansText: Classifying Temporal Topic Dynamics of Twitter Cascades Without Tweet Text

Shashidhar Sundereisan*, Abhay Bhadriraju†, M. Saquib Khan*, Naren Ramakrishnan* and B. Aditya Prakash*
*Department of Computer Science, Virginia Tech.
†Department of Electrical and Computer Engineering, Virginia Tech.
Email: *{shashi, mohak12, naren, badityap}@cs.vt.edu, †abhayr@vt.edu

***Abstract*—Understanding the dynamics of cascades in Twitter is an important modeling problem with multiple applications like viral marketing and the detection and forecasting of emerging events. Key hashtags rise in popularity to a peak and fall, with profiles characteristic to the specific topical area of the hashtag. Traditional text-based classification approaches are inadequate as new hashtags get created dynamically and because social media vocabulary evolves. We demonstrate a text-free approach `SansText` to classify emerging cascades by modeling the phenomenological patterns of rise and fall. We illustrate the utility of this approach over several specific event classes as well as more general topics in a collection of more than *2 million* tweets from multiple countries of Latin America.**

I. Introduction

Twitter and similar forms of social media have are now accepted as *surrogate data sources* that can provide insight into real-world events, e.g., box office sales, influenza outbreaks, and even the movement of earthquakes [23]. Rather than being passive indicators of such events, the effect of influence cascades on Twitter (e.g., propagation of real information or misinformation such as rumors) have shown that social media is very much an active participant in the progression of such events.

Our goal is to identify signals in Twitter that can serve as precursors to population-level events such as flu outbreaks, civil unrest, and elections. Traditional approaches to characterizing and forecasting such events rely on a fixed vocabulary of keywords or hashtags that are tracked through cascades and which are then input to machine learning models for classification. In practice, however, new hashtags emerge as required for a situation, new keywords underscore emergent phenomena, and thus purely text-based approaches are inadequate for dealing with the multitude of possible events that could arise.

We demonstrate the use of phenomenological models to capture the dynamics of information propagation, in particular the use of `SpikeM` [17] model that leverages statistics about a partially revealed cascade to determine the class of events that the cascade is likely to signify, e.g., a political event versus a sports event. Modeling the rise and fall patterns quantitatively provides a text-free approach to detect and classify emerging events, hence our system's name `SansText`, which we believe to be of interest more generally.

We demonstrate the utility of this approach on a dataset of more than *2 million* tweets gathered from the Latin American countries of Argentina, Brazil, Chile, Colombia, Ecuador, El Salvador, Mexico, Paraguay, Uruguay, and Venezuela, over the past two years. Latin America is a rich testbed for our project because of the diversity of events that happen due to empowered citizenry and rapid permeation of digital media. In addition to using data on specialized protest events in Latin America, we demonstrate the applicability of our approach more generally on popular high-volume topics on Twitter.

Our contributions can be summarized as:

1) *Problem Formulation and Approach*: We formulate the domain classification problem using activity profiles, and propose a simple yet powerful and efficient low-cost approach based on learning an aggregate information diffusion model (see Section IV).
2) *General Topics*: We demonstrate the effectiveness of our approach via first classifying simple popular keywords to domains. We compare against several baselines, and also explore the robustness of our approach to different parameter sets of our model (see Section VI).
3) *Protest Data*: We demonstrate the effectiveness of `SansText` on multiple tasks of different granularity, using protest event data from South America. We also show that `SansText` can outperform the baselines and is robust to limited data (see Section VII).

II. Related Work

Although much work has been done on finding topics from tweet text (c.f. [12]), we briefly review closely related work in the context of the dynamics of information diffusion and other more general time-series methods, as the focus of our paper is on activity profiles.

Information Diffusion: [5], [9] study the structural properties in the spread of information in networks, including the blogspace. [22] observed that hashtags diffuse as a complex contagion, and the nature of information diffusion differs with the topics. They did a study on hash-tags in twitter and found that there is a significant variation in the ways that widely-used hash-tags on different topics spread. The differences in their study manifested mostly in the particular probabilities of infection (the so-called 'P-K' curve) based on the number of friends infected. While their study does provide some insight into the diffusion process, it is not predictive, and it is not built for forecasting of events. In [1], the authors study how information propagation is effected by user interests in the twitter network. While in [21], the authors used the cascade ratio and the tweet ratio to understand how cascades of various topics diffuse in Twitter.

The preceding work looks more at the *structural* aspects of propagation. In contrast there is also work on studying just the *temporal* aspects of information propagation. Crane and Sornette [8] studied the rise-and-fall patterns of Youtube video views in a population and found that there were 4 classes, based on a self-excited Hawkes process [11]. Similarly, Yang et al. [25] explores the temporal patterns associated with online content and found there were 6 classes (associated with different *sources*). Matsubara et al. [17] showed that all these patterns can be generated from a single unified model SpikeM, which is succinct and yet powerful. SpikeM has been used before to model some malware propagations as well [18]. In [24], the authors propose a probabilistic framework to model and predict the popularity dynamics of individual items within a complex evolving system. However, these works do not focus on domain/topic classification problems. In this paper we show how to use such models for challenging domain classifications *without using semantics of the tweet text* itself.

To summarize, none of the above works deal with topic classification of online cascades using just activity profiles of keywords, based on an information diffusion model.

Time-series Analysis: This is an old topic with many textbook approaches [2]. Most methods like AR, ARIMA etc. are linear methods (and we describe some of them as baselines later). Non-linear methods tend to be hard to interpret (for example it is hard to relate them to actual physical models) and include [6] where the authors propose a fast and completely automated non-linear forecasting system which can provide estimation of vital forecasting parameters. Forex-foreteller (FF) [14] uses a linear regression model to make currency forecasts with high recall over precision. It explores correlative links between news and financial market fluctuations. It uses a language model to classify incoming news articles to build the forecasting model.

III. Background

Here we briefly describe the recently proposed SpikeM model [17] for modeling the popularity of a hashtag cascade. Although it was proposed only for hashtags, as we will show later, we re-purpose it for more general keywords.

We are interested in the macroscopic properties of hashtag cascades in the network. The model assumes that if a user has used the hashtag in his tweet, that user has been infected. Once infected the user always stays in the infected state (as he or she already knows about the hashtag).

The model assumes a total number of N un-informed population ('bloggers') that can be informed ('infected') by the hashtag. Let $U(n)$ be the number of such bloggers that are *not* infected at time n; $I(n)$ be the count of bloggers that got infected up to time $n-1$; and $\Delta I(n)$ be count of bloggers infected exactly at time n. Then $U(n+1) = U(n) - \Delta I(n+1)$ with initial conditions $\Delta I(0) = 0$ and $U(0) = N$.

Additionally, we let β as the 'infectivity' (essentially popularity) of a particular hashtag. We assume that the popularity of a hashtag at any particular person drops as a specific power-law based on the elapsed time since the hashtag infected *that person* (say τ) i.e. $f(\tau) = \beta\tau^{-1.5}$. Finally, we also have to consider one more parameter for the model: the "external shock", or in other words, the first appearance of a hashtag: let n_b the time that this initial burst appeared, and let $S(n_b)$ be the size of the shock (count of infected bloggers).

Finally, to account for periodicity, we define a periodic function $p(n)$ with three parameters: P_a, as the strength of the periodicity, P_p as the period and P_s as the phase shift.

Putting it all together, the SpikeM model is

$$\Delta I(n+1) =$$
$$p(n+1)\left(U(n)\sum_{t=n_b}^{n}\left(\Delta I(t) + S(t)\right) f(n+1-t) + \epsilon\right)$$

where $p(n) = 1 - \frac{1}{2}P_a\left(\sin\left(\frac{2\pi}{P_p}\left(n + P_s\right)\right)\right)$, and ϵ models noise.

IV. Method

A. Problem Formulation

Informally, the general problem we aim to solve is to classify a keyword into the correct domain, depending only on the temporal characteristics of the activity profile of that keyword. Note that there can be more sophisticated methods which can be employed for this purpose which leverage the actual tweet text as well. Nevertheless we believe our framework gives a different low-cost approach, which is surprisingly powerful and gives robust results and hence is interesting in its own right. More formally our problem stated as:

GIVEN: The temporal activity profile for a keyword on Twitter

FIND: The correct domain class label for the keyword.

In this paper, the particular 'domains' and 'keywords' are motivated by two real-world examples: popular high-volume topics like politics, sports etc., and more specialized 'protest-types' which categorize real-world protest events in Latin America. We will describe these in more detail later in Sections VI-A and VII-A.

B. Proposed Approach

Romero et al [22] discuss the differences in the mechanics of information diffusion, particularly in the so-called 'probability of infection' curve, across different domains. With this in mind, we posit that even the temporal propagation signature of hashtags from different domains are likely to be different, and these differences affect the popularity time series for each hashtag. Further, model parameters obtained for the popularity time series for hashtags from the same domain exhibit similarities, which can be learnt by using appropriate algorithms.

The SpikeM [17] Model provides an analytical tool for modeling popularity time series. It fits an exponential rise and a power law fall to data, and takes into account the periodicity of the activities too, as described in Section III. It is able to model real Twitter data well, where data shows exponential rises and power law falls, along with periodic trends with peaks during weekends. This model has the added advantage that model parameters consider orthogonal aspects of the spread of the infection.

S_b	The value of the external shock applied
β	The infectivity parameter of the virus
n_b	The time at which the external shock S_b was applied
P_a	The amplitude of the periodic part of the time series
P_s	The phase shift of the periodic part of the time series
P_p	The periodicity of the time series
ϵ	The error term

TABLE I. LIST OF PARAMETERS USED IN THE SPIKEM MODEL

The list of the seven SpikeM parameters with a brief explanation is given in Table I. If $X(n), n = 1 \cdots T$ is the sequence of count of keyword occurrences we want to model, we minimize the following:

$$\min_{\boldsymbol{\theta}} \sum_{n=1}^{T} \left(X(n) - \Delta I(n)\right)^2$$

where $\boldsymbol{\theta} = [N \quad \beta \quad S_b \quad P_a \quad P_s \quad P_p]^T$ is the vector of model parameters. We use Levenberg-Marquardt [16] to learn the parameters.

Using the SpikeM parameters learnt from keyword activity profiles as features and training data, our framework SansText learns a classifier that can classify keywords to domains. Classification accuracy is used as a metric to judge the ability of a classifier to classify hashtags to domains.

We find that, though SpikeM parameters can be used successfully to predict the topic of a keyword, only a subset of them are relevant to the classification problem. Thus, we further analyze the importance of each SpikeM parameter to domain-wise classification. Some parameters like n_b, which determines the day the news broke out in the social network, may not be a good predictor for all topics. On the other hand, the value of N, which is the number of people interested in the topic (the inherent 'audience'), turned out to be useful.

We next describe our extensive set-up for SansText in more detail.

V. SETUP

A. Overview

The motivation for our SansText approach is to classify entire tweets based on a small number of keywords. We use 'hash-tags', which generally denote particular contexts, and have been extensively used in Twitter studies before. For example, tweets that have the hash-tag "#manchesterutd" would talk about the soccer club Manchester United and hence the tweet can be classified as belonging to the topic sports. Another way to find such keywords is to use the help of domain experts to find out which keywords are popular in a particular topic (especially in context of protests datasets).

We use both these methods while collecting the data for our experiments. Once these keywords are collected the next step is to collect tweets that have this keyword. We use a large sample of all South American tweets for this purpose. Geographic targeting was done through both geo-tagging and a user referencing their own location in the tweet text or their profile. We use these collected tweets and aggregate them to count the number of occurrences of a keyword in a time period. Thus we obtain a volume time series for each keyword which we can fit with the SpikeM model and collect the parameters of the model. The next step is to use these parameters as features for the classification problem of finding which topic the keyword/hash-tag belongs to. We use different methodologies for getting the ground-truth labels for keywords for each of the experiments, which we will describe later. A subtle point is that we do not use parameter n_b from set of parameters in Table I for classification, as it represents the day the initial spike was observed and hence is not an *inherent* property of the spreading cascade. It is easy to see that this makes some of our predictions *harder*, as there are some topics occur only in certain times of the year e.g. flu occurs mainly in March-Sept (in S. America), and hence just by looking at the timestamps we can guess if the topic of a keyword is flu. Moreover, as our focus in this paper is on actual dynamics, we do not use this feature.

After learning the parameters we test SansText against multiple intuitive and non-trivial baselines listed in V-B using classifiers described in V-C. We describe this process in more detail for each of our experiments later in Sections VI-A and VII-A.

To throughly evaluate our approach, in our experiments we pose the following research questions:

1) Does the choice of the time interval of the aggregation change our results?
2) Are SansText parameters a good feature set for domain-wise classifications?
3) Which SansText parameters are important to the classification problem for?
4) Can we use SansText when the keywords are spread across multiple topics?
5) How much data do we need to make a successful classification?

B. Baselines

We compared SansText to a trivial baseline (Majority) and three non-trivial baselines each of which essentially give us features for each time-series. We give these baselines the same data that SansText gets i.e. a set of time series. We describe each of them briefly:

Majority: The Majority is a trivial baseline and it always gives the output of the class with maximum frequency no matter what the input is. Hence, we are not performing any 'learning' in this method.

Euclidean: The Euclidean distance is the simplest distance between two series x and y of length n and is defined as $\sqrt{\sum_{i=0}^{n}(y_i - x_i)^2}$. A distance of zero implies that the two series are exactly the same. The attributes chosen for classification were the distances to every other keyword. The intuition for this baseline stems from the idea that all keywords from the same topic would have small distances. We would like to point out that such a method has also been used frequently [15]. Apart from the natural problems of the euclidean distance (as it is too 'rigid'), the other disadvantages of this baseline are that for every new keyword we need to calculate the distances to all other keywords in the dataset and that we need to save all $\binom{n}{2}$ distances for classification.

DTW: Dynamic Time Warping (DTW) is also a popular robust distance metric between two time series. It is extensively used in areas like speech processing [19]. The main difference between DTW and Euclidean is that DTW calculates the distance by taking into account that the two series can have peaks at different times. Just like Euclidean, we use the distances from each of the keywords as parameters for the classification problem. We used the recent fast implementation by Rakthanmanon et. al. [20] for finding out these distances[1]. DTW has disadvantages similar to that of Euclidean.

Fourier: The Fourier Transform decomposes a given function into a sum of periodic functions of the form $e^{i\pi n}$ [3]. The Discrete Fourier Transform (DFT) uses functions of the form $e^{\frac{i*2\pi k}{N}}; k \in I; 0 \leq k \leq N$ [3]. Thus, the DFT gives a finite set of coefficients for a discrete time series.

For discrete time series of equal lengths, the same functions are used as the basis when computing the DFT. Each coefficient generated in the DFT can be treated as a feature of the given time series. Intuitively, the DFT allows us to identify the significant periodicities in a time series, thus allowing learning and classification based on similarities in the periodic nature of time series. The disadvantage of this baseline is that we have to save coefficients atleast half the size of the time series. To compute the DFT, we use the Cooley-Tukey FFT algorithm [7], as implemented in the NumPy numerical analysis package for Python.

C. *Classifiers*

We use the following popular classifying methods for our experiments:

1) *Multilayer Perceptron*: In this method, the weights between 3 layers of a neural network are learnt using back propagation.
2) *C4.5*: This is a classic tree based method which uses Information gain as the criteria to split the attributes.
3) *Random Forests*: A set of different classification trees are constructed using different subsets of parameters to learn the model.
4) *Bagging*: It is a method that is used to reduce the variance in the model that we learn by using the same algorithm on different training sets generated by random sampling. We use as the base classifier as REPTree.
5) *Logistic Regression*: Uses the logistic function in order to find a boundary between various classes.

A more detailed description and comparisons between these supervised learning methods is in [4]. In all our tests we use Weka's [10] implementations of these algorithms with the default parameters to the algorithms, unless stated otherwise. Additionally, in all our experiments we use 10-fold cross validation to report the classification accuracy.

VI. Experiments on Popular Data

A. *Data Collection*

In this study, we are interested in tweets from these popular domains: Political, Flu, Sports, Technology and Idioms (these domains have also been used before in prior studies). We define these domains and give examples in Table II. Using Datasift's collection service[2], we collected a list of top 300 hash-tags (by volume) from June 2012 to May 2013 and divided it into these domains. The division was carried out by using majority vote among three of the authors (similar to methodology used in [22]). We show the division of the hash-tags in the online Appendix[3]. The domains Flu, Idioms, Technology, Sports and Political had 11, 10, 11, 12 and 14 hash-tags respectively. After collecting these hash-tags we extract the timestamps of each of occurrence of the hash-tag in a tweet. We then aggregate these occurrence numbers by day and by week to generate a time series of the mentions of the hashtag. Note that every hash-tag would have its own time-series. We then used our model to find the set of parameters that fit the time series. We show one of these plots for three different domains for both the weekly and the daily settings in Figure 1.

B. *Results*

a) Does the choice of the time interval of the aggregation change our results?:
SansText exploits the particular rise-fall nature of the time-series for fitting the model. Hence it is possible that too high/low a granularity will bury/wash out the patterns. We performed all the experiments in this section with both the daily setting as well as the weekly setting—both of these settings show similar results. Part of the reason is that SansText has an explicit periodicity parameters, so it is tolerant to an extent to such natural aggregation levels. Hence due to lack of space, we describe the results of just the daily setting.

b) Are SansText parameters a good feature set for domain-wise classification ?:
In short from Figure 6(a), SansText outperforms all the baselines. Hence the parameters that we use are a good set of features for classification in the Popular Dataset.

We used the parameters that we get from the model fittings as attributes and used classifiers listed in section V-C. Since we use 5 different classification algorithms for each method we only report the values for the best performing one (it could be different for different methods). We show the % improvement of the methods over the weakest method in Figure 6(a). Since Political was the class with the highest frequency of hashtags, the Majority will always predict every hashtag to be Political.

As expected Fourier does not perform well, as while DFT does well with periodicities, it does not do well with spikes (as we need co-efficients from across the frequency spectrum because of a spike in the time-domain). A bit surprising result was the Euclidean fared better than DTW in the daily setting while not in weekly setting. This is because in the daily setting the time-series have multiple local peaks (due to similar periodicities) which align across the time-series (while the major rise-fall peak itself may not align). Euclidean will get a better distance, whereas DTW tries to align the main peak and makes mistakes in the local ones. On the other hand, in the weekly setting, there are fewer periodicities, and hence aligning the main peak is more important, which DTW does it successfully. This also shows that for any method to be successful we should treat the periodic nature of the time series

[1] Code can be found here: http://www.cs.ucr.edu/~eamonn/UCRsuite.html

[2] www.datasift.com

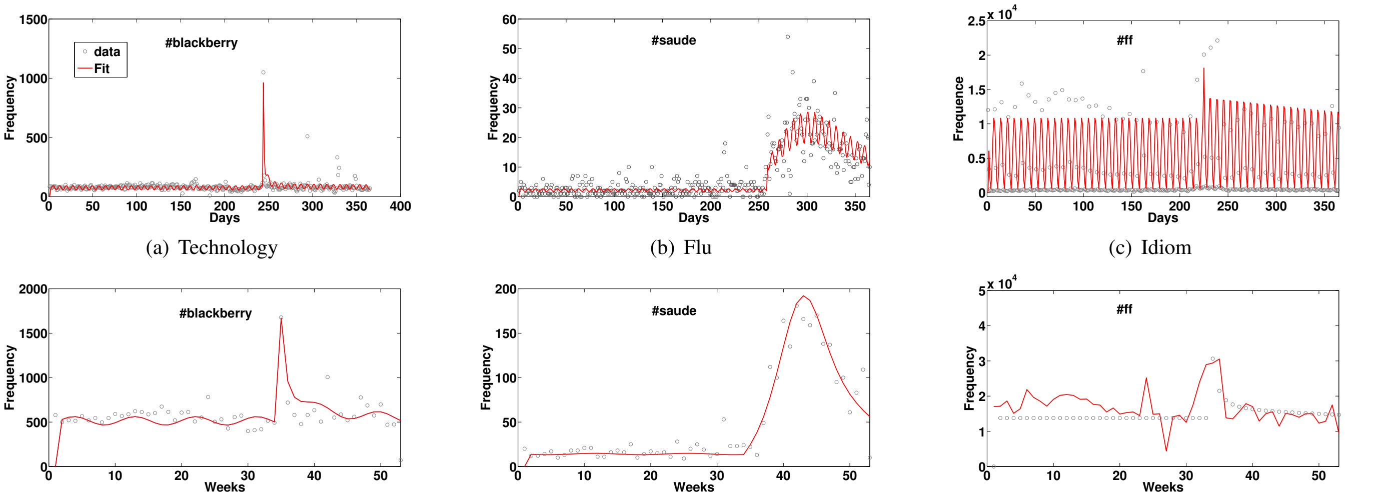
[3] http://people.cs.vt.edu/~shashi/sanstext/

Topic	Description
Idioms	Hash tags that are a group of words or their abbreviations connected together. These hash tags have a conversational meaning like #ff stands for "Follow Friday" a trend which recommends whom to follow this Friday on Twitter.
Flu	Hash tags that related to being hit by Flu like symptoms of flu. Example #fiebre which means Fever.
Technology	Hash tags which relate to Technology like #apple (the company).
Sports	Hash tags which relate to sports like #londres2012, the Olympics in 2012.
Politics	Hash tags which talk about politics like #caprilespresidente which talks about Henrique Capriles Radonski for the next president of Venezuela.

TABLE II. DESCRIPTION OF TOPICS USED IN POPULAR DATASET.

Class	Description
Non Violent Government Policies	Policies by the government that resulted in non-violent protests.
Non Violent Energy and Resources	Protests over energy or resources that were non-violent eg: Hike in gas prices.
Violent Energy and Resources	Protests over energy or resources that were violent.
Non Violent Other	Non-violent protests that have reasons other than energy, resources, govt. policy, housing and employment
Violent Other	Violent protests that have reasons other than energy, resources, govt. policy, housing and employment

TABLE III. CLASS DISTRIBUTION FOR 5 EVENT TYPE CLASSIFICATION PROBLEM IN PROTEST DATASET



(a) Technology

(b) Flu

(c) Idiom

(d) Technology

(e) Flu

(f) Idiom

Fig. 1. How well does SansText model the data? We compare the learnt model with the data for Daily (top row) and weekly (bottom row) granularity for different domains, and plot volume per unit time against time. We observe that SansText (blue) does a good job for fitting the exponential rise and power-law fall of the real data (red).

different from the actual rise and fall for the method to be robust across different settings—which is exactly achieved by SansText.

c) Which SansText parameters are important to the classification problem ?:
In short, from the correlation matrices in Figure 2 and the % improvement when we remove parameters in Figure 4(a), we can infer that *all* the parameters are useful for classification.

To answer this question we investigated two things: (a) we removed each parameter one by one and found out its effect on the classification accuracy (ablation test); and (b) we measured the correlation coefficient of each of the features. We show the results of this experiment for the algorithm that produced the best results in the previous section in Figure 4(a). We are also interested in finding out if there exists redundancy in the set of attributes that we use for classification. If any two of the attributes are highly correlated then we can reduce the feature set. We computed the Pearson's correlation coefficient between all the 7 attributes we used for classification. The correlation matrix is presented in Figure 2.

We observe from Figure 4(a) that only removing β increases the classification accuracy (albeit very slightly). But we observe that β is not correlated with any other parameter as shown in the Figure 2. We also observe that only two parameters have a high value of correlation (> 0.5). These parameters are N and ϵ. We argue that by removing either one of these would affect the classification accuracy as shown in Figure 4(a). The reason that the Pearson's coefficient is large between them is because when there are a lot of people talking about a topic (N) there is a lot of noise (ϵ) in the dataset. From the C4.5 tree we learn for this dataset, we infer that Sports hashtags generally have a high noise ($\epsilon > 250$). Also Technology hashtags have a large shock ($S_b > 14000$) or have a lot of fan following ($\epsilon > 100$).

	N	β	S_b	ϵ	P_a	P_s	P_p
N	1.00	$\emptyset$	$\emptyset$	0.86	$\emptyset$	$\emptyset$	$\emptyset$
β	$\emptyset$	1.00	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
S_b	$\emptyset$	$\emptyset$	1.00	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
ϵ	0.86	$\emptyset$	$\emptyset$	1.00	$\emptyset$	$\emptyset$	$\emptyset$
P_a	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1.00	0.36	$\emptyset$
P_s	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0.36	1.00	$\emptyset$
P_p	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1.00

Fig. 2. Are the seven parameters correlated in the Popular Dataset ? As we can see there is little correlation between the parameters. All values in the range (-0.3, 0.3) are shown as $\emptyset$. Please see section VI-B for explanation on why N and ϵ have a high correlation.

d) How much data do we need to learn `SansText` *parameters ?:*
In this section we will try to find out how much of the time series data is needed to learn `SansText` parameters for the Popular Dataset. We took the hash-tag #saude from the topic Flu for this part of the experiment. We tried learning the parameters for `SansText` for data till n_b, data till n_b-3 and data till n_b+3 and plotted the resulting fits in Figure 3 (left). We observe that all the plots look similar till n_b+3. This suggests that we learn parameters in `SansText` in a robust way. With more data the Root Mean Square Error (RMSE) decreases as follows: 22.14 (green), 19.25 (blue) and 16.91 (red curve). This also implies that we do not need data till the peak in order to learn the parameters.

VII. Experiments on Protest Data

A. Data Collection

For the second dataset, we have access to a Gold Standard Report (GSR) of protests organized by an independent third party (MITRE). The GSR is a database generated by human analysts who scour newspapers in Latin America for reported happenings of civil unrest. This report has a list of all protests along with other metadata like where the protest occurred, the type of the event, the type of population involved, the date of the event, and the date of reporting. We used the keywords collected for these events by [13]. The authors in [13] used location based collection of tweets surrounding the event with a date range of ± 10 days and used TF-IDF to find the important keywords. A subtle point is that each event could have multiple keywords associated with it. With these sources for our data we try to solve two tasks

Task 1: Keyword to Event Type Mapping We have a set of keywords that are known to be related to protests. We can monitor each one of them and find out if they are gaining popularity or not. If a protest related keyword is known to show popularity over a period of time we want to find out if the temporal pattern it displays can help us predict which event type the developing protest belongs to. Since every keyword can belong to multiple event types, we try to find out if `SansText` can be used to predict the event type in such a scenario.

Task 2: Event to Event Type Mapping We find that many events have more than one keyword associated with them. Just as done in task one, we monitor the popularity of various keywords which are known be related to protests. When we observe a set of keywords trending from one geographic location at around the same time, we will collect all their mentions and treat them as an event. In this task, we try to predict what event type an event belongs to. We consider the event time series as the aggregation of all the keywords associated with it.

We describe some of the event types (classes) that are possible in Table III. Note that every keyword can belong multiple event types. We collect all the tweets that use these keywords in the ± 10 days of the event. We then fit our model on this time series and find all the parameters. We then use these parameters as features to predict the event type using the classifiers described in section V-C.

As the results we obtain in both tasks were similar we will describe the results of both the tasks together. We show the keywords and their classes in the online appendix (as in the previous section).

B. Results

a) Can we use `SansText` *when the keywords are spread across multiple topics?:*
In short from the % improvement Figure 6(b), we can infer that `SansText` works even when the keywords are spread unto different topics. We perform better than all other baselines.

Till now we used keywords that belong to a single class. There are instances in which a keyword can belong to different classes. Since `SansText` does not look at the context with which the keyword was used, every keyword-event type instance is a new data point in out setting. We propose to find the event type by looking at how the behavior changes when a keyword is used for different event types. Note that this sort of classification would be a hard task if we were to use traditional methods like NLP since we would have to use the context around the keyword in the tweet text. We can see in Figure 7 (top row) that even though the keyword 'cantar' belongs to various event types the pattern observed in each one of them is different. We try to use these differences in classifying the keyword 'cantar'.

Like in Popular dataset we only report the values for the best performing one (it could be different for different methods). For **Task 1**, Figure 6(b) shows that `SansText` gives an improvement of 165% over the weakest baseline i.e. `DTW`. `SansText` performs atleast twice as better than most of the baselines while we perform 72% better than the next best baseline i.e. `Euclidean`. While for **Task 2**, Figure 6(b) shows that `SansText` performs better than all the baselines.

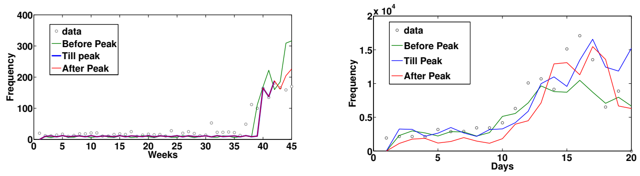
b) Which `SansText` *parameters are important to the classification problem?:*
As we can see from the correlation matrices in Figure 5 and the % improvement while removing parameters one by one in Figure 4 (right), we can infer that all the parameters used are important. Hence we can't remove even one of them from `SansText`.

We performed ablation tests like in Popular dataset. We show the results of this experiment for the algorithm that produced the best results in the section above i.e. Multilayer Perceptron in Figure 4 (right) for both **Task 1** as well as **Task 2**. We compare this result with the correlation coefficient between our parameters. Again, if any two of the attributes are highly correlated then we can reduce the feature set. We found the Pearson's correlation coefficient between all the 7 attributes we used for classification (see Figure 5). We observe from Figure 4 (right) that like in the Popular Dataset only removing β increases the classification accuracy. But we observe that β is not correlated with any other parameter as shown in the Figure 5. We observe from Figure 5 that there is no correlation between any of the parameters and hence we can not remove any of the parameters from `SansText`.

From the C4.5 classification tree we can infer that Non-Violent protests of type 'Energy and resources' have low fluctuations in temporal data if the spike is ignored ($P_a < 0.4$). While keywords belonging to Non-Violent Other category had less than 29000 tweets in this dataset.

c) How much data do we need to learn `SansText` parameters?:
In this section we will try to find out how much of the time series data is needed to learn `SansText` parameters for the Protest Dataset.

We took the keywords for the event that corresponded to the Non-Violent Government Policies on 6th Feb 2013 in Brazil. We tried learning the parameters for `SansText` for data till n_b, data till n_b-3 and data till n_b+3 and plotted the resulting fits in Figure 3 (right). Clearly, we will learn better given more data, but we find that `SansText`'s performance is robust and does not degrade much. With more data the Root Mean Square Error(RMSE) decreases as follows: 2130.2 (green), 1997.83 (blue) and 1771.38 (red curve). This also implies that we do not need data till the peak in order to learn the parameters.



(a) Popular Dataset

(b) Protest Dataset

Fig. 3. How much data do we need to fit the `SansText` parameters? As we can see all the three plots are similar. This implies that we are robust in determining the `SansText` parameters accurately.

VIII. Conclusions

We have demonstrated that activity profiles for hashtags from different domains are modeled well under our framework `SansText`, and showcased its utility for domain classification without requiring any textual analysis. We have demonstrated its effectiveness and superiority over baseline methods in both a general topical domain and in a specialized domain, viz. protest modeling. Inference of the parameters enables the forecasting of keyword and event popularity and classification into different granularities, as evidenced by our results over Latin American tweets. Future work may look into learning even the exponent parameter (currently 1.5) in our framework and using it for classification.

Acknowledgements: The authors would like to thank Ting Hua for discussions about the keywords in the protest dataset. This work was partially supported by the NSF via grant IIS-1353346, by the NSA (under a Science of Security lablet), by the VT College of Engineering and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, IARPA, NSA, DoI/NBC, or the US Government.

References

[1] P. Bogdanov, M. Busch, J. Moehlis, A. K. Singh, and B. K. Szymanski. The social media genome: Modeling individual topic-specific behavior in social media. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 236–242, New York, NY, USA, 2013. ACM.

[2] G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[3] W. L. Briggs and H. V. Emden. *The DFT - an owner's manual for the discrete Fourier transform.* SIAM, 1995.

[4] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[5] D. Centola. The Spread of Behavior in an Online Social Network Experiment. *science*, 329(5996):1194, 2010.

[6] D. Chakrabarti and C. Faloutsos. F4: Large-scale Automated Forecasting Using Fractals. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 2–9, New York, NY, USA, 2002. ACM.

[7] J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90):297–301, April 1965.

[8] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system, 2008.

[9] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501. ACM New York, NY, USA, 2004.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009.

[11] A. G. Hawkes and D. Oakes. A Cluster Process Representation of a Self-Exciting Process. *Journal of Applied Probability*, 11(3):493–503, Sept. 1974.

[12] L. Hong and B. D. Davison. Empirical Study of Topic Modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM.

[13] T. Hua, C.-T. Lu, N. Ramakrishnan, F. Chen, J. Arredondo, D. Mares, and K. Summers. Analyzing Civil Unrest through Social Media. *Computer*, 46(12):80–84, Dec 2013.

[14] F. Jin, N. Self, P. Saraf, P. Butler, W. Wang, and N. Ramakrishnan. Forex-foreteller: Currency Trend Modeling Using News Articles. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1470–1473, New York, NY, USA, 2013. ACM.

[15] E. Keogh and S. Kasetty. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 102–111, New York, NY, USA, 2002. ACM.

[16] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[17] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 6–14, New York, NY, USA, 2012. ACM.

[18] E. E. Papalexakis, T. Dumitras, D. H. P. Chau, B. A. Prakash, and C. Faloutsos. Spatio-temporal mining of software adoption & penetration. In *ASONAM*, pages 878–885, 2013.

[19] L. R. Rabiner and B.-H. Juang. *Fundamentals of speech recognition.* Prentice Hall signal processing series. Prentice Hall, 1993.

[20] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Data Mining a Trillion Time Series Subsequences Under Dynamic Time Warping. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 3047–3051. AAAI Press, 2013.

[21] G. Rattanaritnont, M. Toyoda, and M. Kitsuregawa. Characterizing topic-specific hashtag cascade in twitter based on distributions of user influence. In *Proceedings of the 14th Asia-Pacific International Conference on Web Technologies and Applications*, APWeb'12, pages 735–742, Berlin, Heidelberg, 2012. Springer-Verlag.

[22] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 695–704, New York, NY, USA, 2011. ACM.

Parameters Used	% improvement
$\Theta \setminus P_a$	-5.2
$\Theta \setminus P_s$	-2.6
$\Theta \setminus P_p$	-23.1
$\Theta \setminus N$	-7.7
$\Theta \setminus \beta$	5.1
$\Theta \setminus \epsilon$	-20.6
$\Theta \setminus S_b$	-20.6

(a) Popular Dataset

Parameters Used	% improvement
$\Theta \setminus P_a$	-45.5
$\Theta \setminus P_s$	-4.6
$\Theta \setminus P_p$	-27.2
$\Theta \setminus N$	-18.2
$\Theta \setminus \beta$	9
$\Theta \setminus \epsilon$	0
$\Theta \setminus S_b$	-18.2

(b) Protest **Task 1**

Parameters Used	% improvement
$\Theta \setminus P_a$	-9
$\Theta \setminus P_s$	-11.5
$\Theta \setminus P_p$	-6.6
$\Theta \setminus N$	-16.9
$\Theta \setminus \beta$	3
$\Theta \setminus \epsilon$	-8.9
$\Theta \setminus S_b$	-13.9

(c) Protest **Task 2**

Fig. 4. Ablation Test: As we can see by removing the parameters one by one, we reduce the classification accuracy in most cases for the 3 datasets. Here $\Theta = \{N, \beta, \epsilon, P_a, P_s, P_p, S_b\}$, the list of parameters used in `SansText`.

	N	S_b	ϵ	P_a	P_s	P_p	β
N	1	$\emptyset$	0.40	$\emptyset$	$\emptyset$	$\emptyset$	-0.30
S_b	$\emptyset$	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
ϵ	0.40	$\emptyset$	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
P_a	$\emptyset$	$\emptyset$	$\emptyset$	1	-0.39	0.38	$\emptyset$
P_s	$\emptyset$	$\emptyset$	$\emptyset$	-0.39	1	$\emptyset$	$\emptyset$
P_p	$\emptyset$	$\emptyset$	$\emptyset$	0.38	$\emptyset$	1	$\emptyset$
β	-0.30	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1

(a) Protest **Task 1**

	N	S_b	ϵ	P_a	P_s	P_p	β
N	1	0.32	$\emptyset$	0.45	$\emptyset$	$\emptyset$	$\emptyset$
S_b	0.32	1	$\emptyset$	$\emptyset$	0.32	$\emptyset$	-0.33
ϵ	$\emptyset$	$\emptyset$	1	0.48	$\emptyset$	$\emptyset$	$\emptyset$
P_a	0.45	$\emptyset$	0.48	1	-0.36	$\emptyset$	$\emptyset$
P_s	$\emptyset$	0.32	$\emptyset$	-0.36	1	$\emptyset$	-0.27
P_p	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	$\emptyset$
β	$\emptyset$	-0.33	$\emptyset$	$\emptyset$	-0.27	$\emptyset$	1

(b) Protest **Task 2**

Fig. 5. Are the parameters of `SansText` in the Protest Dataset correlated? As we can see from the correlation matrix for **Task 1** and **Task 2** that there is no correlation between the parameters which means that we can't remove any one of them. All values in the range (-0.3, 0.3) are shown as $\emptyset$.

Method	% improvement
`Majority`	0
`Fourier`	103
`Euclidean`	155
`DTW`	134
`SansText`	173

(a) Popular Dataset

	% improvement
`SansText`	165
`Fourier`	27
`DTW`	0
`Euclidean`	54
`Majority`	33

(b) Protest **Task 1**

	% improvement
`SansText`	40
`Fourier`	0
`DTW`	20
`Euclidean`	11
`Majority`	5

(c) Protest **Task 2**

Fig. 6. How well do we perform compared to the baselines? As we can see we are performing way better than any other baseline in all the three cases

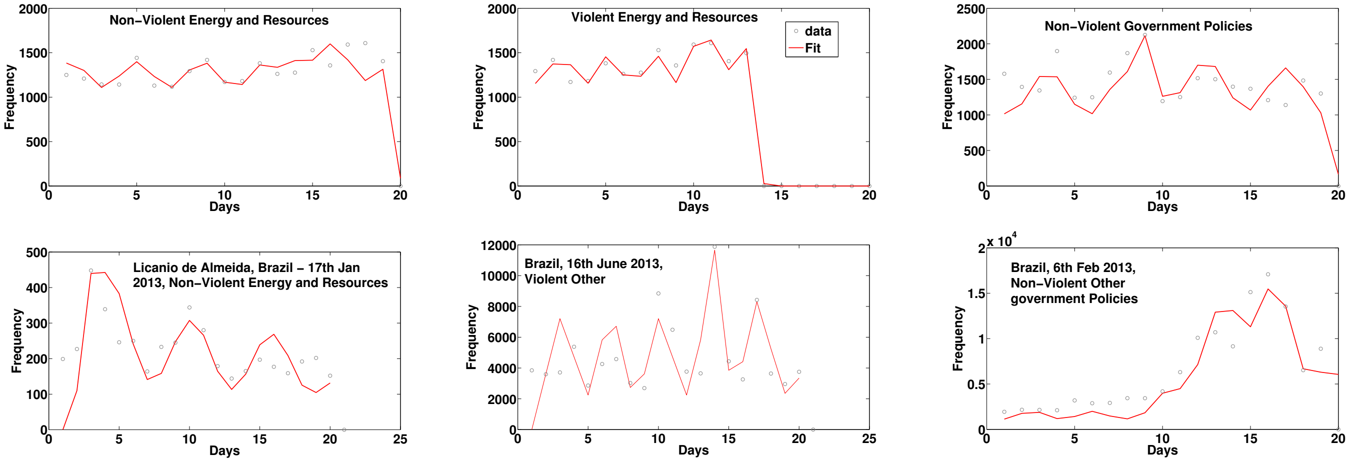


Fig. 7. How well does `SansText` model the data in Protest **Task 1** (top row) and Protest **Task 2** (bottom row)? We plotted the volume per unit time for the same keyword for different event types. As we can see for **Task 1** the patterns are not similar. We use this dissimilarity in **Task 1** to classify the keyword 'cantar' to an event type. While for **Task 2**, we show 3 events. We mention the type of protest with a text in the the figure.

[23] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[24] H.-W. Shen, D. Wang, S. Chaoming, and A.-L. Barabsi. Modeling and predicting popularity dynamics via reinforced poisson processes. In *The Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI, 2014.

[25] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 177–186. ACM, 2011.