Beyond Trending Topics: Real-World Event Identification on Twitter

Hila Becker[1], Mor Naaman[2], and Luis Gravano[1]

[1] Columbia University {hila, gravano}@cs.columbia.edu
[2] Rutgers University mor@rutgers.edu

Abstract. User-contributed messages on social media sites such as Twitter have emerged as powerful, real-time means of information sharing on the Web. These short messages tend to reflect a variety of events in real time, earlier than other social media sites such as Flickr or YouTube, making Twitter particularly well suited as a source of real-time event content. In this paper, we explore approaches for analyzing the stream of Twitter messages to distinguish between messages about real-world events and non-event messages. Our approach relies on a rich family of aggregate statistics of topically similar message clusters, including temporal, social, topical, and Twitter-centric features. Our large-scale experiments over millions of Twitter messages show the effectiveness of our approach for surfacing real-world event content on Twitter.

1 Introduction

Social media sites (e.g., Twitter, Facebook, and YouTube) have emerged as powerful means of communication for people looking to share and exchange information on a wide variety of real-world events. These events range from popular, widely known ones (e.g., a concert by a popular music band) to smaller scale, local events (e.g., a local social gathering, a protest, or an accident). Short messages posted on social media sites such as Twitter can typically reflect these events as they happen. For this reason, the content of such social media sites is particularly useful for real-time identification of real-world events and their associated user-contributed messages, which is the problem that we address in this paper.

Twitter messages reflect useful event information for a variety of events of different types and scale. These event messages can provide a set of unique perspectives, regardless of the event type [9, 23], reflecting the points of view of users who are interested or even participate in an event. In particular, for unplanned events (e.g., the Iran election protests, earthquakes), Twitter users sometimes spread news prior to the traditional news media [13, 20]. Even for planned events (e.g., the 2010 Apple Developers conference), Twitter users often post messages in anticipation of the event, which can lead to early identification of interest in these events. Additionally, Twitter users often post information on local, community-specific events (e.g., a local choir concert), where traditional news coverage is low or nonexistent.

Identifying events in real time on Twitter is a challenging problem, due to the heterogeneity and immense scale of the data. Twitter users post messages with a variety of content types, including personal updates and various bits of information [17]. While much of the content on Twitter is not related to any particular real-world event, informative event messages nevertheless abound. As an additional challenge, Twitter messages, by design, contain little textual information, and often exhibit low quality (e.g., with typos and ungrammatical sentences).

Several research efforts have focused on identifying events in social media in general, and on Twitter in particular [3, 7, 20, 21]. Event identification in social media sites such as Flickr is often performed retrospectively [7], finding patterns in the data after an event has occurred rather than identifying the event as soon as its associated content is posted. Recent work on Twitter has started to process data as a stream, as it is produced, but has mainly focused on identifying events of a particular type (e.g., news events [21], earthquakes [20]). Other work has aimed to identify the first Twitter message associated with an event [18].

Our focus in this work is on *online identification* of real-world event content. We identify each event—and its associated Twitter messages—using an online clustering technique that groups together topically similar tweets. We then compute revealing features for each cluster to help determine which clusters correspond to events. Importantly, we design features to distinguish between real-world events and a special family of non-events, namely, Twitter-centric "trending topics" that carry little meaning outside the Twitter system. These Twitter-centric activities often share similar temporal distribution characteristics with real-world events, as discussed below. Thus, distinguishing between these kinds of content is one challenging task that we address. Specifically, our contributions are as follows:

- We propose a general online clustering framework, suitable for large-scale social media sites such as Twitter, which employs a post-clustering classification step to identify real-world event content (Section 4.1).
- We identify revealing cluster features, to learn event classification models (Sections 4.2 and 4.3).
- We validate the effectiveness of our techniques using a dataset of over 2.6 million Twitter messages (Section 5).

Finally, we discuss our findings and future work (Section 6).

2 Related Work

We describe relevant related work in four areas: event identification in textual news, event identification in social media, and topic detection as well as event identification on Twitter.

Previous work on event identification in textual news (e.g., newswire, radio broadcast) [1] leveraged natural language processing tools (e.g., named-entity extraction, part-of-speech tagging) for online identification of news events in a

stream. Such tools do not perform well over Twitter messages, given the message length and characteristics, as noted above. More significantly, this line of research generally assumes that all documents contain event information. In contrast, the problem that we address is the separation of event messages from other messages.

While event detection in textual news documents has been studied in depth, the identification of events in social media sites is still in its infancy. Looking at text stream data from social data blogs and email, Zhao et al. [25] detect events using textual, social, and temporal document characteristics, but do so retroactively, not in "online" settings. Other research considers event identification in other social media data, such as Flickr [3, 7], where structured context features (e.g., title, description, tags) can help measure the similarity of social media documents (e.g., photographs) that correspond to the same event [3].

Twitter has attracted specialized attention among social media sites, with recent efforts focusing on detection of general topics [19] and trending topics [16, 6] in Twitter messages. While some topics on Twitter correspond to events, others reflect Twitter-specific conversations and other non-event content (see Section 3). The techniques in this paper go beyond detection of emerging topics to identify real-world *events*.

Few related papers explored the idea of detecting events on Twitter, but with different goals or constraints than the work we present here. Sakaki et al. [20] developed techniques for identifying earthquake events on Twitter by monitoring keyword triggers (e.g., "earthquake" or "shaking"). In their setting, the event must be known a priori, and should be easily represented using simple keyword queries. Sankaranarayanan et al. [21] identified late breaking news events on Twitter using clustering, along with a text-based classifier and a set of news "seeders," which are handpicked users known for publishing news (e.g., news agency feeds). Finally, Petrović et al. [18] used locality-sensitive hashing to detect the first tweet associated with an event in a stream of Twitter messages. We use the general text-based classifier suggested in [21] and a method for identifying top events suggested by Petrović et al. [18] as baseline approaches in our evaluation (Section 5).

3 Background and Problem Definition

In this section, we provide an overview of Twitter and the features of Twitter that are relevant to our work. We then define the problem that we address in this paper.

3.1 Background: Twitter

Twitter is a popular social media site, with nearly 200 million registered users as of January 2011. Twitter's core function allows users to post short textual messages, or *tweets*, which are up to 140 characters long. While there are several widely used services that enable the exchange of short messages (e.g., Facebook,

Tumblr), we focus our discussion and experiments on Twitter due to the prominence of this site and the availability of data. However, our techniques could be adapted to similar sites, with an appropriate mapping of the Twitter-specific classification features listed below.

Several features play an important role on Twitter. Specifically, Twitter users can use a *hashtag* annotation format (e.g., #sb45) to indicate what their posted messages are about (e.g., "watching Superbowl 45 #sb45") or capture other aspects related to the message. In addition, Twitter allows several ways for users to directly converse and interact by referencing each other in messages using the @ symbol. A *retweet* is a message from one user that is "forwarded" by a second user to the second user's social network, commonly using the "RT @username" text as prefix to credit the original (or previous) poster (e.g., "RT @justinbieber Tomorrow morning watch me on the today show"). A *reply* is a public message from one user that is a response to another user's message. Replies start with the replied-to user @username (e.g., "@mashable check out our Twitter study"). A *mention* is a message that includes some other username in the text of the message (e.g., "attending a talk by @pogue").

Twitter currently employs a proprietary algorithm to display *trending topics*, consisting of terms and phrases that exhibit "trending" behavior. While Twitter's trending topics sometimes reflect current events (e.g., "world cup"), they often include keywords for popular conversation topics (e.g., "#bieberfever," "getting ready"), with no discrimination between the different types of content.

3.2 Problem Definition

We now define the notion of real-world event in the context of a Twitter message stream, and provide a definition of the problem that we address in this paper.

The definition of event has received attention across fields, from philosophy [10] to cognitive psychology [24]. In information retrieval, the concept of event has prominently been studied for event detection in news [1]. We borrow from this research to define an event in the context of our work. Specifically, we define an *event* as a real-world occurrence e with (1) an associated time period T_e and (2) a time-ordered stream of Twitter messages M_e, of substantial volume, discussing the occurrence and published during time T_e.

According to this definition, events on Twitter include widely known occurrences such as the presidential inauguration, and also local or community-specific events such as a high-school homecoming game or the ICWSM conference. Non-event content, of course, is prominent on Twitter and similar systems where people share various types of content such as personal updates, random thoughts and musings, opinions, and information [17].

As a challenge, non-event content also includes forms of Twitter activity that trigger substantial message volume over specific time periods [4], which is a common characteristic of event content. Examples of such non-event activity are Twitter-specific conversation topics or *memes* (e.g., using the hashtag #thingsparentssay), and *retweet* activities, characterized by a "storm" of retweets of popular Twitter users (e.g., an inspiring comment by Lady Gaga). Our goal is

to differentiate between messages about real-world events and non-event messages, where non-event messages include those for "trending" activities that are Twitter-centric but do not reflect any real-world occurrences.

We are now ready to define our event identification problem, as follows (Figure 1):

> Consider a time-ordered stream of Twitter messages M. At any point in time t, our goal is to identify real-world events and their associated Twitter messages present in M and published before time t. Furthermore, we assume an online setting for our problem, where we only have access to messages posted before time t.

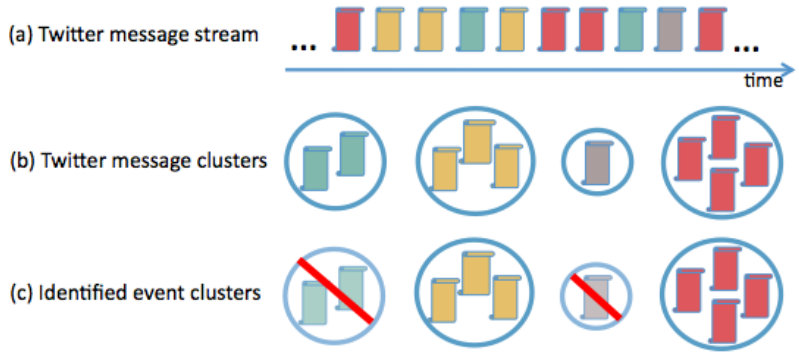

Fig. 1. Conceptual diagram: Twitter event identification.

4 Separating Event and Non-Event Content

We propose to address the event identification problem using an online clustering and filtering framework. We describe this framework in detail (Section 4.1), and then discuss the different types of features that we extract for clusters (Section 4.2), as well as the classification model that we use (Section 4.3) to separate event and non-event clusters.

4.1 Clustering and Classification Framework

We elected to use an incremental, online clustering algorithm in order to effectively cluster a stream of Twitter messages in real time. For such a task, we must choose a clustering algorithm that is scalable, and that does not require *a priori* knowledge of the number of clusters, since Twitter messages are constantly evolving and new events get added to the stream over time. Based on

these observations, we propose using an incremental clustering algorithm with a threshold parameter that is tuned empirically during a training phase. Incremental clustering has been shown to be an effective technique for event detection in textual news documents [2]. Such a clustering algorithm considers each message in turn, and determines a suitable cluster assignment based on the message's similarity to the existing clusters. Specifically, given a threshold τ, a similarity function σ, and message stream $M = m_1, \ldots, m_n$, the algorithm considers each message m_i in order and computes its similarity $\sigma(m_i, c_j)$ against each existing cluster c_j, for $j = 1, \ldots, k$. (Initially, $k = 0$.). If there is no cluster whose similarity to m_i is greater than τ, the algorithm creates a new cluster c_{k+1} for m_i. Otherwise, m_i is assigned to a cluster c_j with the maximum $\sigma(m_i, c_j)$.

For scalability, we use a centroid representation of each cluster, which consists of summary statistics of all messages in the cluster. The centroid of a cluster is the average weight of each term across all documents in the cluster. We represent each message as a *tf-idf* weight vector of its textual content, and use the cosine similarity metric, as defined by Kumaran and Allan [12], as the clustering similarity function σ. Based on our experiments on training data, we perform traditional text processing steps such as stop-word elimination and stemming, and also double the weight of hashtag terms as they are often indicative of the message content.

We have explored different threshold settings and other variations of this clustering algorithm, including a periodic second pass to handle fragmentation, which is a known drawback of this incremental clustering approach. However, the specific optimization of this clustering algorithm is beyond the scope of this paper and is the subject of ongoing and future work, as well as of other related papers [5, 21, 18]. Rather, the focus of this work is on techniques for separating event and non-event content using features of topically similar Twitter messages. Note that the features and techniques that we use for this purpose are general enough that they can be applied to any Twitter dataset that has been clustered or aggregated by topic, irrespective of the specific clustering algorithm used.

To identify all *event* clusters in the stream, we compute a variety of revealing features using statistics of the cluster messages (Section 4.2). Since the clusters constantly evolve over time, we must periodically update the features for each cluster and compute features of newly formed clusters. We subsequently proceed to invoke a classification model (Section 4.3) that, given a cluster's feature representation, decides whether or not the cluster, and its associated messages, contains event information. With the appropriate choice of classification model, we can also select the top events in the stream at any point in time, according to the clusters' probability of belonging to the event class.

4.2 Cluster-Level Event Features

We compute features of Twitter message clusters in order to reveal characteristics that may help detect clusters that are associated with events. While each of these features may not necessarily indicate event content in isolation, combining them with other revealing features in a principled way (e.g., using a trained

classifier) can help identify event clusters, as we will see. We examine several broad categories of features that describe different aspects of the clusters we wish to model. Specifically, we consider temporal, social, topical, and Twitter-centric features.

Temporal Features The volume of messages for an event e during the event's associated time T_e exhibits unique characteristics (see the definition of event in Section 3.2). To effectively identify events in our framework, a key challenge is to capture this temporal behavior with a set of descriptive features for our classifier. We design a set of temporal features to characterize the volume of frequent cluster terms (i.e., terms that appear frequently in the set of messages associated with a cluster) over time. These features capture any deviation from expected message volume for any frequent cluster term or a set of frequent cluster terms. Specifically, we aggregate the number of messages containing each term into hourly bins and define $M_{t,h}$ as the number of messages posted during hour h and containing term t, and M_h as the total number of messages posted during hour h.

For the n most frequent terms in the cluster, where n is determined empirically, we compute two types of features to reveal the trending behavior that is characteristic of events. First, we compute the deviation from expected volume for a term at the time when we compute the features (i.e., at the time when we invoke the classifier; see Section 4.3). This metric captures a single-point representation of trending behavior for each term. Second, we compute the quality of fit of an exponential function to the term's binned data leading up to the time when we invoke the classifier. The exponential fit captures the rate of increase in message volume over time. A good quality fit signifies a true exponential rise in related content, an indication of trending behavior [14].

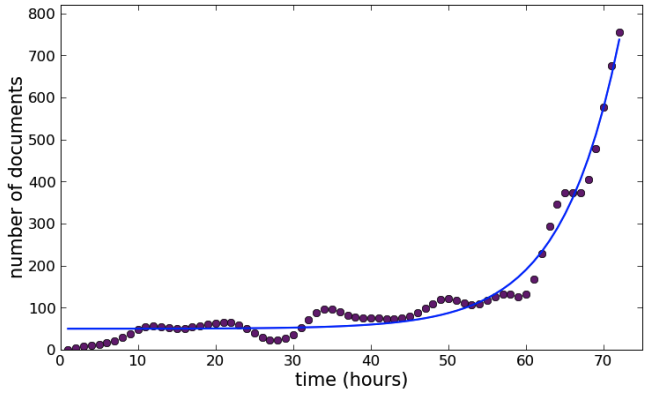

Fig. 2. Documents per hour with the term "valentine" for 72 hours prior to 2 p.m. on Valentine's Day.

We compute the expected number of messages for a term t at the end of hour h_0 by averaging the number of messages containing t in the preceding hours $(1, \ldots, h_0 - 1)$, weighted by the total number of messages at each hour to account for the varying volume of messages across different hours; formally, $\mu_{t,h_0} = \sum_{i=1}^{h_0-1} \frac{M_{t,i}}{M_i}/(h_0 - 1)$. Correspondingly, σ_{t,h_0} is the standard deviation of the number of messages containing t over the preceding hours. We define the deviation from expected message volume for term t at hour h_0 as $(\frac{M_{t,h_0}}{M_{h_0}} - \mu_{t,h_0})/\sigma_{t,h_0}$. The deviation from expected volume features in a cluster, then, include a set of deviation values for the most frequent terms, as well as an average value over all top terms. This value is generated by weighting the top terms by their relative support in the cluster messages (e.g., if terms t_1, t_2, and t_3 appeared in 300, 200, and 100 cluster messages, respectively, their weights would be 0.5, 0.33, and 0.17).

The second set of temporal features reflects the degree to which the volume of messages containing a term t exhibits an exponential growth in the hours leading up to h_0. We compute a histogram using $M_{t,i}$, where $i = (h_0 - 72), \ldots, (h_0 - 1)$; this models the volume of messages with the term for the 72 hours leading up to h_0. This method generally reflects the trending behavior in the social Web [14]. We use the least squares method to fit an exponential function to the histogram, smoothed using a moving average, and compute the R^2 statistic to measure the quality of the fit. Figure 2 shows an example of this exponential trending behavior for the term "valentine" around Valentine's Day, 2010.

Social Features We designed social features to capture the interaction of users in a cluster's messages. These interactions might be different between events, Twitter-centric activities, and other non-event messages [4]. As mentioned (Section 3.1), user interactions on Twitter include retweets, replies, and mentions. Our social features include the percentage of messages containing each of these types of user interaction out of all messages in a cluster.

To motivate the use of these features, consider the Twitter messages in Figure 3. Clusters that include a high percentage of retweets, especially of a single post by a popular Twitter user (e.g., Justin Bieber's message, retweeted over 100 times in Figure 3), may not contain real-world event information [4]. Similarly, a high percentage of cluster messages containing replies (e.g., Paris Hilton's reply in Figure 3) may indicate non-event content, since when people spread event information they tend to do so via general broadcast messages rather than individual conversations. On the other hand, many celebrities, politicians, companies, venues, and shows own Twitter accounts (e.g., Ashton Kutcher's show @FFLShow with guest @kurt13warner in Figure 3). Therefore, unlike retweets, a high percentage of Twitter mentions of one of these entities might imply that the cluster refers to an event, where the entity is an active participant or the subject of the event.

Topical Features Topical features describe the topical coherence of a cluster, based on a hypothesis that event clusters tend to revolve around a central

Fig. 3. Examples of social interaction on Twitter.

topic, whereas non-event clusters do not. Rather, non-event clusters often center around a few terms (e.g., "sleep," "work") that do not reflect a single theme (e.g., with some messages about sleep, others about work, and a few about sleeping at work). Messages in event clusters are likely to share more terms, as they identify key characteristics of the events they describe (e.g., "Couric," "Obama," and "interview" are common among messages describing Katie Couric's interview of President Obama).

To estimate this coherence of a cluster, we compute the average or median similarity of messages to the cluster centroid using the cosine similarity metric. Additionally, we compute the percentage of messages in the cluster containing the most frequent term, the second most frequent term, and so on. Finally, we look at how many of the most frequent terms are contained in at least $n\%$ of the messages in the cluster, for empirically determined values of n.

Twitter-Centric Features While the goal of our classifier is to distinguish between event and non-event data, we highlight the differences between non-event clusters that correspond to Twitter-centric activities, which are a specific class of non-event messages (Section 3.2), and the real-world event clusters that we wish to identify. As discussed above, Twitter-centric activities often exhibit characteristics that resemble real-world events, especially as captured by temporal features, which generally offer a strong signal for the presence of event content. To address this challenge, we design a set of features that target commonly occurring patterns in non-event clusters with Twitter-centric behavior.

Twitter-centric discussions often exhibit unique hashtag usage characteristics (e.g., #whenimolder tag indicating discussion on things Twitter users wish to do when they get older). We design features to capture these characteristics and differentiate the Twitter-centric activities from other non-event content and from real-world events. Specifically, we compute statistics relating to tag usage, including the percentage of cluster messages that contain tags, and the percentage of cluster messages that contain the most frequently used tag. A large value of the latter serves as an indication that the messages in the cluster revolve around a tagged conversation topic.

Importantly, we also determine if the most frequently used tag is a concatenation of multiple words. Multi-word tags are highly indicative of Twitter-centric discussions that do not correspond to real-world events (e.g., #firstdaterulez, #BadWrestlingNames). Unfortunately, identifying them is a challenging task since they often contain short-hand notations, acronyms, and slang that may be difficult to parse. Using a dictionary-based method for parsing the tags into several terms may be inefficient and difficult to implement due to the variety of potential terms that may be included in the tags. We have experimented with identifying these multi-word tags using such an approach with limited success. Instead, we design capitalization-based features to detect such multi-word tags: we observed that when more than one letter of a tag is capitalized by some users, and this capitalization is consistent among these users, it frequently indicates that a tag consists of multiple words. Since we do not rely on a dictionary, our approach can be applied to tweets in any language that uses capitalization rules.

4.3 Event Classification

Using the above features, we train an event classifier by applying standard machine learning techniques (see Section 5). This classifier predicts which clusters correspond to events at any point in time (i.e., at any point in the stream; see Section 3.2). Specifically, to identify event clusters at the end of hour h, we first compute the features of all clusters with respect to h, and then use the classification model with each cluster's feature representation to predict the probability that the cluster contains event information.

Due to the large volume of data on Twitter, it is possible that at any point in time the classifier may label many clusters as events. In an event browsing scenario, where users look for information on current events, it is essential to display a select subset of these identified event clusters. To that end, we are interested in the ability of our classifier to select the top events according to their probability of belonging to the event class, with respect to any point in the stream. Note that a temporal component is built into some of the features, and we recompute the features prior to classification, so the temporal relevance of the top selected clusters is inherently captured by our classifier.

We compare the results of our classifier against several baseline approaches next.

5 Experiments

We evaluated our event identification strategies on a large dataset of Twitter data. We describe this dataset and report the experimental settings (Section 5.1), and then turn to the results of our experiments (Section 5.2).

5.1 Experimental Settings

Data: Our dataset consists of over 2,600,000 Twitter messages posted during February 2010. We are interested in identifying events both with local and with

broad geographical interest. To ensure that our dataset substantially covers local events, we decided to collect messages posted by users of one specific location, namely, New York City (i.e., by Twitter users whose location, as entered by the users and shown on their profile, is in the New York City area)[3]. We chose this location as it consistently generated a high volume of tweets. While the location as reported by Twitter users is not always accurate, it does provide a reliable approximation [11]. Since we do not currently use location-based signals in our identification approach (a task that is reserved for future work), focusing on messages from a specific geo-location does not reduce the generality of our results. We collected these messages via a script, which continuously requested the most recent messages from the Twitter API. For each collected Twitter message, we record its textual content, the associated timestamp (i.e., the time at which the tweet was posted), and the username of the user who posted the tweet.

We cluster our dataset in an online fashion as described in Section 4.1. We use the data from the first week in February to calibrate statistics such as term frequency over time, which are needed to compute our temporal features. We then use the second week of February to train our event classifiers and baselines. Finally, we report our results on test data selected from the latter half of February (i.e., Weeks 3 and 4). **Annotations:** We use human annotators to label clusters for both the training and testing phases of our event identification experiments. These annotators were instructed to label each cluster according to four different categories (see Section 4.3): real-world event, Twitter-centric activity, other non-event, and ambiguous. To ease annotation, as a representation of each cluster, the annotators were shown the 10 most frequent terms in the cluster, along with their respective counts, and sample Twitter messages from the cluster. For clusters with more than one central theme (e.g., with top keywords "south," "park," "west," "sxsw," and "cartman," referring to either the "South Park" show or the "South by Southwest" festival), the annotators used the ambiguous label. Ambiguous clusters were not used for training, but were treated as non-events for testing.

For the training set, we randomly selected 504 clusters from the top-20 fastest-growing clusters according to hourly message volume at the end of each hour in the second week of February 2010. Each cluster was labeled by two annotators, and their agreement was measured using Cohen's kappa (κ=0.683), indicating substantial agreement. After removing 34 ambiguous clusters and dropping 96 clusters on which the annotators disagreed, we were left with 374 clusters.

For the test set, we used 300 clusters collected at the end of five different hours in the third and fourth weeks of February 2010. These five hours were sampled uniformly at random from five bins partitioned according to the volume of messages per hour over these two weeks. This sampling technique assures that we test our classifiers during hours with different volumes of messages.

[3] Note that events with broad geographical interest are also naturally captured in our dataset.

At the end of each hour we select the 20 fastest-growing clusters according to hourly volume, the top-20 clusters according to our classifier (Section 4.3), and 20 random clusters, for a total of 60 clusters per hour, or 100 clusters per method over the five hours. We used two human annotators to label each cluster and achieved substantial agreement (κ=0.83). We discuss our handling of annotator disagreements on the test set in the description of our evaluation.

Training Classifiers: We train a classifier to distinguish between real-world event and non-event clusters (*RW-Event*). We extracted cluster-level features for each cluster in the training set, as described in Section 4.2. We also used a few additional features that did not fall under the groups described in Section 4.2, such as the cluster size and average length of cluster tweets. We used the Weka toolkit [22] to train our event classifier. We first applied a resampling filter to balance the class distribution, which was skewed towards the non-event class, and then we trained and evaluated the classifier using 10-fold cross validation. We explored a variety of classifier types and selected support vector machines (specifically, Weka's sequential minimal optimization implementation) for *RW-Event*, as it yielded the best overall performance in exploratory tests over the training set. We also fit logistic regression models to the output of the support vector machine, to obtain probability estimates of the class assignment.

As a baseline, we use a strong text classification approach that identifies events based on the textual content of the messages in the cluster. Specifically, we trained a Naïve Bayes classifier (*NB-Text*) that treats all messages in a cluster as a single document, and uses the *tf-idf* weights of textual terms as features. This classifier, distinguishing between events and non-events, is similar to the one used by Sankaranarayanan et al. [21] as part of their approach for identifying news in Twitter messages. We train this Naïve Bayes classifier using Weka, with the same methodology described above.

Evaluation: We use our annotated test set of 100 randomly selected clusters to evaluate the performance of each classifier. For this, we use the macro-averaged F_1 metric [15]. This evaluation metric is widely used and is effective for evaluating classification results where it is desirable to assign an equal weight to the classifier's performance on each class. Here, macro-averaged F_1 is preferable to its alternative, micro-averaged F_1 [15], which weighs each instance equally, causing predictions on the larger non-event class to dominate the score. In this evaluation we omit test clusters on which our annotators disagree.

In addition to classification performance, we evaluate our *RW-Event* classifier's ability to identify events among a set of top clusters, ordered by their probability of belonging to the event class at the end of each hour. We refer to this task as "event surfacing." Since the number of clusters in the stream may be large, we only classify clusters that have over 100 messages. Similarly, we do not classify clusters that did not have newly added documents in the hour prior to the time when we invoke the classifier.

As a baseline for the event surfacing task, we consider the event thread selection approach presented by Petrović et al. [18], which selects the fastest-growing threads in a stream of Twitter messages and then re-ranks them based

on thread entropy and unique number of users. Preliminary experiments on our training data indicated that selecting clusters based on such re-ranking strategies (i.e., selecting clusters with the highest number of unique users and entropy above a threshold) yields similar results as selecting the fastest-growing clusters. Note that the re-ranking strategies were not used to select the top clusters, which is our goal, and optimizing the selection of fastest-growing clusters that have the highest number of unique users and low entropy is reserved for future work (in fact, similar features already exist in our models). In addition to the fastest-growing clusters baseline (*Fastest*), we compare our approach against a technique that selects clusters randomly (*Random*).

To evaluate the event surfacing task, we select two standard metrics, namely, *Precision@K* and *NDCG* [8], which capture the quality of ranked lists with focus on the top results. Precision@K simply reports the fraction of correctly identified events out of the top-K selected clusters, averaged over all hours. Precision@K is set-based and does not consider the relative rank of the clusters. An alternative metric that is sensitive to the rank of the events in the top selected clusters is the normalized discounted cumulative gain (NDCG) metric. We use the binary version of NDCG [8], to measure how well our approach ranks the top events relative to their ideal ranking. To handle annotator disagreements in this scenario, where we need to examine ordered lists, removing the disagreements from the evaluation is not desirable given the evaluation metrics used. Instead, we penalize the *RW-Event* classifier if *either* annotator disagreed with our classifier's prediction, but only penalize the baselines if *both* annotators disagreed with their predicted label. We thus give the "benefit of the doubt" to the baselines, hence making our results more robust.

5.2 Experimental Results

We begin by examining the performance of our *RW-Event* classifier against the *NB-Text* baseline classifier on the training and test sets. The performance on the training set reflects the accuracy of each classifier computed using 10-fold cross-validation. The test performance measures how well each classification model predicts on the test set of 100 randomly selected clusters.

Table 1 shows the F_1 scores of the classifiers on both the training and test sets. As we can see, the *RW-Event* classifier outperformed *NB-Text* over both training and test sets, showing that it is overall more effective in predicting whether or not our clusters contain real-world event information. A deeper examination of our results revealed that the *NB-Text* classifier was especially weak at classifying event clusters, accurately predicting only 25% of event clusters on the test set. A sample of event clusters identified by *RW-Event*, and their most frequent terms, are presented in Table 2.

The next set of results describes how well our *RW-Event* classifier performs for the "event surfacing" task. Recall that the goal of this task is to identify the top events in the stream per hour. We report Precision@K (Figure 4) and NDCG@K (Figure 5) scores for varying K, averaged over the five hours selected

Classifier	Validation	Test
NB-Text	0.785	**0.702**
RW-Event	0.849	**0.837**

Table 1. F_1 score of our classifiers on training and test sets.

Description	Terms
Senator Bayh's retirement	bayh, evan, senate, congress, retire
Westminster Dog Show	westminster, dog, show, club
Obama & Dalai Lama meet	lama, dalai, meet, obama, china
NYC Toy Fair	toyfairny, starwars, hasbro, lego
Marc Jacobs Fashion Show	jacobs, marc, nyfw, show, fashion

Table 2. Sample events identified by the *RW-Event* classifier.

for the test set. We compared the results of *RW-Event* to two baselines: *Fastest* and *Random* (Section 5.1).

Not surprisingly, the proportion of events identified by the *Random* technique is very low, as most data on Twitter does not contain event information. The proportion of events identified by the *Fastest* technique was higher than that of *Random*. The *RW-Event* classifier performed well across the board, better than both baselines according to both precision and NDCG.

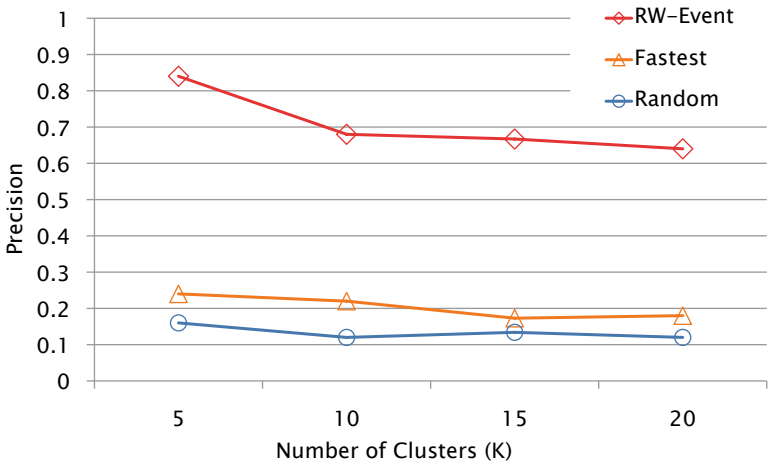

Fig. 4. Precision @ K for our classifier and baselines.

Examining the mistakes made by the *RW-Event* classifier, the most prominent misclassification occurs in cases where a Twitter user (usually a company or service) posts messages on a broad topic (e.g., job listings with tags such

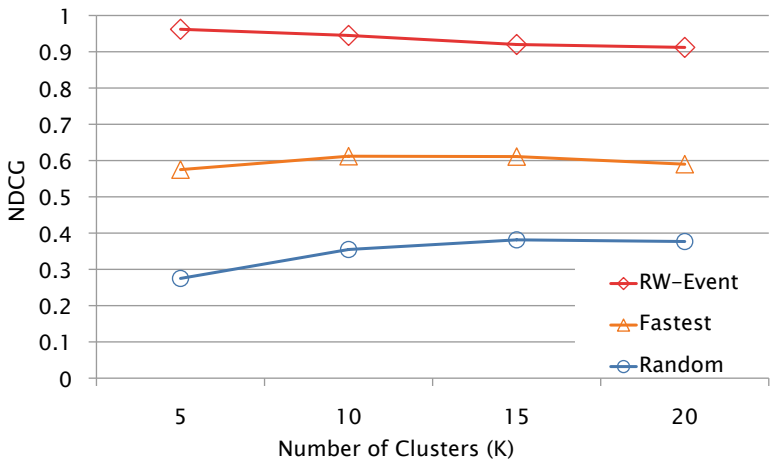

Fig. 5. NDCG @ K for our classifier and baselines.

as #jobs, #nycjobs) using multiple Twitter accounts and a similar message "template," complete with hashtags. A possible reason for this behavior is that features of our model such as the number of messages from the top author were not adequately captured in the training process. Since we selected training data by sampling from the fastest-growing clusters per hour, many of our training examples did not exhibit this behavior and, therefore, we were not able to properly model it. We plan to explore this issue further in future work.

6 Conclusions

We presented an end-to-end approach for identifying real-world event content on Twitter. This work provides the first step in a series of tools that improve on the generic analysis of "trending topics."

Our techniques for event identification offer a significant improvement over baseline and existing approaches, showing that we can identify *real-world* event content in a large-scale stream of Twitter data. We thus help unveil important information from, and about, real-world events as they are reflected through the eyes of hundreds of millions of users of Twitter and similar social media sites.

In future work, we aim to reason even more finely about different types of events that are reflected in Twitter data. For example, real-world events may include news events [21], local small-scale community events, breaking and emergency events [20], and so forth. Given a robust classification of events, extending the work described here, we can improve prioritization, ranking, and filtering of extracted content on Twitter and similar systems, as well as provide more targeted and specialized content visualization.

7 Acknowledgments

This material is based on work supported by NSF Grants IIS-0811038, IIS-1017845, and IIS-1017389, and by two Google Research Awards. In accordance with Columbia Univ. reporting requirements, Prof. Gravano acknowledges ownership of Google stock as of the writing of this article.

References

1. James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publisher, 2002.
2. James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval*, 1998.
3. Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, 2010.
4. Hila Becker, Mor Naaman, and Luis Gravano. Hip and trendy: Characterizing emerging trends on Twitter. *Journal of the American Society for Information Science and Technology*, 2011. To appear.
5. Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.
6. Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on Twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, 2010.
7. Ling Chen and Abhishek Roy. Event detection from Flickr data through wavelet-based spatial analysis. In *Proceedings of the 2009 ACM CIKM International Conference on Information and Knowledge Management*, 2009.
8. W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 2009.
9. Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, 2010.
10. Events, 2002. In Stanford Encyclopedia of Philosophy. Retrieved June 2nd, 2010 from http://plato.stanford.edu/entries/events/.
11. Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from Justin Bieber's heart: The dynamics of the "location" field in user profiles. In *Proceedings of the 29th ACM Conference on Human Factors in Computing Systems*, 2011.
12. Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval*, 2004.
13. Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th International World Wide Web Conference*, 2010.
14. Jure Leskovec, Lars Backstrom, and Jon M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.
15. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge Univ. Press, 2008.

16. Michael Mathioudakis and Nick Koudas. Twittermonitor: Trend detection over the Twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, 2010.
17. Mor Naaman, Jeffrey Boase, and Chih-Hui Lai. Is it really about me?: Message content in social awareness streams. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, 2010.
18. Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.
19. Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.
20. Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International World Wide Web Conference*, 2010.
21. Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: News in tweets. In *Proceedings of the 17th ACM International Conference on Advances in Geographic Information Systems*, 2009.
22. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.
23. Sarita Yardi and danah boyd. Tweeting from the town square: Measuring geographic local networks. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.
24. Jeffrey M. Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127, 2001.
25. Qiankun Zhao, Prasenjit Mitra, and Bi Chen. Temporal and information flow based event detection from social text streams. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, 2007.