Feature Extraction and Analysis for Identifying Disruptive Events from Social Media

Nasser Alsaedi and Pete Burnap
Cardiff School of Computer Science & Informatics
Cardiff University, Cardiff, UK
{N.M.Alsaedi, P.Burnap}@cs.cardiff.ac.uk

***Abstract*—Disruptive event identification is a concept that is crucial to ensuring public safety regarding large-scale events. Recent work on detecting events from social media shows that although these platforms are used for social purposes, they have been emerging as important source of information. Twitter, as a form of social media, is a popular micro-blogging web application serving hundreds of millions of users. User-generated content can be exploited as a rich source of information for identifying 'real-world' *disruptive events* – events that threaten social safety and security, or could cause disruption to social order. In this paper, we present an in-depth comparison of two types of feature that could be useful for identifying disruptive events: temporal and textual features. On the basis of these features, we investigate the dynamics of event/topic identification over time. We make several interesting observations: first, disruptive events are identifiable regardless of the "influence of the user" discussing them, and over a variety of topics. Second, temporal features play a central role in event detection and hence should not be disregarded or ignored. Third, textual features can be used to improve the overall performance of the event detection. We believe that these findings provide new insights for gathering information around real-world events, in particular for detecting disruptive events.**

Keywords—*Data Mining, Event Detection, Feature Selection.*

I. INTRODUCTION

In recent years micro blogging, as one of the social media, has proven to be a fast emerging tool for expressing opinions, broadcasting news, and interaction between people. One of the most popular examples is Twitter, which allows users to publish short tweets (messages limited to 140 characters) about a wide variety of topics, and is widely used to discuss real-world events. Event identification is a concept that is crucial in event management, intelligence gathering, and decision-making.

People tend to comment on real-world events (both local and global), when a topic suddenly captures their attention. For example, sport events, adverse weather updates, terror attacks, etc. Identifying events and specifically *disruptive events* – sub-events that threaten social safety and security, or could cause disruption to social order – from the social media presents several challenges. A key challenge is to distinguishing ambiguous tweets about everyday mundane activity, from events of public interest, in particular those that might impact on public safety. If this could be done then those with responsibility for managing and ensuring public safety would be able to use this information to better manage a potentially harmful situation. Understanding the features of tweets that report disruptive events is therefore the key motivation behind this work.

The dynamic nature of such events leads to a diverse set of linguistic features. This is compounded by the fact that each post is short in length, which means that they offer only limited content for analysis. A second key issue is the velocity at which streamed tweets arrive. Twitter is used to post up to 400 million tweets per day, so event detection algorithms need to incorporate the minimal number of operations in order to reduce computational overheads when analyzing real-time streams.

One way to optimize the identification of the patterns and signals that indicate an event is to undertake feature selection experiments. Because not all features are expected to lead to better system performance or contribute equally towards improved machine classification and/or clustering accuracy, we seek to evaluate the effectiveness of a range of features for identifying events, and, further, features that would distinguish 'normal' events from *disruptive* events. These features may be divided as follows:

- Temporal features: related to the "speed" of diffusion over time by highlighting the "quality" of tweets created by users in different time frames;
- Textual features: which are representative of the text content published to Twitter.

In [1] we proposed a probabilistic framework to distinguish events from non-events, and some early results in identifying disruptive events. In this paper we focus specifically on optimizing feature selection to increase the performance results of the event classification algorithm. We improve the quality of events detected using textual and time dimensions to identify different types of events (disruptive events). The contributions of this paper are:

1. An improved model for feature selection that is suitable for microblog data such as Twitter;

ASONAM '15, August 25-28, 2015, Paris, France
 ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2808867

2. in-depth temporal analysis of event-related information such as lexical, social opinion and social interaction features;
3. experimental identification of features that distinguish disruptive events from other events.

The rest of the paper is organized as follows. In section 2, we summarise related research on feature selection. In section 3, we briefly discuss the main elements of our proposed framework. In section 4 we discuss the method used for feature selection and details of the temporal and textual features. Section 5 presents our experiments and discusses the results. We conclude by highlighting some future directions for research in section 6.

II. RELATED WORK

Event detection, monitoring and tracking have turned into an active research area. In this paper we do not focus on different techniques used for detecting events using social media. Atefeh and Khreich [24] provided a comprehensive survey of techniques for event detection using Twitter. Instead, we focus on feature selection experimentation to determine the optimal selection of features to enhance event detection accuracy. In this section, we summarise the current research on temporal and textual features and the way in which it has been applied to data mining tasks.

Temporal features

The emerging interest in temporal aspects in the context of Information Retrieval (IR) is demonstrated by the recent TREC (Text REtrieval Conference) Knowledge-Base-Acceleration (KBA) [2], TAC (Text Analysis Conference) Knowledge-Base-Population (KBP) [3] and the TREC Temporal Summarization (TS) [5, 12] challenges. Hence, many researchers have focused their attention on analyzing and exploiting temporal information to develop time-specific information retrieval and exploration applications [12]. For instance, Kanhabua et al. [29] proposed three different methods to determine the time of queries using temporal language models, which are built based on the New York Times news collection, where documents are explicitly time-stamped according to the document creation time.

Radinsky et al. [23] proposed a novel semantic relatedness model, Temporal Semantic Analysis (TSA), which constructs a time series for each word of the New York Times collection on the assumption that two words are highly related if their time series are related as well. They incorporate temporal information into models which can be used for studying language evolution over time. The Temporal Summarization (TS) task aims to return short relevant updates about an event from a time-ordered stream of documents. A web-based system is presented in [5] for generating temporal summarization of real-world events such as political conflicts and natural catastrophes that are mirrored by article updates in Wikipedia. They used Wikipedia content for topic detection and tracking (TDT) and Temporal Clustering for Temporal Summarization (TS) to generate meaningful updates and summaries about events in a timeline.

Within the social media domain, posts generally come with a creation time-stamp, which can be utilized for topic detection and tracking (TDT) as demonstrated in [15], which analysed the evolution of stories and topics over time. Similarly, Gabrilovich et al. [16] studied the dynamics of information novelty in some evolving news stories. There has also been much work on the community structure of the blogosphere. The authors of [13] showed that the prediction of information cascades is feasible and the relative growth of a cascade becomes more predictable as more "reshares" are observed over time – hence, these temporal features are key predictors. Rather than attempt to predict cascades, Elsas and Dumais [17] studied the dynamics of document content change with applications to the ranking of documents on the basis of their temporal characteristics. In this paper we enhance previous work by studying the significance of content features in identifying disruptive events across a number of time windows.

Textual features

Textual features can be used as individual features (e.g. n-grams), but many studies have combined them to optimise the solution to data mining challenges, such as information diffusion [8, 13], opinion mining [9, 14], spam and spammer detection [18], and identifying the most knowledgeable posts and famous users [4, 6, 7, 8, 19].

Using the topic model in [7], a set of raw features (number of original tweets, number of retweets, and number of mentions) is used for identifying the most influential Twitter users. Agarwal et al. [14] investigated two kinds of model: a feature based model and a tree kernel based model for the purpose of sentiment classification. They demonstrated that both models outperformed the unigram baseline model which was previously shown to work well for Twitter sentiment analysis. Hashtag popularity was considered by Ma, Sun, and Cong in [8]. They demonstrated that contextual features (such as the number of users, number of tweets, retweet ratio …) are more effective than content features (such as tweets containing URL, the ratio of neutral, positive, and negative tweets …) in predicting hashtag popularity.

III. EVENT DETECTION FRAMEWORK

Identifying "events" and "disruptive events" from social media streams requires us to define these terms first. Wenwen-Dou in [20] defines an event as

"*An occurrence causing change in the volume of text data that discusses the associated topic at a specific time.*"

Hence, a bursty spike of terms or words at a point in time can characterize an event. In [1], we defined a disruptive event in the context of the social media as:

"An event that interferes in the achieving of the objective of an event or interrupts ordinary event routine. It may occur over the course of one or several days, causing disorder,

destabilizing securities and may result in displacement or discontinuity."

In our work, we hypothesize that a disruptive event can be characterized by temporal and textual features. Disruptive events can be small-scale events, such as traffic accidents, or can be wider-reaching, national or even international levels. For example, if a factory is likely to shut down due to a demonstration or a major fire, related companies may find themselves involved or even contact their customers in order to prevent unexpected losses or long delays. Therefore, monitoring social media and identifying anomalies over time allow organizations or even governments to react promptly and effectively before they can escalate and become damaging to the wider society. This is the key motivation for distinguishing disruptive events from other types.

As we receive high volume of tweets per day, human monitoring is impractical. Figure 1 shows our proposed framework, which enables us to automatically identify meaningful events from Twitter. The method is based on collecting a series of data for a given location over predefined time frame windows. The five-step framework consists of data collection, pre-processing, classification, on-line clustering and summarization. (See [1] for full details of each step and the framework evaluation.)

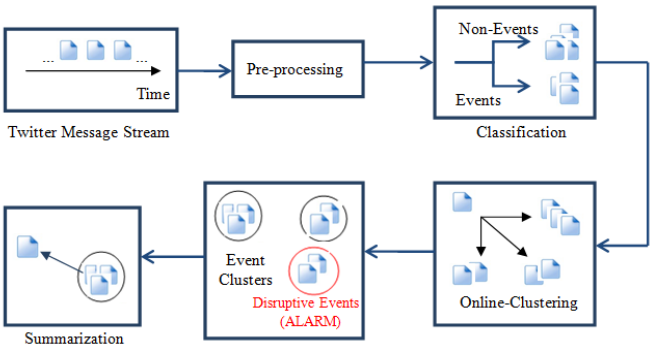

Fig. 1. Twitter Stream Event Detection Framework

IV. FEATURE SELECTION

Feature selection is a fundamental problem in mining large data sets. The problem is not limited to the total processing time but involves reducing the dimensionality to achieve better generalization. In this work, we chose to implement an improved version of the unsupervised feature selection presented in [25]: first, it resolves the issue of the high-computational complexity involved in searching large data sets. Second, the computation time is reasonably acceptable even for large data sets where other algorithms perform well only with medium sized data sets. Third, the unsupervised feature selection results are among the best clustering performances of real-world data sets.

In fact, not all features are expected to improve the system's performance or lead to more accurate discrimination of the clustering algorithm. Indeed, for many reasons the inclusion of some features could be detrimental to system performance; for example:

- The irrelevant input features induce greater computational cost. With more features, the computational cost for predictions increase polynomially [10,11]
- The irrelevant input features could lead to overfitting [22].
- The training algorithm with multiple features could result in some scalability issues [1, 11, 22], as was noticed empirically.

Let the original number of features be D and the original feature set be $O = \{F_i, i = 1, \dots, D\}$. We represent the dissimilarity between features F_i and F_j by $S(F_i, F_j)$. The higher the value of S, the more dissimilar the features. Let r_i^k represent the dissimilarity between feature F_i and its kth nearest-neighbor feature in R, where R is the reduced feature subset.

The dissimilarity between two features $S(F_i, F_j)$ is calculated by the Maximal Information Compression Index (MICI) which was proposed by Mitra et al. in [25]. The MICI is a well-known index for measuring dissimilarity between features and it has been applied in many pattern recognition and data mining tasks. The Maximal Information Compression Index is defined as:

$$\lambda(x, y) = \left[a - \sqrt{a^2 - 4b(1 - \rho(x, y)^2)}\right]/2$$

where $a = var(x) + var(y)$

$b = var(x).var(y)$

The *correlation coefficient* is defined as $(x, y) = \frac{cov(x)}{\sqrt{b}}$, var() denotes the variance of a variable, and cov() the covariance between two variables.

Algorithm 1. Feature Selection Algorithm

Step 1: Choose an initial value of $k \leq D - 1$. Initialize the reduced feature subset R to the original feature set O.

Step 2: For each feature $F_i \in R$, compute r_i^k

Step 3: Find feature $F_{i'}$ for which $r_{i'}^k$ is minimum.
Retain this feature in R and *discard* k nearest features of $F_{i'}$.
Let $\varepsilon = r_{i'}^k$

Step 4: If $k >$cardinality$(R) - 1$: $k =$ cardinality$(R) - 1$

Step 5: If $k = 1$: **Go to Step 8**.

Step 6: **While** $r_{i'}^k > \varepsilon$ **do**:
a) $k = k - 1$, $r_{i'}^k = \inf_{F_i \in R} r_i^k$
b) **If** $k = 1$: **Go to Step 8**.
End While

Step 7: **Go to step 2**.

Step 8: Return feature set R as the reduced feature set.

A. Temporal Features

Temporal features are important factors that have been overlooked in many event detection studies via the social media. The volume of tweets and the continually updated commentary around an event suggest that informative tweets from several hours ago may not be as important as new tweets [21]. For this reason we identify the most frequent terms in the cluster across a range of time windows. In our experiments we use a range of time windows to improve the efficiency of the event clustering system in terms of accuracy and total running time.

The time windows are related to the "speed" of diffusion over time. This has been shown to be an important feature in predicting thread length on Facebook [4], a primary mechanism in predicting popularity in Twitter [8, 22], and as the most important factor in influencing a cascade through the network [13].

B. Textual Features

There are two main tasks in this paper regarding textual features: first, we analyse different textual features in order to select the best contributors to the task of event detection. Second, we focus on ranking features with performance measures in mind and remove irrelevant features that might introduce greater computational cost. First we introduce these features in detail.

❖ Near-Duplicate measure

The average content similarity over all pairs of tweets posted in a (1-hour time slot) cluster is calculated using:

$$\sum_{a,b \in set\ of\ pairs\ in\ tweets} \frac{similarity\ (a,b)}{|set\ of\ pairs\ in\ tweets|}$$

where the content similarity is computed using the cosine similarity over words from tweet a, b vector representation $\overrightarrow{V}(a), \overrightarrow{V}(b)$ of the tweet content:

$$similarity\ (a,b) = \frac{\overrightarrow{V}(a).\overrightarrow{V}(b)}{|\overrightarrow{V}(a)||\overrightarrow{V}(b)|}$$

If the two tweets have a very high similarity, we assume that one of them is a near-duplicate of the other. The original tweet is considered as the first tweet in a particular time frame and/or the shortest tweet in length. Even though duplicates are less likely to provide additional information about an event, several users independently witnessing an event and tweeting about it would effectively increase the confidence level of an event.

❖ Retweet ratio

Retweeting represents the influence of a tweet beyond the one-to-one interaction domain. Popular tweets can propagate multiple hops away from the source as they are retweeted throughout the network [7]. Hence, the number of retweets is an indication of popularity. Furthermore, retweeting in a social network can serve as a powerful tool to reinforce a message when not only one but a group of users repeats the same message [7, 8]. Therefore, the retweet ratio indicates the tweets surrounding an event in which users agree with the message or wish to spread the information (warning, advice, evidence etc.) with other users. The retweet ratio has been implemented to detect events and to estimate rumors in the social media stream [26]. We calculate this attribute by normalizing the number of times a tweet appears in a timeframe to the total number of tweets in that timeframe.

❖ Mention ratio

A mention is a mechanism used in Twitter to reply to users, engage others or to join a conversation in a form of (@username). A user can mention one or more users anywhere in the body of the post. Hence, we calculate the number of mentions (@) relative to the number of tweets in a cluster. Ordinary users show great passion for celebrities and as a result the most mentioned users are celebrities who are sometimes mentioned by users who do not necessarily read their posts [7, 13]. Regarding event reporting, users tend to mention journalists, politicians and official accounts such as news agencies or government official accounts to drive their attention about an event or to add more credibility to their event-related posts.

❖ Hashtag ratio

Hashtags are an important feature of social networking sites and can be inserted anywhere within a message. Some Hashtags indicate their posted messages (#bbcF1) and others are dedicated originally to events such as (#abudhabigp). In addition, topic related hashtags are used as an information seeking index on Twitter to search Twitter for more tweets belonging to the same topic. The use of hashtags became a coordinating mechanism for disruptive-related activity on Twitter [1, 27]. The Hashtag ratio is the ratio of tweets containing hashtag over the total number of tweets in that timeframe.

❖ Link or Url ratio

As Twitter is limited to 140 characters per message it is common in the Twitter community to include links when tweeting to share additional information or for referencing. Clusters that have tweets with links from popular websites (news agencies or government sites) may boost the level of confidence in that information and hence encourage more adoption to such tweets and clusters. Additionally, the co-occurrence of URLs in a cluster confirms that these tweets refer to the same event and improves the level of confidence in the event. This attribute is calculated by the fraction of tweets with URL to the total number of tweets in a timeframe.

❖ Tweet sentiment

Users post real-time messages in microblogging websites giving their opinions on a variety of topics. They may discuss news, complain about services and express positive or negative sentiment about products [9, 14]. In fact, companies manufacturing such products have developed techniques to analyze these posts to get a sense of the sentiment toward their products [14]. Here, we investigate the role of sentiment strength in reporting disruptive events. We study whether sentiment polarity in posts (0 indicates neutral, 1 indicates positive or negative

sentiment) is significant features when reporting events. Subsequently, we investigate the influence of positive, negative and neutral sentiment on identifying disruptive events, given that negative sentiment tweets are more likely to be retweeted as shown in [6, 8, 9].

To calculate sentiment we use a semantic classifier based on the SentiStrength model in [9].The SentiStrength algorithm is suitable because it is designed for short informal text with abbreviations and slang. Moreover, it combines a lexicon-based model with a set of additional linguistic rules for spelling correction, negations, booster words (e.g., very), emoticons, and other factors. In our classifier we did not assign a sentiment score to each tweet as the SentiStrength model does because we are interested in studying the effect of average positive or negative sentiment polarity in tweets with respect to events.

❖ Dictionary-based feature

One of the main objectives of this work is the ability to automatically detect messages that contain precise information about disruptive events such as a labor strike or a fire. To enrich such rare event identification, present tense verbs, popular event nouns and adjectives that describe events as they take place are considered typical features. This bag of words model uses a dictionary of trigger words to detect and characterize disruptive events which are manually labeled by experts from several management departments: traffic control, crisis, emergency departments, and others.

Examples of present verbs: witness, notice, observe, participate, engage, listen etc.

Examples of event nouns: breaking news, update, situation, delay etc.

Examples of event adjectives: urgent, live, latest, severe, horrifying etc.

V. EXPERIMENTAL EVALUATION

We now describe the experiments that we conducted to test the accuracy of event classification using the features described in the previous section, and present the results of the studies of various features performed on a real-world dataset held out from the training phase.

A. Experimental Settings

Dataset: Our dataset consisted of 1,698,517 tweets, and was collected from 15 October 2013 to 05 November 2013 using Twitter's Streaming API. We have chosen to study a major sporting event - the Formula 1 Motor Racing Grand Prix - due to its global interest. The event was hosted in Abu Dhabi between 1st and 4th November 2013.

Framework Evaluation: We sampled 85,000 event-related tweets from the study dataset using the Naïve Bayes event classifier as presented in [1], which we used to train, test and evaluate our clustering algorithm. We used the first 15 days of data (from 15 Oct until 29 Oct) to train the clustering algorithm and to tune the thresholds using the validation set. Then we tested the clustering algorithm on unseen data from the 6 days between 30 Oct and 4 Nov. Threshold values were varied from 0.10 to 0.90 at graded increments of 0.05% with a total of 17 tests, in order to find the best cut-off of τ =0.45 (63 character difference). Figure 2 illustrates the F-measure scores for different thresholds where the best performing threshold τ =0.45 seems to be reasonable because it allows some similarity between posts but does not allow them to be near-identical. In order to evaluate the clustering performance we employed three human annotators to manually label 800 clusters where the most highly retweeted post represented that cluster and the surrounding tweets were assumed also to represent the event. The task of the annotators was to choose one of the eight different categories: politics, finance, sport, entertainment, technology, culture, disruptive event and others (if an event does not fall into any of the other predefined categories). The agreement between annotators was calculated using Cohen's kappa (K=0.794) which indicates an acceptable level of agreement. We used only **635 clusters** which all annotators agreed on as the **gold standard**. The framework was able to achieve an average F-measure of 80.85.

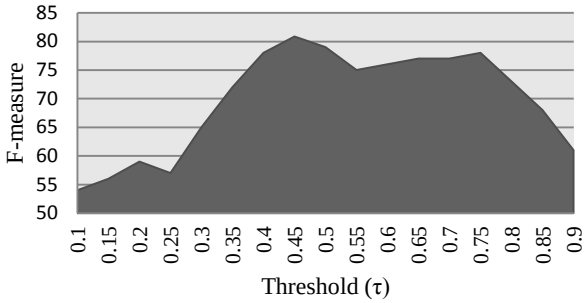

Fig. 2. F-measure of online clustering over different thresholds

B. Evaluation Matrix

To measure the effectiveness of the classifiers based on our proposed features, we used the standard classification metrics of precision, recall, accuracy, and F1 measure. Precision is a measure of false positives. Recall is a measure of false negatives. The F-measure is a harmonized mean of precision and recall. Accuracy is the proportion of correctly classified tweets to the total number of tweets. A false positive arises when the outcome is incorrectly predicted as X class when it is actually Y class. A true positive is when actual X class events are correctly predicted as X class events.

$$\text{Precision(P)} = \frac{tp}{tp+fp} \qquad \text{Recall(R)} = \frac{tp}{tp+fn}$$

$$\text{F} - \text{measure} = \frac{2\times P\times R}{P+R} \qquad \text{Accuracy} = \frac{tp+tn}{tp+fp+fn+tn}$$

The discrimination power between different proposed features can be measured by generating a Receiver Operating Characteristics (ROC) curve. ROC graphs are commonly used in machine learning and data mining research, as well as in medical decision making for organizing classifiers and visualizing various features [28]. ROC curves plot false positive rates on the horizontal axis and true positive rates on the vertical axis for varying thresholds. The closer the ROC curve is to the upper left corner, the higher is the overall accuracy. The coordinate

(0, 1) represents 100% sensitivity (no false negatives) and 100% specificity (no false positives).

C. Experiment 1

In the first set of experiments we study each feature individually. We use accuracy and running time to select the best temporal setting. However, we use feature selection method (outlined in Algorithm 1) to eliminate textual features.

Temporal features

Here we analyse the efficiency of the proposed temporal features in terms of the *event prediction accuracy* (A) and the *total running time* (RT). We calculate A (Figure 3) and RT (Figure 4) for a range of time windows; 1 minute, 30 minutes, 1 hour, 3 hours, 6 hours, 12 hours and 24 hours. To attain the best value for temporal features, we had to look at the following optimisation problem:

$$\{Clustering\ Accuracy - k \cdot Running\ Time\}.$$

where k is the threshold which maximizes the criterion.

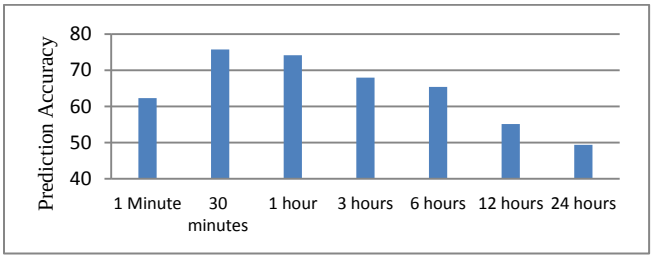

Fig. 3. Accuracy (A) obtained using various temporal settings

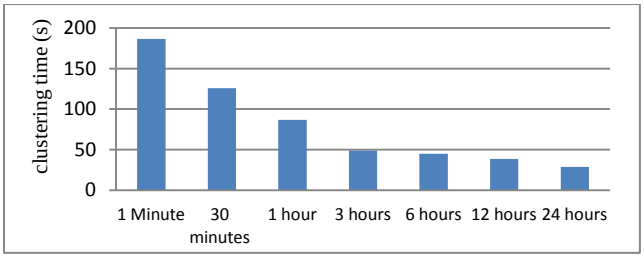

Fig. 4. Efficiency comparison with various temporal granularities (s)

The results presented in Figures 3 and 4 show that the 1–hour time window requires much less computational clustering time than for the 1 minute and 30 minute windows, while producing the second best level of accuracy after the 30 minute window. This suggests that tweets published recently are better predictors of events than older tweets, but also that a lead-in time is required (since 1 minute is too short to provide the same level of accuracy).

Clustering the tweets every hour provides a small reduction in the clustering accuracy but significantly reduces the computational processing requirements; therefore for the remaining experiments we set the time window for clusters to 1 hour. We find that reporting disruptive events are more likely in 1 hour than are other events. For example: reporting a rare event such as a car accident is more probable in a 1-hour time frame and so is the reporting of bigger disruptive events, such as natural disasters where people are posting an instantaneous reaction. However, reporting general events such as political or financial events is more likely to take several hours or even days.

Our approach is linear with respect to both the cardinality and dimensionality, i.e. $O(ndk)$; thus its scalability to a large and high dimensional dataset is applicable where n is the the number of entities (posts) to be clustered, d is number of dimensions and k is the number of clusters.

Textual features

Here we investigate the discriminative power of each individual feature in classifying disruptive events in order to show the robustness of each feature individually so the least discriminative features can be removed to reduce the computational workload required to compute the results. The results are shown in Figure 5 and Table 1. Figure 5 shows the ROC curve for each feature and Table 1 presents the performance results according to the F-measure and the difference between the F-measure of each feature and of the temporal feature which is selected to be the baseline for this experiment.

The near-Duplicate measure, Favorite ratio and Sentiment ratio are the least discriminative features, which would suggest that they appear in all the different types of event, not only in disruptive ones. But the Dictionary-based model, Retweet ratio and Hashtag ratio are the most discriminative, suggesting that references to present time and references to descriptive terms (e.g. live, breaking etc.) are good discriminators. The retweet ratio suggests that other Twitter users pick up on event commentaries and propagate them more often through the network than non-event tweets. Linking content features such as Hashtags and URLs is also very predictive of events, suggesting that tweets reporting events provide evidence or further information (via URL), or are bound to an event and made more discoverable via a self-defined topic discriminator in the form of a Hashtag.

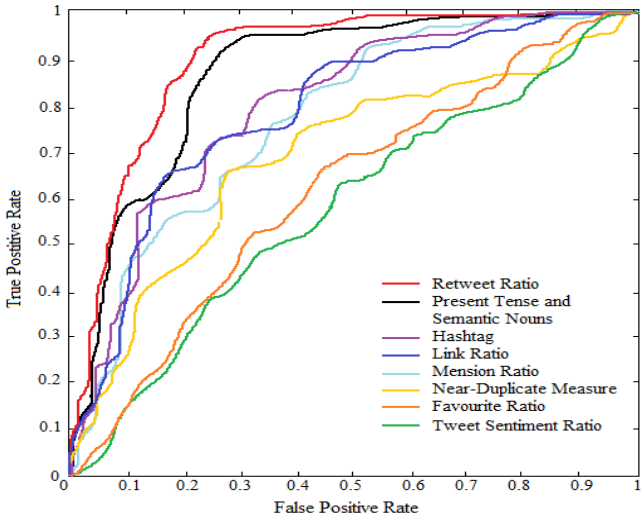

Fig. 5. ROC curves of the various proposed features

TABLE 1. COMPARISON OF THE PERFORMANCE USING VARIOUS TEXTUAL FEATURES.

Model	F-measure	F-measure Diff
Baseline (Temporal)	74.14	-
Near-Duplicate measure	74.69	0.55
Retweet ratio	77.57	3.43
Mention ratio	75.73	1.59
Hashtag ratio	76.13	2.99
Link or Url ratio	76.81	2.67
Favorite ratio	74.16	0.02
Tweet sentiment polarity	73.63	-0.51
Dictionary-based feature	77.43	3.29

Some of the above features such as the Retweet ratio are common to all events, including disruptive events, but some features such as the Dictionary-based feature and Mention ratio are more related to disruptive events. Combining textual features together (for example: the Retweet ratio, Hashtag and Dictionary-based feature) makes them more prone to distinguish disruptive events. Overall, all proposed features have a positive improvement over the baseline features except for tweet polarity, which is investigated in more detail in the next section.

Tweet sentiment

From the previous experiment we found that the polarity of tweets has a negative impact on classifier accuracy when reporting events. The goal of this experiment is to further examine whether positive, neutral or negative sentiment tweets have an effect in regard to the reporting of events. We re-ran the experiments, this time distinguishing between positive, negative and neutral sentiment.

The main observation made from Table 2 is that tweets with negative sentiment lead to a better F-measure than other sentiment measures. One possible reason is that tweets with negative sentiment are more likely to be retweeted, as shown in [6, 8, 9], which we have seen in our previous experiment is the most predictive feature in tweets about events. Therefore, negative tweet sentiment has a high adoption rate regarding disruptive event tweets. However, positive sentiment and neutral sentiment as predictive features did not significantly improve event detection results. Reporting disruptive events usually involves negative terms and sentiment whereas events in general can be positive, negative or neutral.

Table 2. F-measures for Positive, Neural and Negative Sentiment models, which clearly shows that the negative model outperforms others by at least 1.43%

Model	F-measure	F-measure Diff.
Positive sentiment ratio	74.27	0.13
Neutral sentiment	74.40	0.26
Negative sentiment ratio	75.83	**1.69**

Limitations in using sentiment as a discriminative predictive are twofold: first, the length constraint of a tweet limits the amount of sentiment that can be expressed; second, detecting sentiment polarity is a difficult task due to sarcasm.

Ranking top textual features

After investigating each feature individually, and further probing the tweet sentiment, we discarded features with less than a 1.60 improvement over the temporal baseline, to reduce the complexity of the clustering processes and therefore the computational overhead. Only the most predictive features were used to identify the disruptive events tweets. The ranking of the most influential textual features is presented in Table 3.

TABLE 3. THE MOST EFFECTIVE FEATURES (ABOVE 1.60 DIFFERENCES)

Rank	Feature	F-measure Diff
1	Retweet ratio	3.43
2	Dictionary-based feature	3.29
3	Hashtag ratio	2.99
4	Link or Url ratio	2.67
5	Negative sentiment ratio	1.69

D. Experiment 2

In this experiment, we use a unigram model as our baseline which is a bag-of-words textual features model (the dictionary-based feature). Figure 6 compares the performance of various models: First, the temporal model that uses the 1-hour setting. Second, the textual model which uses all the textual features in Table 1. Third, the textual model that implements only optimal features from Table 3. A combination of the temporal model and all textual features is our fourth model. The last model integrates temporal model and the optimal textual features model.

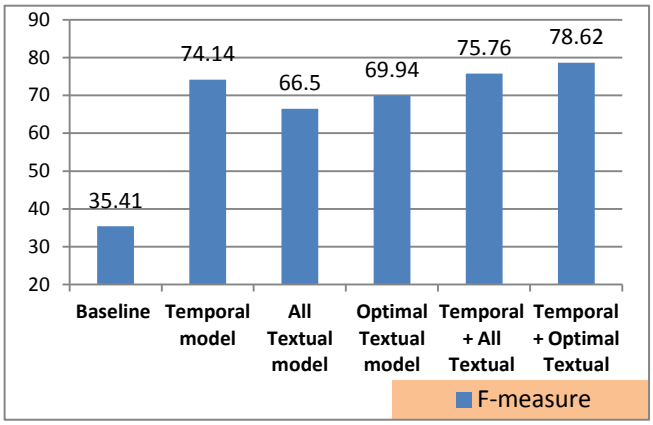

Fig. 6. Results of various models for the disruptive event identification task using the F-measure.

The results in Figure 6 demonstrate that all proposed models attain significantly better performance than the baseline. The temporal model outperforms both textual models by obtaining a performance score of 7.64% over all

textual features and 4.20% compared with the optimal textual features. In addition, using optimal features selected by our approach leads to higher F-measure than using all features. The differences are consistent even when combining textual and temporal features. Finally, an integration between the temporal model and the optimal textual features lead to the best performance which indicates that selecting optimal features improves the event identification task.

VI. CONCLUSION

In this paper, we presented an extensive analysis of various features related directly to Twitter data and showed how they can be used discriminatively to distinguish between disruptive events and other events. The results identify that the temporal and textual features are key predictors of disruptive event identification. Our experimental results also show that the performance of the optimal textual features when combined with temporal features is much better than that of all textual features model. This framework could be used to develop a situational awareness system for the purposes of enriching decision making which can be implemented in many fields such as crisis management, information intelligence, or even daily police work.

There are many directions for future work. One of the main options is to explore additional and relevant features such as spatial features and social network features with relationships of various strengths. Another direction is to compare and validate the performance of the proposed event detection framework against other well-known algorithms such as the state-of-the-art Labeled Dirichlet Allocation (LDA) method.

REFERENCES

[1] Alsaedi, N., Burnap, P. and Rana, O. 2014. A Combined Classification-Clustering Framework for Identifying Disruptive Events. Proceedings of 7th ASE International Conference on Social Computing (SocialCom 2014), pp. 1–10.

[2] Frank, J.R., Kleiman-weiner, M., Roberts, D. a, Niu, F., Ce, Z., Christopher, R. and Soboroff, I. 2012. Building an Entity-Centric Stream Filtering Test Collection for TREC 2012. TREC.

[3] Ji, H. and Grishman, R. 2011. Knowledge Base Population : Successful Approaches and Challenges. Acl, pp. 1148–1158.

[4] Backstrom, L., Kleinberg, J., Lee, L. and Danescu-niculescu-mizil, C. 2013. Characterizing and Curating Conversation Threads : Wsdm, pp. 13–22.

[5] Georgescu, M., Pham, D.D., Kanhabua, N., Zerr, S., Siersdorfer, S. and Nejdl, W. 2013. Temporal Summarization of Event-related Updates in Wikipedia. WWW Companion, pp. 281–284.

[6] Hecht, B., Hong, L., Suh, B. and Chi, E. 2011. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11), pp. 237–246.

[7] Cha, M., Haddadi, H., Benevenuto, F. and Gummadi, P. 2010. Measuring User Influence in Twitter: The Million Follower Fallacy. ICWSM.

[8] Ma, Z., Sun, A. and Cong, G. 2013. On predicting the popularity of newly emerging hashtags in twitter. Journal of the American Society for Information Science and Technology 64(7), pp.1399-1410.

[9] Thelwall, M., Buckley, K. and Paltoglou, G. 2011. Sentiment in Twitter events. Journal of the American Society for Information Science and Technology 62(2), pp. 406–418.

[10] Sarma, T.H., Viswanath, P. and Reddy, B.E. 2013. Single pass kernel k -means clustering method. 38(June), pp. 407–419.

[11] Cui, Y., Wong, W. and Cheung, D. 2009. Privacy-Preserving Clustering with High Accuracy and Low Time Complexity. DASFAA 2009, pp. 456–470.

[12] Alonso, O., Strötgen, J., Baeza-yates, R. a and Gertz, M. 2011. Temporal Information Retrieval : Challenges and Opportunities. Twaw 11, pp. 1–8.

[13] Cheng, J., Adamic, L., Dow, P., Jon, K. and Jure, L. 2014. Can cascades be predicted? WWW '14.

[14] Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R. 2011. Sentiment analysis of twitter data. Proceedings of the ACL 2011 Workshop on Languages in Social Media, pp. 30–38.

[15] James, A. 2002. Introduction to topic detection and tracking. In Topic detection and tracking: event-based information organization, pp. 1–16.

[16] Gabrilovich, E., Dumais, S. and Horvitz, E. 2004. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. WWW '13, pp. 482–490. WWW '14.

[17] Elsas, J.L. and Dumais, S.T. 2010. Leveraging temporal dynamics of document content in relevance ranking. WSDM '10.

[18] Lee, K., Caverlee, J. and Webb, S. 2010. Uncovering social spammers: social honeypots+ machine learning. SIGIR '33.

[19] Petrovic, S., Osborne, M. and Lavrenko, V. 2011. RT to Win! Predicting Message Propagation in Twitter. ICWSM.

[20] Dou, W., Wang, X., Skau, D., Ribarsky, W. and Zhou, M.X. 2012. LeadLine: Interactive visual analysis of text data through event identification. (VAST 2012), pp. 93–102.

[21] Becker, H., Naaman, M. and Gravano, L. 2011. Beyond Trending Topics: Real- Event Identification on Twitter. ICWSM, pp. 1–17.

[22] Burnap, P., Williams, M.L., Sloan, L., Rana, O., Housley, W., Edwards, A., Knight, V., Procter, R. and Voss, A. 2014. Tweeting the terror: modelling the social media reaction to the Woolwich terrorist attack. Social Network Analysis and Mining 4, p. 206.

[23] Radinsky, K., Agichtein, E., Gabrilovich, E. and Markovitch, S. 2011. A word at a time: computing word relatedness using temporal semantic analysis. Proceedings of the WWW'11, pp. 337–346.

[24] Atefeh, F. and Khreich, W. 2013. A Survey of techniques for event detection in twitter. Computational Intelligence 0(0).

[25] Mitra, P., Murthy, C. a and Pal, S.K. 2002. Unsupervised Feature Selection Using Feature Similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI 24(3), pp. 301–312.

[26] Takahashi, T. and Igata, N. 2012. Rumor detection on twitter. SCIS '6 and ISIS '13, pp. 452–457.

[27] Tsur, O. and Rappoport, A. 2012. What ' s in a Hashtag ? Content based Prediction of the Spread of Ideas in Microblogging Communities. WSDM'12. pp. 16–23.

[28] Fawcett, T. 2006. An introduction to ROC analysis. Pattern Recognition Letters 27(8), pp. 861–874.

[29] Kanhabua, N. and Nørvåg, K. 2010. Determining time of queries for re-ranking search results. In Proc. ECDL, pp. 261–272.