Bigram-based features for real-world event identification from microblogs

Conference Paper · July 2017

DOI: 10.1109/ICCCNT.2017.8204004

CITATIONS
3

READS
36

3 authors, including:

Surender Singh Samant
BITS Pilani, Hyderabad
7 PUBLICATIONS **55** CITATIONS
SEE PROFILE

Aruna Malapati
BITS Pilani, Hyderabad
43 PUBLICATIONS **508** CITATIONS
SEE PROFILE

BIGRAM-BASED FEATURES FOR REAL-WORLD EVENT IDENTIFICATION FROM MICROBLOGS

Surender Singh Samant, N. L. Bhanu Murthy, Aruna Malapati

Department of Computer Science and Information Systems
Birla Institute of Technology and Science, Pilani
Hyderabad Campus, Telangana, India
Email: {surender.samant, bhanu, arunam}@hyderabad.bits-pilani.ac.in

***Abstract*—Social media such as microblogs have provided users an open platform to publish content of their choice. Users of microblogs share short content at a very high frequency. This has provided an opportunity to use microblogs as sensor of real world occurrences. Real world events cause a significant shift in the frequency of particular type of messages. We use Twitter microblog to analyze and extract general statistical and linguistic features to distinguish real-world event related content from others. We introduce a method for near real-time identification of events by continuously processing top bigrams from twitter messages. Our experiments show that even with general non twitter-specific features, we can identify events with good accuracy.**

***Index Terms*—event detection, classification, text mining, twitter**

1. INTRODUCTION

Before social media became prevalent, significant real world occurrences were discussed only in traditional media such as newspaper and television. End users would only be the consumer of information and news. With the emergence of social media such as microblogs, such occurrences sometimes get reported and discussed by users even before the traditional media. Microblogs allow short messages to be posted, so they are very high frequency and high volume source of information where content is updated at a very fast rate.

Twitter has emerged as the most popular microblogging site where users send short messages (tweets in short). We used twitter as the microblog of choice for this work as it provides a streaming application programming interface (API) and the messages are posted at a high rate. We note that this work does not use any twitter specific characteristic, so we can substitute any other microblogged stream in place of twitter with minimal change.

We call a new significant occurrence in real world that get discussed in twitter as an event, senders of twitter messages as users, and twitter messages as tweets. Many tweets discuss events, but non-event related tweets comprise the vast majority. Irrelevant tweets can be filtered out and specific event-related tweets analyzed to automatically identify events soon after they start getting discussed.

Event extraction and event identification from twitter has been an actively researched area due to its all-inclusive nature, where anyone from a common person to a celebrity can post a tweet for anyone to read. Another important reason for the research interest is the possibility of creating a dataset that suits the problem at hand. We can create dataset of our choice from the huge amount of messages that users post on twitter and that is made publicly available for real time download by a streaming API.

We collected millions of tweets over different periods of time, preprocessed them, grouped similar tweets using bigram-based approach, and analyzed them to identify features of tweets that discuss currently occurring real-world events (event tweets). Various potential features of event tweets were identified that distinguish them from non-event tweets. The features were then used to create a dataset of samples that were manually labelled as event or non-event. Correlation analysis was performed to rank features, and multiple classifiers were used to evaluate the performance of our system on the dataset.

2. RELATED WORK

There is a significant interest in research involving social media based event identification. Sakaki [1] used tweets as sensors to propose a spatiotemporal model to detect earthquakes in a region and warn users in the nearby regions where the earthquake waves were about to reach. Becker [2] identified many twitter-specific similarity metrics to create clusters of similar tweets that become bigger, or disappear with time. These clusters were then classified as representing events or non-events. Weiler [3] used geographic and time information contained in Twitter data and log-likelihood ratio based approach to identify significant local events. Sankarnarayanan [4] used a seed of users in twitter that were known to discuss news to jumpstart a method for breaking news identification. Ritter [5] used latent variable based models to extract words related to events as well

1. **online shopping 1293**
2. shop online 413
3. weight loss 265
4. **karun nair 99**

(a)

- on diet for bodybuliders drum set online shopping india
- soluble fiber diet ccs online shop review
- of atkins diet 60er style online shop
- lose weight business plan online gift shop
- to get calcium in your diet mcafee antivirus 2014 online shopping
- diet food nike air max tn shop online

- what a treat to watch sweeps by karun nair.. proving better than ak with his jhaadhoo
- karun nair's batting is not too pretty to look at. but it's effective nonetheless!
- individual highest score by any indian against england by karun nair 243(338)*;what an innings.....;india 669/6(178.3);
- karun nair ended the calendar year with a double ton.

(b)

Figure 1. (a) Top four bigrams during one hour in Dec 19, 2016. The number next to bigram is the number of tweets containing words in bigram in any order. (b) Examples of tweets containing the first and fourth of the top bigrams. The fourth bigram represents event about cricket player Karun Nair hitting a double century.

as for part of speech tagging of tweets. Petrovic [6] used a locality-sensitive hashing based method for first story detection from twitter. The goal was to identify the first message that discussed an event. Krstaji [7] used a hybrid approach that combines twitter-specific features along with sentiment-analysis based term scores for event identification.

Our work differs from [1], [3] as the domain and region of events is not restricted. It does not depend on any initial seed of users known to discuss events [4]. Our goal is not to extract event words like [5] or first story detection [6]. The objective of [2], [7] is most similar to ours, but we do not depend upon any twitter specific feature; this single fact makes our work unique and different from other previous works. Our work is a bigram-based approach to group similar tweets and extract bigram-specific, linguistic, and frequency-based features from tweets for event identification in real time. We use only one hour of timestamped tweets to identify an event. This makes our method suitable for near real-time event identification using any microblogging platform that provides a stream of good volume of timestamped messages.

The proposed system with its features is discussed in section 3, the experiments and results are presented in section 4, followed by conclusion.

3. PROPOSED SYSTEM

Twitter provides a streaming API that was used to collect tweets. Keywords can be mentioned in the API to get tweets related containing them. We selected a set of generic words for tracking as any significant English text message is expected to contain at least one of them. Some examples of such generic words are: a, the, of, for, etc. Twitter makes available only randomly selected 1% of published tweets but the volume of twitter is so high that this limit doesn't affect the analysis process. For example, the rate was $\approx$ 2800 English tweets collected per minute.

Since text about an event would contain common words, we used this as a basis of grouping of tweets. We observed that discussion about a significant real world event tend to have multiple common words. This is expected as a number of people use words that are either a named entity or a verb from a fixed set of vocabulary when discussing the same event. Hence, there is a high probability that a word, or a group of contiguous words (after discarding stop words) will be common among different tweets that discuss the same event.

We grouped tweets according to a common word, a common bigram, and common trigram. Grouping tweets containing a single common word resulted in too many unrelated tweets forming a group. Grouping by a common trigram proved to be too restrictive resulting in very few tweets matching the words. This was prohibitive for our work since the change in frequency of tweets containing specific words is a major attribute of an event. Bigram based grouping provided a balance between the two. It resulted in sufficient number of similar tweets representing event related discussions so we used grouping based on bigrams.

The process runs periodically every M minutes to identify the set of top B bigrams (TB) within a period of past one hour as shown in Figure 2. We set M=20 and B=50 for the purpose of this work. Many bigrams only differ in the word ordering, only one of them is used resulting in a decrease in number of TB to process further.

For each TB, all timestamped tweets T during past one hour containing the words of TB in any order and separated by any distance are collected. So, tweets in group T do not necessarily contain the exact bigram, but both the words of the bigram are present. The bigrams were stemmed before looking for words as many tweets may not contain the exact words, but stems would be same. Similar bigrams were merged into one and the count of tweets was updated. An example of this is shown in Figure 1 where the word *shopping* would be stemmed to *shop*. So, the two sets of bigrams would be merged into one bigram - *online shop*. Count of tweets containing this stemmed bigram in any order also gets updated to 1706. Figure 1(b) shows examples of tweets that contain *shop*. This step results in a greater number of tweets in T.

The ratio of count of T to total tweets is calculated for every minute during the past hour(we call this ratio as frequency). The duration of one hour is divided into N equal intervals. For each N as well as the whole hour, various frequency based features are computed. We used N=3 for two reasons. First, it is the minimum intervals required to know the general trend of tweets e.g. if it is continuously increasing, continuously decreasing, first increasing then decreasing, etc. Second, since our event identification process runs periodically every 20 minutes, this division helps the previous event identification process to get over before the beginning of the next cycle. We want to start the next cycle of event identification as soon as possible, so we do it

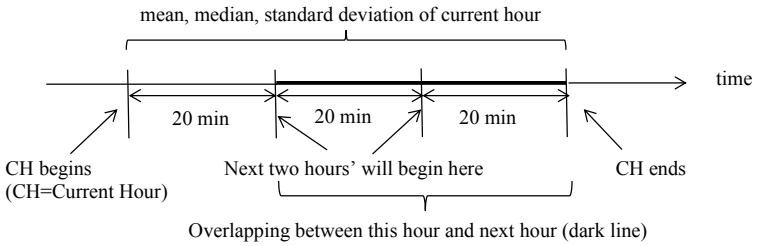

Figure 2. A new event discovery begins every 20 minutes. Tweets of the last 40 minute of Current hour's calculations overlap with tweets of the first 40 minutes of the next hour (shown by the thick line)

periodically every 20 minutes when the results of the earlier calculations have been saved. This makes sure that a real-world event is identified by this method within 40 minutes of it getting discussed in twitter - 20 minutes of delay if it takes place at the earliest in the most recent interval and 20 minutes more for the worst case processing time.

Frequency based approach for event-identification from bigrams is challenging since the majority of top bigrams are not even-related. An example is shown in Figure (1) that shows only the top four bigrams detected by our system on December 19, 2017. Among them, only the fourth one is related to an event.

3.1. Features

We experimented with various potential features that we will discuss now.

3.1.1. Frequency based features. For each of the 20-minute intervals, the mean, median and standard deviation of frequency is calculated. These would provide the information about the distribution of frequency during the continuous 20-min intervals in the past one hour. The same set of measures is also calculated for the whole one hour.

3.1.2. Line of best fit. The per-minute frequency of tweets containing bigrams to all tweets (T_ratio) will follow a trending behaviour [2], [8] that is exponential for significant events. An example of this is shown in Figure 3 where Y axis has been scaled up by 1000 to make the numbers more readable. This behaviour is natural as the number of tweets about an event that has recently taken place will be less, but increase with time as more users get to know about it. Since R^2 statistic has been reported as unsuitable for nonlinear models [9], we take log of the exponential function to fit log(T_ratio) with time (in minutes) on the x axis. The following conversions were applied:

$$Y = Ae^{Bt}, \text{ where } Y = T\_ratio$$
$$logY = logA + Bt \quad (1)$$

In (1), B is the Slope and log A is the intercept This converts the problem into that of fitting a straight line and R^2 statistic can now be applied on the fitted curve. Root mean squared error (RMSE), slope of the fitted line, and R^2 were used as potential features.

3.1.3. Cohesiveness. The cohesiveness of each group of bigram tweets can be calculated by finding similarity of each tweet to the centroid. A vocabulary V out of all the words in T was first created. This is the bag of words with each word in the vocabulary represents one binary feature (1 if it present in the tweet, 0 if it is absent). A centroid is then computed as the vector whose features are calculated by averaging the corresponding features of the group's tweets (2). Cosine similarity (3) measure was used to find similarity of each tweet in T to the centroid. The similarity of all tweets T was averaged to get average similarity as shown in (4).

$$C = \sum_{i=1}^{|V|} C_i, \text{ where } C_i = \frac{\sum_{k=1}^{|T|} V_{k,i}}{|T|} \quad (2)$$

$$cosine\_sim(C,T) = \frac{\sum_{i=1}^{|V|}(C_i, T_i)}{\sqrt{\sum_{i=1}^{|V|} {C_i}^2}\sqrt{\sum_{i=1}^{|V|} {T_i}^2}} \quad (3)$$

$$Similarity_{tweets} = \frac{\sum_{i=1}^{|T|} cosine\_sim(C, T_i)}{|T|} \quad (4)$$

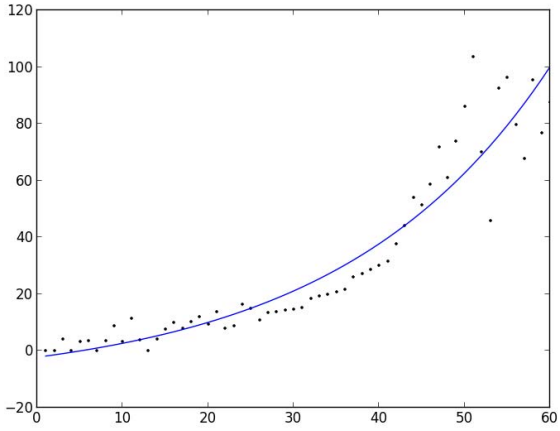

Figure 3. Curve fit to per-min ratio of frequency of tweets containing bigrams to total tweets. Some missing values were filled using regression (three such groups of points can be seen in a straight line).

[Synset('clang.n.01'), Synset('crash.n.02'), Synset('crash.n.03'),
Synset('crash.n.04'), Synset('crash.n.05'), Synset('crash.v.01'),
Synset('crash.v.02'), Synset('crash.v.03'), Synset('crash.v.04'),
Synset('crash.v.05'), Synset('crash.v.06'), Synset('crash.v.07'),
Synset('barge_in.v.01'), Synset('crash.v.09'), Synset('crash.v.10'),
Synset('crash.v.11'), Synset('crash.v.12'), Synset('doss.v.01')]
(a)

'crash.v.01' : The branch crashed down on my car.
'crash.v.02' : The car crashed through the glass door.
'crash.v.03': The plane crashed into the ocean.
'crash.v.04': The terrorists crashed the gate.
(b)

Figure 4. Synsets of word *crash* are shown in (a). Each item in the synset represents a different sense (as a noun n or a verb v in this example). Examples of the first four senses in which *crash* is used as a verb shown in (b).

3.1.4. Verb Similarity. Tweets about events discuss it using same or very similar verb. If similarity of verbs can be calculated in a bigram tweet T, it could be a potential feature. We used Wordnet[10] to find if a word is a verb. Wordnet is a popular lexical database for text analysis. It has different meanings of a word grouped into different contexts called synsets. Words with a similar meaning will have a common synset. Since a verb can have multiple meanings, it can be part of different synsets. Different words that are used with the same meaning will have one common synset among them. As an example, Figure 4 shows synsets for the word crash and example usage as verb from Wordnet. Each synset in the example has a word or word collocation followed by the part of speech and a number. The synsets are ordered in decreasing order of popular usage. Some synsets of crash have a different word or collocation name such as barge_in or doss as those words are more popular in usage than crash in the respective senses.

$$Similarity_{verbs} = \frac{\sum_{i<j,i,j\in n} SV_i \cap SV_j}{\binom{n}{2}}, \text{ where}$$

$$SV_i \cap SV_j = \begin{cases} 1, \text{if there is a common element} \\ 0, \text{otherwise} \end{cases} \quad (5)$$

We select top ten verbs (Vtop) in T and find out the fraction out of all possible pairs of verbs that have a common synset. If synset of a verb is SV, the similarity function we used is given by (5).

3.1.5. Inter-Bigram Group Similarity. During the event hour, a tweet may have more than one top bigram, so it would be present in multiple top groups. If there are a number of different groups of tweets having similar tweets, it indicates occurrence of a popular event. The similarity of different groups was measured using cosine similarity between their centroids. Multiple thresholds were used to count the number of similar groups within the thresholds. These counts along with average cosine similarity between similar groups were used as potential features.

3.1.6. Bigram Position. A bigram's presence mostly at the beginning or the end of a tweet indicates a sponsored or scripted group of tweets or a spam. This knowledge was used to create two potential features. Each of these features denote the fraction of tweets containing bigram at the beginning or end of tweet. This feature would be helpful to eliminate false positives.

Along with the above features, the following intuitive features were also used: presence of a verb in the bigram after removing some common conversational verbs, size of the group, and average tweet length in the group.
Figure 6. shows the main steps that are taken for event identification.

3.2. Feature Correlation Analysis

First, univariate statistical correlation measure was calculated between individual features and the class label (event or non-event) using Pearson's correlation. Features related to goodness of fit and verb similarity are expected to perform well on these tests as they are computed from the overall distribution of frequency in the past one hour. Some features such as mean and median of the one hour frequency would not perform well on univariate measures as they are supposed to be used along with other features as a subset. Table 1 shows the top ranked features. Slope of the fitted line (see line of best fit), R^2, and tweet similarity are ranked high as they are very discriminative event features. Standard deviation of the one hour frequency has a good ranking as there is significant change in frequency when an event takes place. There is also good correlation score for the mean of frequency of the third 20- minute interval. This is due to a new event getting discussed in the last 20 minutes as a result of design of our experiment. Recall that we look for new events periodically after 20 minutes and the mean of the third interval is the ony new mean(rest has already been seen by the earlier event identification process twenty minutes ago (as shown in Figure 2). The binary feature that

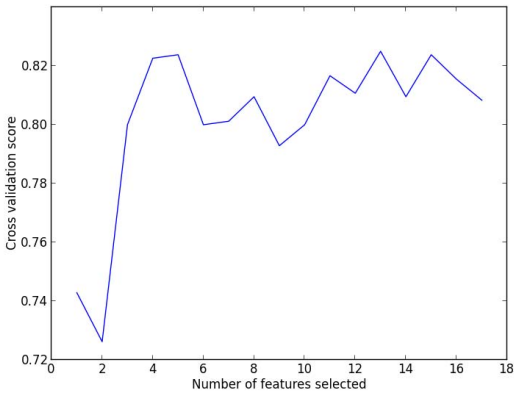

Figure 5. Recursive Feature Elimination on the set of all features; a smaller subset of features was found to be maximizing the performance.

Find_events(Microblog stream S of one hour):
1) PT = Preprocess (S)
2) At each time interval of M=20 minutes:
   Identify_Event(PT)

Identify_Event(PT):
1) Count top N bigrams contained in PT
2) For each of the top N bigrams at time t:
   i) Collect all tweets T in the past hour containing both words of the bigram in any order. Calculate ratio of T to PT during the hour (call it ratio).
   ii) Divide tweets into intervals and compute statistical features of each interval, and for the complete hour
   iii) Compute text features from T
   iv) Run classifier using the computed features

Preprocess (S):
1) Discard tweets containing majority of non-English words and retweets
2) Discard tweets that repeat the same set of words (likely sponsored/script-generated)
3) Remove hashtags, links, references from every tweet
4) Discard too small tweets (< 5 non-stop words)
5) From the remaining tweets extract timestamp and text

Figure 6. Steps taken during one hour of Event identification process.

is ratio of tweets having the bigram at the beginning makes up for the last weakly correlated feature as the training data has a small number of such samples.

There are features that are strongly correlated to events as a subset. Stability selection [11] was used to rank the best features in various subsets of data and features. Stability selection(SS) searches for best features by aggregating multiple runs of feature selection methods on various subset of features as well subsets of data.

Recursive Feature Elimination (RFE) was also performed to rank best performing features from the training set. Figure 5 shows RFE applied on all features on the training data (refer section 4.1 for training set). The number of features found to be maximizing the accuray by RFE on our dataset was was used by classifiers in the same cross-validation loops where all features were used. This provided a direct comparison of classifiers' performance on the two sets of features (see section 4.2).

4. EXPERIMENT

We now discuss our experimental setup and results.

4.1. Dataset

Twitter's Streaming API was used to collect tweets over different periods of time to get coverage of multiple events of different types. Since the streaming API allows for downloading tweets based on tracking keywords, generic words such as as: a, the, of, for, etc. were used as tracking words since any English tweet is expected to contain at least one of these words. Even though the tweets published by Twitter are only 1% randomly selected ones, we expect it to be representative sample of the set of all tweets.

Pearson Correlation	Stability Selection
Slope (0.562)	tweet_sim(0.92)
R^2 (0.555)	Slope(0.86)
tweet_sim (0.307)	mean1(0.81)
std_1hr (0.252)	bigram_end(0.706)
mean3 (0.240)	bigram_vb(0.634)
bigram_begin (0.101)	similarity_vb (0.58)

TABLE 1. TOP SIX FEATURES AND ACCORDING TO PEARSON CORRELATION AND STABILITY SELECTION

The experiment processed $\approx$ 100 million raw tweets collected over different periods of time. Since people tweet according to their convenient time of the day, using tweets only for a certain fixed time of the day is likely to contain majority content from a subset of total users. So, we collected tweets from different time ranges and different dates and months to counter the above effect.

Since the raw tweets contain many irrelevant messages and other metadata, they were preprocessed to get a better dataset for analysis. Two fields from each tweet were used for the purpose of these experiments: timestamp and text. As a preprocessing step, hashtags, references and links were removed from all tweets as they are twitter-specific features. Non-English tweets, too short tweets, tweets containing swear words, repeated tweets, and retweets were discarded. For the purpose of finding top bigrams during each hour, stop words were also removed. Otherwise, many top bigrams would contain a non-significant stop word. This resulted in reduction in size of preprocessed tweets to $\approx$ 17 million. Many bigrams show up two times with only change in the order of words. Such duplicate top bigrams were removed such that only one of them remained to be considered further.

There were times when the tweets were not available for a few minutes due to various reasons such as as network or server issues. Such missing values were filled using linear regression by looking at the frequency at the vicinity of the missing minutes. One such case is shown in Figure 3.

Two human annotators labelled the top groups as event or non-events by looking at the set of tweets T corresponding to each group of tweets. Only groups where the two annotators agreed were considered further. The annotator agreement using Cohen's Kappa coefficient was 0.78.

An equal number of event and non-event groups were used as ground truth to evaluate classifiers. The full dataset contain 124 samples equally divided into event and non-event classes. Scikit-learn toolkit [12] was used for all the experiments.

4.2. Results

The classifiers were evaluated first by 10-fold cross validation on the stratified samples and then by 5-fold cross-validation. Grid search was used to select the best hyper-parameters in terms of F-score (accuracy) of various classifiers. Naive Bayes classifier (NB) was used as baseline classifier, where, in each fold, the complete training set was divided into two classes and these were used as two labelled documents to train the classifier. Each test instance in the fold would then be used against the trained set and its accuracy would be measured. Ensemble methods Random

Classifier	Accuracy	Accuracy13
NB	0.75	-
RF	0.79	0.80
ET	0.81	0.84
SVM	0.79	0.81
RF_5cv	0.81	0.80
ET_5cv	0.82	0.80
SVM_5cv	0.79	0.81

TABLE 2. ACCURACY IN TERMS OF F-SCORE OF VARIOUS CLASSIFIERS BY 10-FOLD AND 5-FOLD CROSS-VALIDATION. THE ACCURACY COLUMN DISPLAYS ACCURACY OF THE CLASSIFIERS USING ALL FEATURES, COLUMN ACCURACY13 DISPLAYS RESULTS USING 13 OPTIMUM FEATURES AS FOUND BY RFE.

Forest (RF) and Extra Tree (ET) classifiers were among the best performing classifiers along with Support Vector Machine (SVM), so we used them in further tests. The performance using all features and best subset of features selected by RFE is shown in Table 2. The accuracy here is the average of F1-score of events and non-events.

From the results, NB baseline has performed well. It is due to the smaller dataset size because of which many of the folds have similar tweets in both training and test sets, resulting in its good performance. As the dataset is increased, we expect its performance to go down. In one experiment, similar top bigram groups were removed if they contained many of the same set of tweets. That resulted in a decrease in performance and the range of F-score was in between 0.69 to 0.73 in multiple tests, whereas the performance of other classifiers was not affected.

The bigger dataset would also have more features represented in training and test and it is expected that the performance of ensemble classifiers and SVM would improve with it. Figure (5) shows that currently the reduced set of features results in an improved performance of classifiers. This is due to under-representation of many of the features because of which they were adversely impacting the effects of strong features, but this effect was negated when more of the stronger features were considered.

In all tests, the performances of Random Forest, Extra Tree and Support Vector Machine were similar. Extra tree and Random Forest were consistent performers across many tests. A reason for their good performance is the inbuilt randomness in them that lets them generalize better than other classifiers. Naive Bayes (NB) had overall lowest accuracy among all the classifiers, but it remain a simple yet effective classifier.

The results for 5-fold cross validation are similar. This shows the stability of the method accross various strong classifiers and different splits of data.

5. CONCLUSION

In this work, we proposed a method to identify events in near real-time from twitter, without using any of twitter specific characteristics. Various features were identified that could be extracted from any microblog similar to twitter to identify real-world events. The accuracy of classifiers using the identified features on the dataset was found to be good. The proposed method can be used to predict, with good accuracy, a new real world event soon after it starts to get discussed in social media. The proposed method could be used along with twitter-specific information to extract more features to classify events only from twitter. It would also be interesting to take as source various other microblogs and combine their messages to get a set of more potential features. An extension of this work is to refine the verb-features as they are very informative in event-related discussions. Another extension is involving more n-grams to detect even more events. Named Entity recognition for twitter could also be used to get even more features.

References

[1] Sakaki, T., Okazaki, M., Matsuo, Y. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. *Proceedings of the 19th International conference on World Wide Web, Raleigh, North Carolina, USA, 2010, Apr 26-30, 851-860.*

[2] Becker, H., Naaman, M., Gravano, L. Beyond Trending Topics: Real-World Event Identification on Twitter. *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, 2011, July 17-21, 438-441.*

[3] Weiler A., Schol M. H., Wanner F., Rohrdantz C., Event Identification for Local Areas using Social Media Streaming Data. *Proceedings of the 3rd ACM SIGMOD Workshop on Databases and Social Networks : DBSocial 2013; New York, NY, USA, June 23 2013.*

[4] Sankaranarayanan, J., Samet, H., Teitler, B., Lieberman, M., Sperling, J. TwitterStand: News in Tweets!. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information, Seattle, WA, USA, 2009, Nov 46, 42-51.*

[5] Ritter A. et al. Open domain event extraction from Twitter . *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD 12, pages 11041112, New York, NY, USA, 2012.*

[6] Petrovic S., Osborne M., Lavrenko V. Streaming First story detection with application to Twitter. *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 181-189, 2010.*

[7] M. Krstajic, C. Rohrdantz, M. Hund, and A. Weiler, Getting There First: Real-Time Detection of Real-World Incidents on Twitter. *2nd IEEE Workshop on Interactive Visual Text AnalyticsTask-Driven Analysis of Social Media as part of the IEEE VisWeek 2012,October 15th, 2012, Seattle, Washington.*

[8] Jure Leskovec, Lars Backstrom, and Jon M. Kleinberg, "Meme-tracking and the dynamics of the news cycle. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.*"

[9] Spiess, Andrej-Nikolai, Natalie Neumeyer. "An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach". *BMC Pharmacology. 2010; 10: 6.*

[10] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM Vol. 38, No. 11: 39-41 1995.*

[11] Nicolai M., Peter Buhlmann. "Stability selection". *Journal of the Royal Statistical Society: Series B Volume 72, Issue 4, pages 417-473, September 2010.*

[12] Pedregosa et al. "Scikit-learn: Machine Learning in Python" , *Journal of Machine Learning Research, Vol 12, pp. 2825-2830, 2011.*