Social Network Analysis and Mining (2020) 10:61
https://doi.org/10.1007/s13278-020-00658-3

ORIGINAL ARTICLE

Feature selection methods for event detection in Twitter: a text mining approach

Ahmad Hany Hossny[1] · Lewis Mitchell[1] · Nick Lothian[2] · Grant Osborne[2]

Received: 17 September 2019 / Revised: 18 May 2020 / Accepted: 20 May 2020 / Published online: 23 July 2020

Abstract
Selecting keywords from Twitter as features to identify events is challenging due to language informality such as acronyms, misspelled words, synonyms, transliteration and ambiguous terms. In this paper, We compare and identify the best methods for keyword selection as features to be used for classification purposes. Specifically, we study the aspects affecting keywords as features to identify civil unrest and protests. These aspects include the word count, the word forms such as n-gram, skip-gram and bags-of-words as well as the data association methods including correlation techniques and similarity techniques. To test the impact of the mentioned factors, we developed a framework that analyzed 641 days of tweets and extracted the words highly associated with event days along the same time frame. Then, we used the extracted words as features to classify any single day to be either an event day or a nonevent day in a specific location. In this framework, we used the same pipeline of data cleaning, prepossessing, feature selection, model learning and event classification using all combinations of keyword selection criteria. We used Naive Bayes classifier to learn the selected features and accordingly predict the event days. The classification is tested using multiple metrics, such as accuracy, precision, recall, F-score and AUC. This study concluded that the best word form is bag-of-words with average AUC of 0.72 and the best word count is two with average AUC of 0.74 and the best feature selection method is Spearman's correlation with average AUC of 0.89 and the best classifier for event detection is Naive Bayes Classifier.

Keywords Social networks analysis · Feature selection · Keyword volume · Event detection · Civil unrest

1 Introduction

Event identification from social media is studied frequently in the last few years as a classification problem, where the text content is used as features among other features such as retweets, likes and shares (Sakaki et al. 2010; Walther and Kaisser 2013; Li et al. 2012). Text is used as features either by tracking keywords temporal signal (Weng and Lee 2011; Guzman and Poblete 2013) or by topic modeling via word clustering (Cordeiro 2012; Abdelhaq et al. 2013; Cataldi et al. 2010) or by calculating the sentiment and the polarity of each post (Thelwall et al. 2011; Popescu and Pennacchiotti 2010).

The key challenge facing keyword-based models is to specify the words to be used as features to train the model, especially that people on twitter use words in a nonstandard way. Using words as features in Twitter is challenging due to the informal nature of the text, the limited length of the tweets and multilingual text. (Fung et al. 2005; Mathioudakis and Koudas 2010; Petrović et al. 2010). The key challenges for analyzing the text on Twitter are listed below:

- The tweet length of 280 letters makes sentiment analysis and topic modeling challenging per each tweet without grouping.
- The frequent usage of abbreviations, misspelled words and acronyms makes multiple words undetectable.

✉ Ahmad Hany Hossny
ahmad.hossny@adelaide.edu.au

Lewis Mitchell
lewis.mitchell@adelaide.edu.au

Nick Lothian
nick.lothian@d2dcrc.com.au

Grant Osborne
grant.osborne@d2dcrc.com.au

[1] School of Mathematical Sciences, University of Adelaide, Adelaide, Australia

[2] Data to Decisions Cooperative Research Centre (D2D CRC), Adelaide, Australia

- Transliterating non-English words using Roman script distorts the signals from similar words in other languages (e.g., the term “boss” in Arabic means “look,” while in English it means “manager”)
- Ambiguous semantics: multiple meanings for the same words according to the context (e.g., “Strike” may refer to a lightning strike or a football strike or a protest)
- Synonyms: similar meanings are expressed by multiple words (e.g., the terms “rally” and “protest” are used interchangeably)

Identifying the keywords most associated with the events of interest is an essential step for event detection using keyword volume. The efficiency of keywords as features varies according to the criteria of the word count per feature, the word form and the techniques used to associate the words time series with the event time series. The word count is important to capture the context of the statement (Forman 2003; Baker and McCallum 1998).

The word form can be n-gram, skip-gram or bag-of-words, which represent the relation between words in each statement such as idioms and grammatical structures (Fernández et al. 2014). And association methods such as correlation, similarity and distance-measuring techniques are a key factor to capture the distributional semantic nature of the text (Yang and Pedersen 1997). According to the hypothesis of the distributional semantic, the words related to the event are likely to be used more frequently on the day of an event than any other day (Mandera et al. 2017; Landauer 2006).

In this research, we examine the different factors affecting the feature selection process to find the best word count, word form and data association method as well as the best combination of the three factors. We do not study the whole text classification problem as it has a broad spectrum that varies according to the nature of the problem, as stated by Allahyari et al. (2017). We only focus on event detection as it has time-series nature and limited training data, and it varies by time.

The proposed framework is used to extract best words to detect civil unrest events in multiple cities, including Melbourne, Sydney, Brisbane and Jakarta. For each city, we manually labeled all civil unrest event days in each city as reported in the press for 641 days to build a binary vector of event/nonevent days, which will be considered as the golden standard records (ground truth) to compare with. A day will be identified as event day if it matched the criteria listed below:

- The count: A group of 100 people or more gathered at one known place
- The cause: The group of people must have a common cause such as human rights, labor rights, anti-racism or even support some sports team
- Description: The gathering is described in the press using any of the terms “protest,” “strike,” “march,” “riot,” “revolt,” if it is not covered in the press, and it is unlikely to be a planned protest. Impulsive protests are out of the scope of this study
- Report count: The protest is mentioned in at least two press articles.
- Elimination: We did not eliminate any protest due to its cause. For example, one of the protests we detected was about the love for “Nutella”

The count of civil unrest events varied in cities according to multiple factors such as population volume, economy, living standards, city activities, political regime, police performance and even the city climate along the year. For example, Melbourne city had 208 days classified as civil unrest event days out of the 641 days covered in this study, while another city such as Brisbane had only 113 events in the same time frame, We extracted the words highly associated with the days of events using the aforementioned criteria and used the selected words to perform binary classification to find event–nonevent days, as described in Fig. 1.

This problem can be formulated as a document classification problem, except we do not classify documents with related contents. We instead classify days as event/nonevent day with huge unrelated content in a noisy environment. The noisy environment, as well as Twitter, and challenges mentioned above made traditional techniques for textual feature

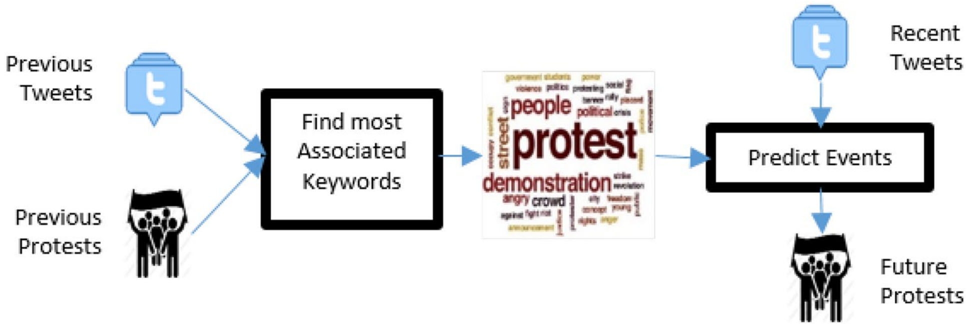

Fig. 1 The proposed framework extracts keywords matching the events of interest, and these keywords will be used later as features to identify similar events

selection unable to find the most informative keywords in accordance with events, due to the spurious data as well as the casual usage of the keywords. The methods discussed by Forman and Yang in Yang and Pedersen (1997); Forman (2003) did not consider either the temporal nature of the data or that of social media streams.

Many researchers used text mining for social analysis as connected networks, media content and people influence (Carley 2003). Jane Diesner et al. analyzed network text to find the organizational structure of covert networks (Diesner and Carley 2004). Also, Jolene Zywica and James Danowski investigated the hypothesis of social compensation by predicting offline popularity using sociability and self-esteem and then mapped the meanings of popularity using semantic networks (Zywica and Danowski 2008). James A. Danowski also studied the identifying actors in a social network by analyzing the time series of the text corpora (Danowski and Cepela 2010). Hewapathirana et al. introduced a spectral embedding approach to track the changes happening in noisy dynamic networks (Hewapathirana et al. 2020). Hossny et al. used singular value decomposition with k-means clustering to enhance the keywords signals to correlate the words with the events vector (Hossny et al. 2018). Also, Taleb et al. introduced a framework to track the temporal communities changes and predict potential events in dynamic social networks (Khafaei et al. 2019)

Text mining researchers studied the factors affecting the results along the whole mining pipeline starting by unstructured input until the output decision. Such factors include the tokenization, lemmatization, stemming, morphological analysis, syntax analysis, keyword selection methods, machine learning algorithm, model selection methods and parameter optimization techniques. These NLP methods can be logical or rule based such as inductive logic programming (Hossny et al. 2008, 2009) or probabilistic such as Bayesian classifiers and decision trees (Chien and Wu 2007; Kurihara and Sato 2006) or using deep learning and high-performance computing (Azzam et al. 2017). This study focus solely on the keyword selection method as studying the combination of all factors will lead to a huge number of possibilities that are difficult to cover.

In Sect. 2, we describe the framework design including the data preparation, the data architecture and the metrics. Section 3 explains the impact of the word form and the word counts as features on the classification results. Section 4, explains the different data association methods and how effective will each method be in the feature selection process. In Sect. 5 we discuss the combinations of the mentioned criteria and conclude the best combination to achieve the best classification results.

2 Framework design

2.1 Data preparation

The framework design aims to detect protests in Melbourne on any day. In this study, we used all tweets issued in Melbourne within 641 days, starting by December 2015 and ending at September 2017. This timeframe is split to 500 days to train the model and 141 days to test the model on multiple randomized folds. The daily tweet count exceeded 4 million tweets after removing non-English tweets, which constitute 2.4 billion tweets in Melbourne along the whole timeframe.

To process this huge amount of data, we used a parallel cluster that has 12 nodes, each node has eight virtual cores and each core has a 2TB memory attached. we used the MapReduce functionality to distribute the tasks per data item on all processors, where the “map” is used to execute instructions per data item, and the “reduce” is used

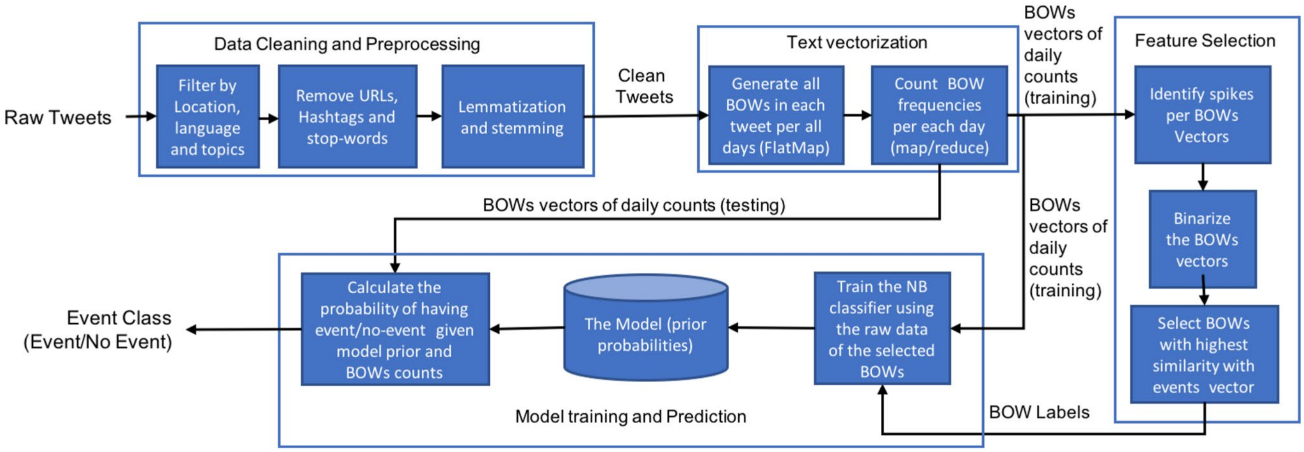
Data Cleaning and Preprocessing
Raw Tweets
Filter by Location, language and topics
Remove URLs, Hashtags and stop-words
Lemmatization and stemming
Clean Tweets
Text vectorization
Generate all BOWs in each tweet per all days (FlatMap)
Count BOW frequencies per each day (map/reduce)
BOWs vectors of daily counts (training)
Feature Selection
Identify spikes per BOWs Vectors
Binarize the BOWs vectors
Select BOWs with highest similarity with events vector
BOWs vectors of daily counts (testing)
BOWs vectors of daily counts (training)
Event Class (Event/No Event)
Calculate the probability of having event/no-event given model prior and BOWs counts
The Model (prior probabilities)
Train the NB classifier using the raw data of the selected BOWs
Model training and Prediction
BOW Labels

Fig. 2 Data processing pipeline including word-count vectorization, selecting words as features, training the model and predicting events

to apply aggregate tasks such as summation, average and grouping

The big data architecture used three main operations exchangeably to perform the mentioned four tasks, which are map, reduce, filter and explode:

- Map: to perform some tasks on all data items in a parallel way, such as lemmatizing or stemming the tweets. The instruction we used for this is named "map" in Python
- Reduce: to perform aggregate operations where all data items having the same key (word), will be grouped into aggregate functions such as summation or counting, or grouping. This process reduces the size of the data frame considerable, as any repeated features will be reduced to a single feature. The instruction we used for this is named "Reduce" in Python
- Filtration: to filter out any data item that does not match the criteria for selection. The instruction we used for this is simply "Filter" in Python
- Explode: convert a single data item into a list of other data items that will be used differently. Extracting the n-grams, skip-grams or bags-of-words is a good example of data explosion as a single statement can be exploded to more than 900 data items. The instruction we used for this is named "flatMap" in Python

The tweets are filtered to the location of interest such as Melbourne City because the protests are usually city-based, and to avoid the noisy social signals coming from other cities in addition to reduce the data volume. The location of each tweet is not always stated clearly in the meta-data or the headers of the tweets. So we identified the location of each tweet by tracking as many of the fields below as possible:

1. Tweet location as reported by Twitter, which is only available if the user allowed twitter app to track his location. Many users do not allow this
2. Time zone can be used to shortlist the probable places where the tweet is issued; for example, most of the population in west Australia lives in Perth city.
3. Longitude and latitude allow us to determine the exact location for issuing the tweet, but this requires the user to allow detecting location, which is not usually happening
4. The location stated in the user profile can help in determining the location of issuing the tweet, where more than 50% of the tweets are issued from someplace inside the listed city in the profile. And, this piece of information is usually listed in all profiles.
5. Location of followers and followees, as users are likely to follow and to be followed by people in the same place. Celebrities and frequent travelers are exceptions as they have followers and followees from everywhere around the globe.

First, we preprocessed the data to extract the keywords in various forms and counts from each tweet as features and then count each feature per each day. Preprocessing includes data cleaning, NLP analysis, feature counting and golden-truth preparation. Example 1 shows tweet cleaning, preparation and vectorization to be used to train the model. These steps are explained as follows:

- Data cleaning is achieved via a sequence of steps as follows: (1) exclude any tweet of any non-English language, (2) exclude the tweets with URLs, (3) remove hashtags, (4) remove non-Latin letters, (5) remove HTML tags, (6) remove punctuation and (7) remove the stopping words according to NLTK list (Loper and Bird 2002).
- Extract the bags-of-words in each tweet by building a list of every two co-occurring words. The count of bags-of-words extracted from a tweet with size n equals $n * (n - 1)$. Each tweet consists of 12 words on average, which makes 132 bags of words per tweet. The average count of different BOWs exceeds 10 million daily after excluding the BOWs with single appearances and after grouping repeated BOWs.
- Every word in each BOW is lemmatized by NLTK lemmatizer to avoid the morphological and grammatical changes to the word shape (e.g., ate → eat)
- Every word in each BOW will be stemmed using Lancaster stemmer to return similar words to the dictionary origin (E.g., British → Brit)
- BOWs are counted in Melbourne tweets for each day to prepare the term frequency vectors.

Example 1:
Original Tweet:
Protesters may be unmasked in wake of Coburg clash https://t.co/djjVIfzO3e (News) #melbourne #victoria

Cleaned Tweet:
protest unmask wake coburg clash news

List of words: ‘protest’, ‘unmask’, ‘wake’, ‘coburg’, ‘clash’, ‘news’

Two-words n-grams: [‘protest’, ‘unmask’], [‘unmask’, ‘wake’], [‘wake’, ‘coburg’],. . . , [‘clash’, ‘news’]
Three-words n-grams: [‘protest’, ‘unmask’, ‘wake’], [‘unmask’, ‘wake’, ‘coburg’], . . . , [‘coburg’, ‘clash’, ‘news’]
Two-words skip-grams: [‘protest’, ‘unmask’], [‘protest’, ‘wake’], [‘protest’, ‘coburg’], . . . , [‘unmask’, ‘wake’],. . . , [‘clash’, ‘news’]
Three-words skip-grams: [‘protest’, ‘unmask’, ‘wake’], . . . , [‘unmask’, ‘wake’, ‘coburg’],. . . , [‘coburg’, ‘clash’, ‘news’]
Two-words BOWs: [‘protest’, ‘unmask’], [‘protest’, ‘wake’], [‘coburg’, ‘protest’], . . . , [‘unmask’, ‘wake’],. . . , [‘clash’, ‘news’]
Three-words BOWs: [‘protest’, ‘unmask’, ‘wake’], . . . , [‘coburg’, ‘unmask’, ‘wake’],. . . , [‘clash’, ‘coburg’, ‘news’]

Assuming the feature will be a single word:
‘protest’ training : $[x_{1,1}, x_{1,2}, x_{1,3}, \dots, x_{1,500}]$
‘unmask’ training : $[x_{2,1}, x_{2,2}, x_{2,3}, \dots, x_{2,500}]$
‘protest’ testing : $[x_{1,501}, x_{2,502}, \dots, x_{1,641}]$
‘unmask’ testing : $[x_{2,501}, x_{2,502}, \dots, x_{2,641}]$

Assuming a time frame of 20 days
Feature vector: [2,3,3,4,5,3,2,3,8,3,3,1,3,9,3,1,2,4,5,1]
Event days(*groundtruth*): [0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,1,0,1,0]

Once we prepare the words as feature vectors, we measure the association between the training split for each feature vector and that of the ground truth vector. The feature can be a single word or n-gram or skip-gram or bag-of-words, and the association method can be a correlation, similarity metric or a distance metric. Once we calculate the feature associativity with the ground truth, we rank the features according to the correlation or the similarity score and use the top 100 features to classify the days as event day or non-event day (Fig. 2).

The classification step is evaluated using multiple classification algorithms, such as the logistic regression, Naive Bayes, decision trees, SVM and KNN. In this paper, we report the results using Naive Bayes classifier only, as it achieved better results with limited training data, and it is known to be one of the best classifiers for text classification problems (Zhang et al. 2007). And the evaluation is performed using multiple metrics including the area under the ROC curve, the area under the PR curve, the accuracy, the precision, the recall and the F1-score (Tables 1, 2) (Koyejo et al. 2014).

2.2 The big data architecture

Preparing the data for this study required a huge amount of data estimated by 2.4 billion tweets in Melbourne, which make around 24 billion words along the entire timeframe. This amount of data needed a big data architecture to manage the necessary processes needed for data storage, processing, learning and prediction as indicated in the points below:

1. To stream data from GNIP (Twitter) using Kafka.
2. To store the data in the cluster using Hadoop.

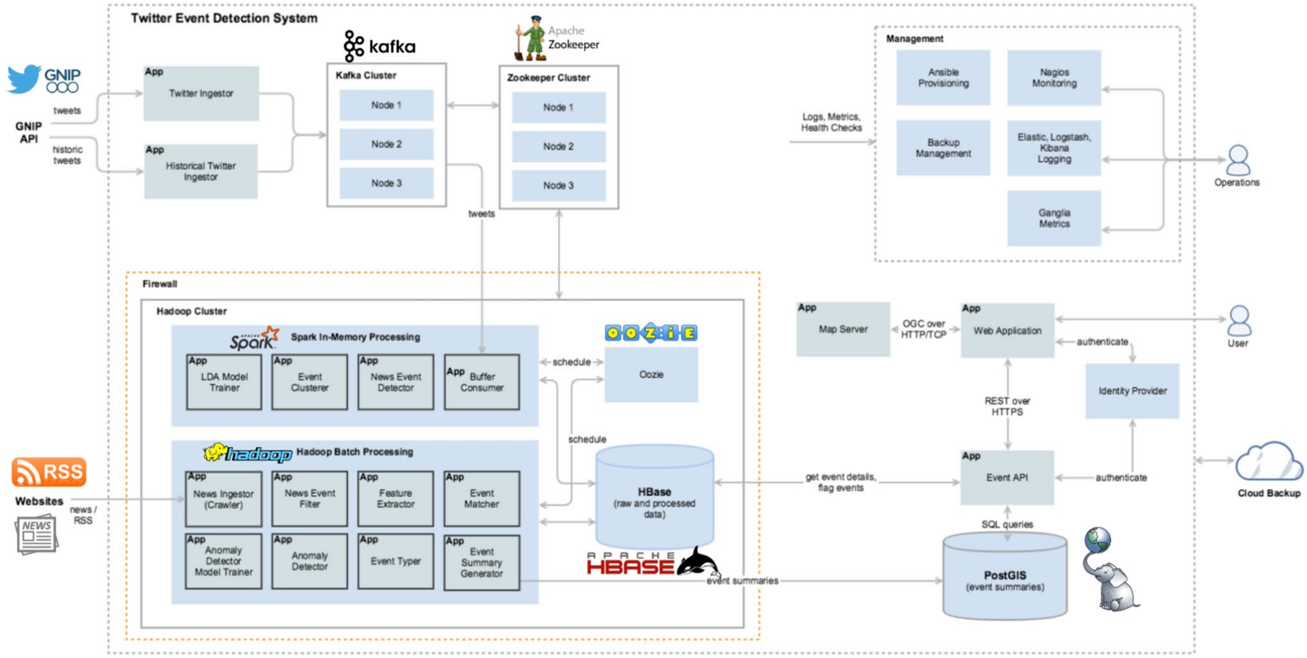

Fig. 3 High-level overview of the big data architecture processing social network streams

Table 1 Confusion matrix for binary classification and the corresponding array representation used in this study

	Actual positive class	Actual negative class
Predicted positive class	True positive (*tp*)	False negative (*fn*)
Predicted negative class	False positive (*fp*)	True negative (*tn*)

3. To perform in-memory NLP tasks such as cleaning, ingesting the data in a parallel scheme using Spark and NLTK.
4. To select and reduce features before building the model in the learning stage using Spark ML with scikit-learn.
5. To store the ingested data and extracted features in a database of HBase and PostgreSQL to facilitate structured access to the data
6. To use the selected features to build the model that will be used later for prediction.

Figure 3 illustrates how the big data framework is designed technically including all the servers, the clusters, the parallels systems, the processing units, the nonstructured storage, the structured database and the Web interface.

2.3 The metrics

In this study, we used multiple metrics to validate and verify the quality of our results. Each metric has its objective, advantages and limitations in evaluating the classification results, which are listed below:

Table 2 Most frequently used classification metrics

	Formula	Description
Accuracy(acc)	$\frac{tp + tn}{tp + fp + tn + fn}$	the ratio between the correctly predicted instances and total number of instances
Precision(p)	$\frac{tp}{tp + fp}$	How many of the correctly classified instances are relevant
Recall(r)	$\frac{tp}{tp + fn}$	How many of the relevant instances are classified correctly
F-score(F1)	$\frac{2 * p * r}{p + r}$	the harmonic mean of precision and recall, immune to data imbalance

The area under the ROC curve or shortly the area under the curve (AUC) evaluates the quality of binary classification models for each class. The ROC curve represents the relation between the false positive rate (FPR) and the true positive rate (TPR). The ROC curve covers the whole spectrum of the TPR and FPR between zero and one, which makes it a statistically significant metric. The area under the ROC curve varies between zero and one, where 0.5 represents the random classifier where the TPR equals exactly the FPR, and one represents the perfect classification, where the TPR is always one regardless of the FPR. The main disadvantage of the AUC is the sensitivity to the data imbalance, which may require more data balancing before testing the predictions. The AUC is calculated using Eq. 1

$$AUC = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n} \tag{1}$$

where S_p is the sum of positive observations, n_p is the count of positive examples and n_n denotes the negative observations. The AUC is proven to be a better metric than the accuracy for evaluating the performance of binary classifiers. Using AUC to evaluate the multiclass classification is slightly more complicated where it uses one-vs-all classification scheme which will generate n AUC scores for each class, it can also use the one-vs-one scheme that generates $n * (n - 1)$ AUC scores for all combinations of all classes.

3 The impact of word form and count

In this section, we differentiate word forms by the number of features extracted from each tweet, and the features frequency within the data frame. n The number of extracted features affects the total size of the data frame, which might be scaled up by a factor of 1120 for a skip-gram of three words and a sentence of 16 words, which can make the problem computationally very costly. On the other hand, the recurrence factor affects the aggregate reduction processes such as summation or grouping. As if we have the same feature occurring for a million times, it can be reduced into a single record with the sum of the million values.

By word form, we mean the words are sequential and contiguous in the multiword feature such as n-gram, skip-gram and bag-of-words (BOWs). The multiword features can capture the parts of speech, the phrases, the idioms and the context of the sentence, which help to resolve the lexical ambiguity, where the same word has multiple meanings or usages according to the intention, the context and the author (Mikolov et al. 2013; Levy and Goldberg 2014).

For example, the term "strike" can be used to refer to a "lightning strike" or a "football strike" or "protest strike." On the other hand, using complicated word forms causes the signal to be weaker as the term "strike" in the expression "strike storm" and the expression "strike rain" will be considered in different terms, even though the two expressions refer to the same meaning of "strike" and the probably occur in the same tweet. Example 1 lists examples of n-grams, skip-grams and bag-of-words. The subsections below explain each term in detail.

3.1 N-grams

N-grams are the sequence of n contiguous words in one sentence, which is extracted using a sliding window of n words from the start to end. The most simple word form is the single word, which is usually referred to as a unigram. The unigram has a limited number of occurrences limited to the dictionary of interest, which can be 300k words of the dictionary words and up to 4 million words, including slang words and acronyms. Unigrams are computationally feasible due to the limited number of words and the recurrent use of the same words within millions of tweets, which make the reduction for aggregation very effective in reducing the total size of the data frames.

Another word form is the n-gram, which considers the n number of adjacent words as a single entity or feature. N-gram can capture the phrase structure as a noun phrase or verb phrase. It can also capture the idioms, and it sometimes can capture the context according to the nature of the document and the length of the N-grams (N-words) (Lampos and Cristianini 2012; Cheng et al. 2006). The main drawback of n-gram is the low probability of repetition as it is unlikely to have the same terms in the same order adjacently multiple times. The number of n-gram features extracted from a tweet of length m is calculated by the formula of $(m - n + 1)$

3.2 Skip-grams

Skip-gram is similar to the n-grams as it takes n number of words in the same sentence and maintains the word order, but it relaxes the constraint of the contiguity. This allows skip-grams to capture the context better as it identifies the word meaning by co-occurring words in the same tweet, regardless of the contiguity factor (Cheng et al. 2006). In the skip-gram word form, the model uses the word to identify context words in the surrounding window. The skip-gram model gives higher weights to the near context words and smaller weights for the distant context words (D'hondt et al. 2012).

Although extracting skip-grams from a sentence can list few word sequences that capture idioms or phrases, but we cannot count on that as most of the other generated word-sequences are likely not adjacent (Shazeer et al. 2015). Skip-grams are more likely to be repeated in multiple tweets than n-gram as they do not require specific adjacency. On the other hand, skip-gram causes the data to explode as it considers all ordered combinations of co-occurring words. For example, a sentence of length 15 words can generate 910 skip-grams of size three words for each tweet. The count of the skip grams of length n and generated from a sentence of length m is simply the combination of all words in the sentence $\binom{m}{n}$, which is equal to $m!/n!(m-n)!$. Some other factors affect the number of generated skip-grams including how many words will be skipped and if the skipping will be rigid to a specific number of steps or any number of steps less than some upper bound (Guthrie et al. 2006).

3.3 Bag of words

Bag-of-words is the set of words in any sentence grouped in n-size bags regardless of order or contiguity. This allows bags of words to capture all the contexts and build a strong signal as the combination of words is more likely to be repeated than ordered words as in the skip-grams and the n-grams (Wallach 2006).In the bag of words, we can identify the meaning of the current word from a window of surrounding context word. Although bag-of-words causes the sentence to explode as it considers all combinations of all words, the repetition factor makes any aggregate process such as count or summation or grouping more feasible computationally than skip-gram or n-gram, which less re-occur frequently (Blumenstock 2008).

The count of the generated bags-of-words from each tweet is similar to the count of the skip-grams as they both consider all combinations of all words $\binom{m}{n}$, which is equal to $m!/n!(m-n)!$. As bag-of-words has the advantage that words (A,B,C) = (B,A,C) = (B,C,A) = ..., it can reduce the total number of features through any aggregation process by a factor of n! that represents the number of permutations of any number of words in the bag. This makes the total number of BOWs much less than skip-grams within the whole data frame, which makes it computationally more feasible. According to Mikolov, BOWs are faster while skip-gram is slower but does a better job for infrequent words (Mikolov et al. 2013).

As we used the different combinations of word forms and word counts as features to classify civil unrest events, we found that the best word form is bags-of-words (BOWs) as it achieved the highest classification results in all metrics for the same word count. We also found that the best word count to be used as a feature is two as it achieved the best classification scores for all metrics using the same word form. By combining the word form and word count, we found that bags-of-words of size two words per bag are the best feature for classification purposes. The next section compares the different combinations of word forms and word counts for each data association method according to the metrics explained in Subsect. 2.3.

4 Data association methods

In this section, we explain the most frequently used data association methods used to select the words as features from social networks to match events. For each method, we will explain its concept, assumptions and limitations and its usability in selecting words as features. The data association methods can be either statistical-based such as correlation techniques, or similarity-based such as mutual information or cosine similarity.

4.1 Pearson's correlation

Pearson's correlation is a measure of linear association between two variables, as the slope of the regressed line equals Pearson's correlation times the ratio of standard deviations. Pearson's correlation assumes the variables are related and the outliers are eliminated as a single outlier can rotate the correlation line away from the mainstream to reduce the error, and the variables follow the normal

Table 3 Classification results using the features selected using Pearson's correlation considering different word forms and word counts

Word form	Count	AUC	F1	AUPR	Precision	Recall	Accuracy
n-gram	1	0.708	0.527	0.624	0.628	0.456	0.754
n-gram	2	0.642	0.446	0.549	0.497	0.429	0.674
n-gram	3	0.676	0.490	0.593	0.558	0.469	0.699
BOW	2	0.764	0.498	0.618	0.651	0.407	0.754
BOW	3	0.731	0.546	0.637	0.624	0.494	0.754
Skip-gram	2	0.689	0.450	0.557	0.516	0.426	0.676
Skip-gram	3	0.601	0.312	0.544	0.605	0.265	0.683

distribution and homoscedasticity, where variance around the regressed line is the same (Benesty et al. 2009; Havlicek and Peterson 1976).

Let the means of X and Y be $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ and $\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$, respectively. The Pearson's correlation coefficient $\rho_{pearson}$ will be defined as:

$$\rho_{Pearson}(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{(x)})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}\left(x_i - \bar{(x)}^{2}\right)\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

In the case of joint normal distributions, Pearson's correlation coefficient follows the t-distribution with n^2 degrees of freedom when X and Y are not dependent. If the variables are not joint normally distributed, Fisher's transformation can be used to get an asymptotic normal distribution (Lawrence and Lin 1989).

If X and Y are linearly dependent, we set $\rho_{Pearson}(X, Y) = \pm 1$. In the case of a perfect positive (increasing) linear relationship, Pearson's correlation coefficient is set to +1, and in the case of a perfect negative (decreasing) linear relationship, it is set to −1. If X and Y have no linear association whatsoever, $\rho_{Pearson}(X, Y) = 0$, and in the case of partial linear dependency, $-1 < \rho_{Pearson}(X, Y) < 1$. Although no data association will score zero correlation, zero correlation does not necessarily imply no data association.

As we search for best keywords matching social events through Twitter, Pearson's correlation did not prove to be the best feature selection method as the word counts in Twitter are not guaranteed to satisfy the assumptions of Pearson's correlation such as normality, linearity and homoscedasticity (Kim et al. 2012).

In our study, we used Pearson's correlation to find the words in different forms and counts that are most correlated with event days. These words will be used as features that train and test the classifier. Table 3 reports the results of Naive Bayes classifier using the selected features.

4.2 Spearman's correlation

Spearman's correlation is a ranking metric that calculates the linear correlation between the ranking variables for each of the independent and the dependent variables. It is the application of the linear correlation between the variables after converting the observed data into rank variables. The dependency on the rank variable instead of the actual variable allows Spearman's correlation to relax the constraints of Pearson's correlation, although it will require the signals to be independent and the response variables to be monotonic (Hauke and Kossowski 2011). Spearman's correlation does not require the variables to be cardinal, so it can evaluate the association between the variables if they are ordinal as well (Kruskal 1958).

Let X and Y be the vectors of the observed data of size n, and let rgX_i and rgY_i be the rank variables for X and Y, respectively. The Spearman's correlation will be defined as :

$$r_s = \rho_{rg_X, rg_Y} = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{3}$$

where ρ denotes the Pearson's correlation applied to rank variables, $cov(rg_X, rg_Y)$ denotes the covariance of the rank variables and $\sigma_{rg_X}\sigma_{rg_Y}$ are the standard deviations of the rank variables of X and Y, respectively.

Spearman's correlation can measure the association of the words with social events better than Pearson's correlation, but only for short timeframes where the word count and event count are guaranteed to be monotonically increasing or decreasing. For longer timeframes, the word counts and event counts fluctuate up and down, which makes Spearman's correlation not the best option to find the most associated words with the events (Zhang et al. 2007).

Spearman's correlation $\rho_{Spearman}$ takes values between −1 and 1, where the correlation coefficient equals 1 for the monotonically increasing relationship (for all x_1 and x_2 such that $x_1 < x_2$, we have $y_1 < y_2$), and the correlation coefficient equals to -1 for the monotonically decreasing relationship (where x_1 and x_2 such that $x_1 < x_2$,we have $y_1 > y_2$). If the random variables are monotonically independent, $\rho_{Spearman}(X, Y) = 0$, and if the variables are

Table 4 Classification results using the features selected using Spearman's correlation considering different word forms and word counts

Word form	Word count	AUC	F1	AUPR	Precision	Recall	Accuracy
n-gram	1	0.876	0.713	0.770	0.765	0.675	0.836
n-gram	2	0.915	0.810	0.843	0.827	0.796	0.887
n-gram	3	0.851	0.662	0.738	0.764	0.586	0.820
BOW	2	0.907	0.780	0.812	0.766	0.795	0.866
BOW	3	0.917	0.775	0.816	0.803	0.754	0.869
Skip-gram	2	0.926	0.793	0.838	0.853	0.744	0.883
Skip-gram	3	0.838	0.682	0.776	0.850	0.572	0.840

partially monotonically dependent, $-1 < \rho_{Spearman}(X, Y) < 1$. $\rho_{Spearman}(X, Y) = 0$ does not necessarily mean the random variables X and Y are totally independent and they are just monotonically independent, but they may have another kind of data association (Myers and Sirois 2006).

In this study, we used Spearman's correlation to find the words in different forms and counts that are most correlated with event days. These words will be used as features that train and test the classifier. Table 4 reports the results of Naive Bayes classifier using the selected features.

4.3 Distance correlation

Distance correlation is first introduced by Szekely et al. (2007) to calculate the nonlinear correlation between two variables (Székely et al. 2007). Distance correlation calculates the statistical distance between two probability distributions by dividing the Brownian covariance between X and Y (distance covariance) by the product of the distance standard deviations (Székely and Rizzo 2009; Ayache et al. 2000). The formulation of the distance correlation is stated below:

$$dCor(X, Y) = \frac{dcov(X, Y)}{\sqrt{dVar(X)dVar(Y)}} \tag{4}$$

where

- $dCov(X, Y) = \sqrt{\frac{1}{n^2} \sum_{k=1}^{n} A_{k,l} B_{k,l}}$
- $dVar(X) = dCov(X, X)$
- $dVar(Y) = dCov(Y, Y)$
- $A_{k,l} = a_{k,l} - \bar{a}_k - \bar{a}_l - \bar{a};$
- $B_{k,l} = b_{k,l} - \bar{b}_k - \bar{b}_l - \bar{b};$
- $a_{k,l} = ||x_k - x_l||$ is the distance between x_k and x_l
- $b_{k,l} = ||y_k - y_l||$ is the distance between x_k and x_l
- $\bar{a}_k$ is the kth row mean for x
- $\bar{a}_l$ is the first column mean for x
- $\bar{b}_k$ is the kth row mean for y
- $\bar{b}_l$ is the first column mean for y

The value of dCor can vary between 0 and 1, where the 1 means a perfect variable dependence and 0 means the random variables are not dependent. If the variables are partially dependent, $0 < Cor(X, Y) < 1$. The main advantage of distance correlation is that the zero score implies necessarily there is not any kind of statistical dependence or association between the two variables.

Distance correlation is more capable of detecting the words associativity with social events than any other correlation techniques due to its tolerance to nonlinearity and its ability to measure the distance between distributions. The only issue with this technique is it requires a noise-free statistical distribution for the word signal, which is very challenging considering the noisy nature of Twitter and social networks in general (Li et al. 2012).

In this study, we used distance correlation to find the words in different forms and counts that are most correlated with event days. These words will be used as features that train and test the classifier. Table 5 reports the results of Naive Bayes classifier using the selected features.

Table 5 Classification results using the features selected using distance correlation considering different word forms and word counts

Word form	Count	AUC	F1	AUPR	Precision	Recall	Accuracy
n-gram	1	0.807	0.642	0.707	0.689	0.606	0.796
n-gram	2	0.773	0.566	0.656	0.653	0.508	0.765
n-gram	3	0.756	0.545	0.632	0.620	0.491	0.756
BOW	2	0.832	0.683	0.732	0.682	0.686	0.809
BOW	3	0.824	0.639	0.720	0.738	0.572	0.808
Skip-gram	2	0.800	0.616	0.685	0.660	0.582	0.783
Skip-gram	3	0.781	0.573	0.679	0.716	0.484	0.783

Table 6 Classification results using the features selected using mutual information similarity metric considering different word forms and word counts

Word form	Count	AUC	F1	AUPR	Precision	Recall	Accuracy
n-gram	1	0.530	0.396	0.504	0.311	0.566	0.485
n-gram	2	0.544	0.388	0.488	0.323	0.501	0.534
n-gram	3	0.561	0.404	0.500	0.333	0.523	0.537
BOW	2	0.566	0.427	0.533	0.325	0.629	0.492
BOW	3	0.573	0.458	0.572	0.338	0.720	0.490
Skip-gram	2	0.553	0.433	0.544	0.323	0.665	0.479
Skip-gram	3	0.561	0.448	0.560	0.333	0.695	0.487

4.4 Mutual information

Mutual information measures the amount of information held in one variable describing the other variable. Mutual information assesses the similarity of the joint distributions for two variables by multiplying the marginal distributions of each variable. This makes *MI* more generic than correlation because it is not limited by the numerical cardinal values. Mutual information can also be applied to categorical, binary and ordinal values (Fraser and Swinney 1986). Considering that *MI* uses the similarity of the distribution, it focuses mainly on comparing the whole statistical distribution of the two variables instead of pairing the individual observations of the two variables. This makes *MI* more useful to clustering than classification (Viola and Wells 1997).

The MI of two continuous random variables X and Y can be defined as:

$$MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y).log(\frac{p(x,y)}{p(x)p(y)}) \tag{5}$$

where $p(x, y)$ is the joint probability of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions for X and Y. In the case of continuous random variables, the summation process will be replaced by integration (Wells et al. 1996).

MI scores range between zero and one, where one means a high certainty that one variable holds information about the other and the zero score refers to the low certainty that one variable holds information about the other, although zero MI score does not imply variables independence. For our problem for feature selection, MI can be used to cluster the words with similar distributions assuming the data are not noisy, but it could not find the best words associated with the days of events (Church and Hanks 1990; Dodds et al. 2011).

In this study, we used the mutual information method to find the words in different forms and counts that are most correlated with event days. These words will be used as features that train and test the classifier. Table 6 reports the results of Naive Bayes classifier using the selected features.

4.5 Cosine similarity

Cosine similarity calculates the cosine of the angle between two vectors. The cosine metric evaluates the similarity of direction for the vectors instead of the similarity of the magnitude. The score of cosine similarity is valued to be 1 if the two vectors have the angle of zero between their directions, and the score is valued to be zero if the two vectors are perpendicular (Crandall et al. 2008). In case the two vectors are oriented to opposite directions of each other, the similarity score will be −1.

Cosine similarity is frequently used in high-dimensional spaces such as text mining and information retrieval, where each word is considered a feature represented as a new dimension, and each document is identified by a vector of features (dimensions) associated with the word count in the document (Matsuo et al. 2007). Cosine similarity is a good indicator of how similar two documents can to be in terms of their topics (Singhal 2001). Cosine similarity is formulated as: let α , β be two vectors $\alpha = p1, p2, p3 \ldots$ and $\beta = q1, q2, q3 \ldots$ and let "*" denote a scalar product between two vectors. The cosine similarity between α, β is given by

$$CS = \frac{(\alpha * \beta)}{(|\alpha| \times |\beta|)} \tag{6}$$

where $\alpha * \beta = (p1 * q1 + p2 * q2 + p3 * q3 + \cdots) = \sum_{i=1}^{n} p_i * q_i$, $|\alpha| = \sum_{i=1}^{n} p_i^2, |\beta| = \sum_{i=1}^{n} q_i^2$, CS is cosine similarity, $|\alpha|$ and $|\beta|$ are the magnitudes of vectors α and β, respectively.

Cosine similarity is used for feature selection for document clustering purposes, where it is used to find how redundant are the features. Gram–Schmidt orthogonal feature selection method used the cosine similarity as well as both backward elimination approach and forward features selection (Dubey and Saxena 2016; Hazewinkel 2001).

Cosine similarity is useful for text clustering such as topic modeling, where the orientation of the word vectors is more important than the magnitude. Cosine similarity was not useful in finding the best words as features for event

Table 7 Classification results using the features selected using cosine similarity metric considering different word forms and word counts

Word form	Count	AUC	F1	AUPR	Precision	Recall	Accuracy
n-gram	1	0.668	0.446	0.533	0.437	0.469	0.653
n-gram	2	0.587	0.345	0.454	0.344	0.374	0.607
n-gram	3	0.513	0.224	0.362	0.287	0.191	0.616
BOW	2	0.658	0.413	0.524	0.458	0.411	0.670
BOW	3	0.643	0.252	0.423	0.300	0.349	0.591
Skip-gram	2	0.653	0.384	0.506	0.421	0.412	0.648
Skip-gram	3	0.525	0.187	0.355	0.199	0.301	0.566

detection within social networks because the word daily magnitude emphasize how many people are talking about the event, which reflects the social event size such as the protest volume (Sayyadi et al. 2009; Pennacchiotti and Gurumurthy 2011).

In this study, we used the mutual information method to find the words in different forms and counts that are most correlated with event days. These words will be used as features that train and test the classifier. Table 7 reports the results of Naive Bayes classifier using the selected features.

4.6 Jaccard similarity

Jaccard similarity compares the individual members of two sets to identify common elements versus the different ones. The key advantage of Jaccard similarity is it ignores the null values in the two vectors and considers the nondefault correct matches compared to the mismatches, which makes the metric immune to data imbalance. Jaccard index outperforms cosine similarity as it retains the sparsity property and it also allows the discrimination of the collinear vectors.

Jaccard index is calculated for the sets and binary vector as the ratio between the intersected elements and common elements as indicated in Eq. 7, which makes the score value between zero for no common elements and one for all elements in common. The Jaccard index is generalized to calculate the similarity of between vectors of cardinal numerical data using the formula in Eq. 8

$$JS(x,y) = \frac{(x \cap y)}{(x \cup y)} = \frac{(x \bigwedge y)}{(x \bigvee y)} \tag{7}$$

$$JS(x,y) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)} = \frac{tp}{fp + fn + tp} \tag{8}$$

where tp is the true positives, fp false positives and fn is false negatives. Comparing the true positives with the false positive and false negatives can indicate whether specific word values match the event days without being a lucky coincidence, as any mismatched dates for false negatives or false positives will be penalized by increasing the denominator value. Jaccard distance evaluates the dissimilarity between two vectors that is the complement of the Jaccard similarity (1-Jaccard similarity). Jaccard distance can also be interpreted as the ratio of the size of the symmetric difference between two sets to the union (Niwattanakul et al. 2013).

Jaccard index is a useful metric for event detection and keyword selection as features, as it measures the term frequency in event days relative to term frequency in nonevent days and ignores the true negatives where the event did not occur, and the term did not appear. This method is specifically useful if the term consisted of multiple words such as 2-words or 3-words n-grams, BOWs or skip-grams, as the longer term leads to more zeros in nonevent days. On the other hand, the single-word term can occur in any day in a spurious accidental way that will distort Jaccard similarity index (Unankard et al. 2015; Nasution et al. 2016).

In this study, we used the Jaccard index method to find the words in different forms and counts that are most correlated with event days. These words will be used as features that train and test the classifier. Table 8 reports the results of Naive Bayes classifier using the selected features.

Table 8 Classification results using the features selected using Jaccard similarity metric considering different word forms and word counts

Word form	Count	AUC	F1	AUPR	Precision	Recall	Accuracy
n-gram	1	0.754	0.593	0.655	0.567	0.629	0.741
n-gram	2	0.825	0.681	0.727	0.652	0.715	0.799
n-gram	3	0.822	0.598	0.668	0.636	0.570	0.770
BOW	2	0.898	0.801	0.827	0.732	0.887	0.867
BOW	3	0.883	0.795	0.826	0.778	0.819	0.873
Skip-gram	2	0.864	0.752	0.785	0.706	0.805	0.841
Skip-gram	3	0.879	0.791	0.821	0.763	0.825	0.869

Table 9 Comparing the best result for each data association method considering its word form and word count. The features selected using Spearman's method and skip-gram with two-words achieved best classification results

Method	Form	Count	AUC	F1	AUPR	Precision	Recall	Accuracy
Pearson	BOW	2	0.764	0.498	0.618	0.651	0.407	0.754
Spearman	skip-gram	2	0.926	0.793	0.838	0.853	0.744	0.883
Distance	BOW	2	0.832	0.683	0.732	0.682	0.686	0.809
Mutual information	BOW	3	0.573	0.458	0.572	0.338	0.720	0.490
Cosine	Unigram	1	0.668	0.446	0.533	0.437	0.469	0.653
Jaccard	BOW	2	0.898	0.801	0.827	0.732	0.887	0.867

Table 10 Average classification scores for event detection in multiple cities using keywords selected by Spearman's method, word form of skip-gram of size two. The results are cross-validated randomly on tenfold

City	Event count	Tweet count	AUC	F1	AUPR	Precision	Recall	Accuracy
Melbourne	208	4.1M	0.926	0.793	0.838	0.853	0.744	0.873
Sydney	212	4.9M	0.909	0.732	0.791	0.789	0.704	0.860
Brisbane	113	2.1M	0.804	0.562	0.598	0.594	0.539	0.855
Perth	181	2.6M	0.899	0.698	0.714	0.675	0.725	0.883
Jakarta	219	8.9M	0.873	0.718	0.855	0.866	0.671	0.762

5 Conclusion and discussion

This research aimed to identify the best way to select the features from a huge number of tweets according to the word association with the events of interest. We developed a comparative study to compare the impact of the different word forms, word counts and correlation methods on the selected features. To evaluate the quality of the selected features, we used them to train Naive Bayes classifier and test how good the classifier will be able to classify the event/nonevent days.

The classification results, including the AUC and the F-score, emphasized that the best feature selection combination is Spearman's correlation method with skip-gram word form and two as the word count, as stated in Table 9. Despite these results, we would recommend to use the bag-of-words for event detection as it achieves similar results to skip-grams, while being more feasible computationally. This is because bags-of-words generate less features per tweet and it is easier to be aggregated as explained in Sects. 3.2 and 3.3.

The pipeline proposed in this research is applied to multiple cities using the selected feature selection combination (Spearman + skip-gram + two words) and achieved relatively close scores as listed in Table 10. The results using the selected features varied according to each city nature such as the population, the activities, the economy and the climate. Due to the temporal nature of the event detection problem, our observed dataset is limited to 641 days for all cities. The key factors affecting the results for each city are the number of events happening in the city in association with the number of tweets coming from each city.

For example, Melbourne is an event-rich city that has at least one civil unrest event weekly and has on average 4 million tweets daily, which makes correlating the features with the events pretty informative using the aforementioned combination. On the other hand, Brisbane has limited population, limited tweets and a small number of civil unrest events due to the nature of the city as a tourist attraction and its all-year good climate; these factors made the feature correlation less accurate due to the lack of events. Jakarta achieved less scores than Melbourne, despite the fact that Jakarta is more reached by people and tweets, but this is because we applied the same pipeline on the Indonesian language without any natural language preprocessing such as stemming, lemmatization or morphological analysis.

This work is useful for detecting historical events as well as live events such as protests, sports events, criminal events and accidents. The selected features can also be used for predicting future events according to the historical patterns of the selected features, especially the events that involve a mass volume of people such as protests or the vents that are affected by people impressions such as stock market trends.

Acknowledgements This research was fully supported by the School of Mathematical Sciences at the University of Adelaide. All the data, computational framework and technical support were supported by Data-To-Decision-Collaborative-Research-Center (D2DCRC)

References

Abdelhaq H, Sengstock C, Gertz M (2013) Eventweet: online localized event detection from twitter. Proc VLDB Endow 6(12):1326–1329

Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K (2017) A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919

Ayache A, Cohen S, Véhel JL (2000) The covariance structure of multifractional brownian motion, with application to long range dependence. In: Acoustics, speech, and signal processing, 2000. ICASSP'00. Proceedings. 2000 IEEE international conference on, vol 6, pp. 3810–3813. IEEE

Azzam A, Tazi N, Hossny A (2017) A question routing technique using deep neural network for communities of question answering. In: International conference on database systems for advanced applications. Springer, pp 35–49

Baker LD, McCallum AK (1998) Distributional clustering of words for text classification. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 96–103

Benesty J, Chen J, Huang Y, Cohen I (2009) Pearson correlation coefficient. In: Noise reduction in speech processing, Springer, pp 1–4

Blumenstock JE (2008) Size matters: word count as a measure of quality on wikipedia. In: Proceedings of the 17th international conference on World wide web. ACM, pp 1095–1096

Carley KM (2003) Dynamic network analysis. na

Cataldi M, Di Caro L, Schifanella C (2010) Emerging topic detection on twitter based on temporal and social terms evaluation. In: Proceedings of the tenth international workshop on multimedia data mining, ACM, p 4

Cheng W, Greaves C, Warren M (2006) From n-gram to skipgram to concgram. Int J Corpus Linguistics 11(4):411–433

Chien JT, Wu MS (2007) Adaptive Bayesian latent semantic analysis. IEEE Trans Audio Speech Lang Process 16(1):198–207

Church KW, Hanks P (1990) Word association norms, mutual information, and lexicography. Comput Linguistics 16(1):22–29

Cordeiro M (2012) Twitter event detection: combining wavelet analysis and topic inference summarization. In: Doctoral symposium on informatics engineering, pp 11–16

Crandall D, Cosley D, Huttenlocher D, Kleinberg J, Suri S (2008) Feedback effects between similarity and social influence in online communities. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, ACM , pp 160–168

Danowski JA, Cepela N (2010) Automatic mapping of social networks of actors from text corpora: time series analysis. In: Data mining for social network data, Springer, pp 31–46

D'hondt E, Verberne S, Weber N, Koster C, Boves L (2012) Using skipgrams and POS-based feature selection for patent classification

Diesner J, Carley KM (2004) Using network text analysis to detect the organizational structure of covert networks. In: Proceedings of the North American association for computational social and organizational science (NAACSOS) conference, vol 3. NAACSOS

Dodds PS, Harris KD, Kloumann IM, Bliss CA, Danforth CM (2011) Temporal patterns of happiness and information in a global social network: hedonometrics and twitter. PloS ONE 6(12):e26752

Dubey VK, Saxena AK (2016) Cosine similarity based filter technique for feature selection. In: Control, computing, communication and materials (ICCCCM), 2016 international conference on, IEEE, pp 1–6

Fernández J, Gutiérrez Y, Soriano JMG, Martínez-Barco P (2014) Gplsi: Supervised sentiment analysis in twitter using skipgrams. In: SemEval@ COLING, pp 294–299

Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3(Mar):1289–1305

Fraser AM, Swinney HL (1986) Independent coordinates for strange attractors from mutual information. Phys Rev A 33(2):1134

Fung GPC, Yu JX, Yu PS, Lu H (2005) Parameter free bursty events detection in text streams. In: Proceedings of the 31st international conference on very large data bases, VLDB Endowment, pp 181–192

Guthrie D, Allison B, Liu W, Guthrie L, Wilks Y (2006) A closer look at skip-gram modelling. In: Proceedings of the 5th international conference on language resources and evaluation (LREC-2006), sn, pp 1–4

Guzman J, Poblete B (2013) On-line relevant anomaly detection in the twitter stream: an efficient bursty keyword detection model. In: Proceedings of the ACM SIGKDD workshop on outlier detection and description, ACM, pp 31–39

Hauke J, Kossowski T (2011) Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data. Quaest Geogr 30(2):87

Havlicek LL, Peterson NL (1976) Robustness of the pearson correlation against violations of assumptions. Percept Mot Skills 43(3-suppl):1319–1334

Hazewinkel M (2001) Orthogonalization. Encyclopedia of mathematics. Kluwer Academic Publishers, 2002, Dordrecht

Hewapathirana IU, Lee D, Moltchanova E, McLeod J (2020) Change detection in noisy dynamic networks: a spectral embedding approach. Soc Netw Anal Mining 10(1):14

Hossny A, Shaalan K, Fahmy A (2008) Automatic morphological rule induction for arabic. In: Proceedings of the workshop on human language translation and natural language processing within the arabic world (LREC08), pp 97–101

Hossny A, Shaalan K, Fahmy A (2009) Machine translation model using inductive logic programming. In: 2009 International conference on natural language processing and knowledge engineering, IEEE, pp 1–8

Hossny AH, Moschuo T, Osborne G, Mitchell L, Lothian N (2018) Enhancing keyword correlation for event detection in social networks using svd and k-means: twitter case study. Soc Netw Anal Min 8(1):49

Khafaei T, Taraghi AT, Hosseinzadeh M, Rezaee A (2019) Tracing temporal communities and event prediction in dynamic social networks. Soc Netw Anal Min 9(1):59

Kim C, Park S, Kwon K, Chang W (2012) An empirical study of the structure of relevant keywords in a search engine using the minimum spanning tree. Expert Syst Appl 39(4):4432–4443. https://doi.org/10.1016/j.eswa.2011.09.147. http://www.sciencedirect.com/science/article/pii/S0957417411014709

Koyejo OO, Natarajan N, Ravikumar PK, Dhillon IS (2014) Consistent binary classification with generalized performance metrics. In: Advances in neural information processing systems, pp 2744–2752

Kruskal WH (1958) Ordinal measures of association. J Am Stat Assoc 53(284):814–861

Kurihara K, Sato T (2006) Variational Bayesian grammar induction for natural language. In: International colloquium on grammatical inference, Springer, pp 84–96

Lampos V, Cristianini N (2012) Nowcasting events from the social web with statistical learning. ACM Trans Intell Syst Technol (TIST) 3(4):72

Landauer TK (2006) Latent semantic analysis. Wiley Online Library, New Jersey

Lawrence I, Lin K (1989) A concordance correlation coefficient to evaluate reproducibility. Biometrics pp 255–268

Levy O, Goldberg Y (2014) Dependency-based word embeddings. ACL 2:302–308

Li R, Lei KH, Khadiwala R, Chang KCC (2012) Tedas: a twitter-based event detection and analysis system. In: Data engineering (ICDE), 2012 IEEE 28th international conference on, IEEE, pp 1273–1276

Li R, Zhong W, Zhu L (2012) Feature screening via distance correlation learning. J Am Stat Assoc 107(499):1129–1139

Loper E, Bird S (2002) NLTK: The natural language toolkit. In: Proceedings of the ACL-02 workshop on effective tools and methodologies for teaching natural language processing and computational linguistics—vol 1, ETMTNLP '02. Association for computational linguistics, Stroudsburg, PA, USA, pp 63–70. https://doi.org/10.3115/1118108.1118117

Mandera P, Keuleers E, Brysbaert M (2017) Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: a review and empirical validation. J Mem Lang 92:57–78

Mathioudakis M, Koudas N (2010) Twittermonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD International conference on management of data, SIGMOD '10, ACM, Indianapolis, Indiana, USA pp 1155–1158 https://doi.org/10.1145/1807167.1807306

Matsuo Y, Mori J, Hamasaki M, Nishimura T, Takeda H, Hasida K, Ishizuka M (2007) Polyphonet: an advanced social network extraction system from the web. Web Semant Sci Serv Agents World Wide Web 5(4):262–278

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

Myers L, Sirois MJ (2006) Spearman correlation coefficients, differences between. Wiley StatsRef, Statistics Reference Online

Nasution MK, Noah SAM, Saad S (2016) Social network extraction: superficial method and information retrieval. arXiv preprint arXiv:1601.02904

Niwattanakul S, Singthongchai J, Naenudorn E, Wanapu S (2013) Using of jaccard coefficient for keywords similarity. In:

Proceedings of the international multiconference of engineers and computer scientists, vol 1

Pennacchiotti M, Gurumurthy S (2011) Investigating topic models for social media user recommendation. In: Proceedings of the 20th international conference companion on World wide web, ACM, pp 101–102

Petrović S, Osborne M, Lavrenko V (2010) Streaming first story detection with application to twitter. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics, HLT '10 . Association for computational linguistics, Stroudsburg, PA, USA, pp 181–189 . http://dl.acm.org/citation.cfm?id=1857999.1858020

Popescu AM, Pennacchiotti M (2010) Detecting controversial events from twitter. In: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM, pp 1873–1876.

Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, ACM, pp 851–860

Sayyadi H, Hurst M, Maykov A (2009) Event detection and tracking in social streams. In: ICWSM

Shazeer N, Pelemans J, Chelba C (2015) Sparse non-negative matrix language modeling for skip-grams. Proc Interspeech 2015:1428–1432

Singhal A (2001) Modern information retrieval: a brief overview. IEEE Data Eng Bull 24(4):35–43

Székely GJ, Rizzo ML, Bakirov NK et al (2007) Measuring and testing dependence by correlation of distances. Ann Stat 35(6):2769–2794

Székely GJ, Rizzo ML et al (2009) Brownian distance covariance. Ann Appl Stat 3(4):1236–1265

Thelwall M, Buckley K, Paltoglou G (2011) Sentiment in twitter events. J Assoc Inform Sci Technol 62(2):406–418

Unankard S, Li X, Sharaf MA (2015) Emerging event detection in social networks with location sensitivity. World Wide Web 18(5):1393–1417

Viola P, Wells WM III (1997) Alignment by maximization of mutual information. Int J Comput Vis 24(2):137–154

Wallach HM (2006) Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd international conference on machine learning, ACM, pp 977–984

Walther M, Kaisser M (2013) Geo-spatial event detection in the twitter stream. In: ECIR, Springer, pp 356–367

Wells WM, Viola P, Atsumi H, Nakajima S, Kikinis R (1996) Multimodal volume registration by maximization of mutual information. Med Image Anal 1(1):35–51

Weng J, Lee BS (2011) Event detection in twitter. ICWSM 11:401–408

Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. ICML 97:412–420

Zhang H, Li D (2007) Naïve bayes text classifier. In: Granular computing, 2007. GRC 2007. IEEE international conference on, IEEE, pp 708–708

Zhang J, Ackerman MS, Adamic L (2007) Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th international conference on World Wide Web, ACM, pp 221–230

Zywica J, Danowski J (2008) The faces of facebookers: investigating social enhancement and social compensation hypotheses; predicting facebook and offline popularity from sociability and self-esteem, and mapping the meanings of popularity with semantic networks. J Comput Mediat Commun 14(1):1–34

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.