Proceedings of the Tenth International AAAI Conference on
Web and Social Media (ICWSM 2016)

Sensing Real-World Events Using Arabic Twitter Posts

Nasser Alsaedi, Pete Burnap, Omer Rana
School of Computer Science
& Informatics, Cardiff University, UK
{AlsaediNM, BurnapP, RanaOF}@cardiff.ac.uk

Abstract

In recent years, there has been increased interest in event detection using data posted to social media sites. Automatically transforming user-generated content into information relating to events is a challenging task due to the short informal language used within the content and the variety of topics discussed on social media. Recent advances in detecting real-world events in English and other languages have been published. However, the detection of events in the Arabic language has been limited to date. To address this task, we present an end-to-end event detection framework which comprises six main components: data collection, pre-processing, classification, feature selection, topic clustering and summarization. Large-scale experiments over millions of Arabic Twitter messages show the effectiveness of our approach for detecting real-world event content from Twitter posts.

Introduction

People tend to comment on real-world events when a topic suddenly catches their attention, for example, a sporting event, adverse weather update, terror attack, etc. Identifying events from social media presents several challenges. The heterogeneity and immense scale of the data are key challenges. This is compounded by the fact that each social media post is short in length, which means that only a limited content is available for analysis. Other challenges are inherent to the microblogging language and nature which include the frequent use of informal, irregular, and abbreviated words, the large number of spelling and grammatical errors, and the use of improper sentence structure and mixed language. Additionally, social media characteristics and popularity have attracted spammers to spread advertisements, pornography, viruses and other malicious activities (Becker, Naaman, and Gravano 2011); (Atefeh and Khreich 2015); (Imran et al. 2015).

Our focus in this work is on real-world event identification using Arabic content posted on Twitter. Arabic is a rich Semitic language which is highly productive, both derivationally and inflectionally. Arabic poses many challenges for data mining tasks where most of these challenges are due to orthography and morphology. It is true that some of these challenges are shared with other languages but it exhibits considerable complexity from theoretical to computational linguistics. Furthermore, the language processing becomes even more challenging when considering the language used in social media sites, where dialects are heavily used (Alsaedi and Burnap 2015).

Many researchers have proposed models and techniques for the purpose of identifying events in social media in general including unsupervised learning (Becker, Naaman, and Gravano 2011), signal processing techniques (Weng and Lee 2011), topic model using Latent Dirichlet Allocation (LDA) (Pan and Mitra 2011), graph models (Sayyadi and Raschid 2013) and many more (Petrović, Osborne, and Lavrenko 2010); (Walther and Kaisser 2013); (Schulz, Schmidt, and Strufe 2015). However, none of these approaches perform particularly well with the Arabic content.

Our focus in this paper is on the real-world event identification using Arabic content. We identify each event and its associated Twitter messages using an integrated classification-clustering system that groups together topically similar tweets. We then compute three revealing features (temporal, spatial and textual features) for each cluster to help determine which clusters correspond to events. Finally, we automatically select the top post (or the most representative posts) that being discussed within clusters. We validate the effectiveness of our framework using a dataset of over 16 million Arabic Twitter messages.

Problem Definition

Definition Events in social media are real world happenings that are reflected by change in the volume of text data that discusses the associated topic at a specific time.

Consider a text stream $D = (D_1, D_2, ..., D_n)$ where D_i is a document, and the length of D is $|D|$. A document d_i consists of a set of features,$(F_1, F_2, ..., F_k)$, and is reported at time t_i. In the text stream D, $t_i \leq t_j$ if $i < j$. Dividing the text stream, D, into time windows, W_i of the same length, per day, per 12 hours, ... , per minute. The problem of real-time event detection is a problem to find a set of events per unit time, where an event consists of a minimal set of features, in time windows $W_i, W_j, ...$ that together identifies the event with the largest number of documents that contain the similar features. Figure 1 illustrates the relationship between events and the different sets of features.

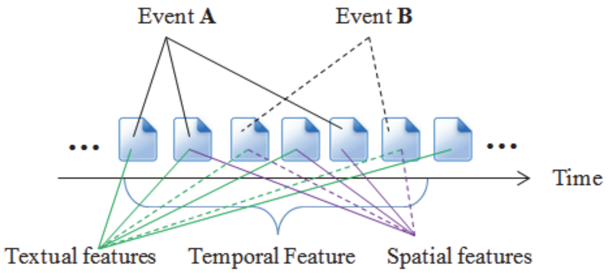

Figure 1: Document clustering using various sets of features

System Design

As we receive high volume of posts per hour with wide variety of tweets, traditional monitoring is impractical. Our proposed framework is based on collecting a series of data over time windows for a given location which supports the auto-identification of meaningful events from Twitter.

Data Collection

We collect user-generated updates directly from Twitter Streaming API as it allows subscription to a continuous live stream of data. Our goal is to detect events in a given location without prior knowledge of these events. Thus, we collect tweets based on a set of keywords that describes a region (e.g., Iraq, Syria, Egypt, ...) using Arabic language. We also collect user updates from users who selectively add the required region as their location. We also make use of geographic Hashtags in the data collection process (e.g., #Ramadi, #Aleppo, #Cairo, ...). Data are stored temporarily using MongoDB database (www.mongodb.org).

Pre-Processing

The goal of this step is to represent data in a form that can be analyzed efficiently. We perform text processing techniques such as stop-word elimination (Term frequency and TF-IDF are the criteria used for classifying stop words) and stemming (an updated version of Khoja stemmer (Diab, Hacioglu, and Jurafsky 2004)) for Arabic text. To the Arabic stop word list included in the Khoja stemmer, we added more stop words which are determined using Term frequencies and the TF-IDF of the training corpus. Moreover, posts that were less than 5 words long were removed, as were messages where over half the total words were the same word, since these posts were less likely to have useful information.

Classification

This step aims to distinguish events from noise or irrelevant posts. The early classification dramatically reduces the number of posts to be processed in the following steps because these steps will process only event-related updates. Words from each status are considered as features and Naive Bayes classifier was chosen for the classification task.

The features and their corresponding category (event or non-event) were provided to the classifier, constituting the training set. From the training data the likelihood of each post belonging to either class is derived on the basis of the occurrence of the post's features in the training data. When a new example is presented, the class likelihood for the unseen data is predicted on the basis of the training instances.

Algorithmic steps:

1. Input posts.
2. Extract features from posts.
3. These features and their corresponding labels are used to train the learning algorithm (Naive Bayes classifier).
4. New posts are presented to the trained classifier to predict their label according to their extracted features.

Feature Selection

Different events can be characterized by different set of features: temporal, spatial and textual features. These features are computed for the Online-Clustering. The temporal features are related to the "speed" of diffusion over time by highlighting the "quality" of posts created by users in different time frames. The spatial features include Neighbourhood (Local) granularity, City (Intermediate) granularity and Country (General) level. Textual features consist of many content representative features such as Near-Duplicate measure, Retweet ratio, Mention ratio, Hashtag ratio, Url ratio, Text sentiment and Dictionary features. A full description of these features is given in (Alsaedi, Burnap, and Rana 2015).

Temporal features Temporal features are important factors that have been considered in many data mining studies including event identification. The volume of posts and the continually updated commentary around an event suggest that informative posts from several hours ago may not be as important as new posts (Becker, Naaman, and Gravano 2011). For this reason we retain the most frequently occurring terms in a cluster in hourly time frames and compare the number of posts published in an hour that contain term t to the total number of posts published during that hour.

Spatial features (geospatial, regional) Events are characterized by a rich set of spatial and demographic features. We make use of three statistical location approaches; the first one is from Twitter itself where the source latitude and longitude coordinates are extracted (if provided by the user). The second method depends on shared media (photos and videos) by using the GPS coordination of the capture device (if supported). Third, OpenNLP (http://opennlp.sourceforge.net) and Named-Entity Recognition (NER) are implemented for geotagging the tweet content (text) to identify places, street names, landmarks, etc.

Textual features Textual or content features have been identified as contributing to the spread of a post in the social media. Most of these features have been widely addressed in the data mining literature including; Near-Duplicate measure (Walther and Kaisser 2013), Retweet ratio (Becker, Naaman, and Gravano 2011), Mention ratio (Becker, Naaman, and Gravano 2011), Hashtag ratio (Imran et al. 2015), Link or Url ratio (Atefeh and Khreich 2015), Text sentiment (Imran et al. 2015), Dictionary-based feature (Walther and

Kaisser 2013). The dictionary-based feature uses a dictionary of trigger words such as present tense verbs, popular event nouns and adjectives to characterize real-time events.

Online-Clustering

The classification step separates event-related tweets from non-event posts such as chats, personal updates, etc. Non-event posts are filtered. To identify the topic of an event, we define temporal, spatial and textual set of features, which are detailed in the previous section. We then apply an online clustering algorithm, which is outlined in **Algorithm 1**.

Using a set of features $(F_1, ..., F_k)$ for each document $(D_1, ..., D_n)$ we compute the cosine similarity measure between the document and each cluster $(C_1, ..., C_k)$ where the similarity function is computed against each cluster c_j in turn for j = 1, . . . , m and m is the number of clusters (initially $m = 0$). We use **the average** weight of each term across all documents in the cluster to calculate the centroid similarity function $E(D_i, c_j)$ of a cluster. The threshold parameters are determined empirically in the training phase.

Algorithm 1: Online Clustering Algorithm

```
Input  : $n$ set of documents $(D_1, ..., D_n)$
         Threshold $\tau$
Output: $k$ clusters $(C_1, ..., C_k)$
while $\tau$ is given do
    compute the centroid similarity function E $(D_i, c_j)$
    of each cluster $c_j$ ;
    if centroid similarity E $(D_i, c_j) \geq \tau$ then
        1) A new cluster is formed containing $D_i$ ;
        2) The new centroid value = $D_i$ ;
    else
        1) Assign it to the cluster which gives the
        maximum value of E $(D_i, c_j)$ ;
        2) Add $D_i$ to cluster $j$ and recalculate the new
        centroid value $c_j$ ;
    end
end
```

Summarization

After grouping documents (tweets) into different clusters, the next natural step is to automatically summarize and represent topics being discussed in these clusters. Each cluster may contain hundreds of posts, images or videos, and the task of finding the most representative update would indubitably save the decision maker's time and effort.

The temporal Term Frequency - Inverse Document Frequency (TF-IDF) we propose here generates a summary of top terms without the need of prior knowledge of the entire dataset unlike popular TF-IDF approach (Salton and Buckley 1988) and its variants. The temporal TF-IDF is based on the assumption that words which occur more frequently across documents over a particular interval (timeframe) have a higher probability of being selected for human created multi-document summaries than words that occur less frequently (Salton and Buckley 1988). The temporal TF-IDF considers a set of tweets in a cluster for each timeframe to be represented as a document. The total number of clusters equals the total number of documents which is a sub-set of the entire dataset. This reduces the overall computational complexity and overcomes the limitations of the TF-IDF based approaches. We define the temporal TF-IDF weighting scheme of a new document d for a collection C as:

$$w_{ji} = \frac{1}{norm(d_i)} f_{ji} \times log(1 + \frac{N}{N_j}) \qquad (1)$$

where f_{ji} is the term frequency of word in document d_i and N_j is document frequency of word in a collection and N is the total number of documents in the collection. Therefore, this summarizer selects the most weighted post as summary as determined by the Temporal TF-IDF weighting.

Experiments

Experimental Settings

Dataset: Our dataset consists of over 16 million Arabic tweets (16,101,284) and was collected from 1 October 2015 until 30 November 2015 using Twitter's Streaming API. Since we are interested in identifying events in Arabic language, we limited ourselves to tweets published in Arabic-speaking countries.

Annotations: We use human annotators to label classes for classification and clusters for the online clustering (training and testing phases). For complete details of our annotation guidelines, please refer to (Alsaedi and Burnap 2015).

(a) *Classification*: From our collected data, three annotators manually labelled 5000 Arabic tweets in to two classes "Event" and "Non-Event" to train our Naive Bayes Classifier. Event instances outnumbered the non-event ones as the training set consisted of 1900 Non-Event tweets and 3100 event-related posts. Agreement between our three annotators, measured using Cohen's kappa coefficient, was substantial (kappa = 0.825). A ten-fold cross validation approach was used to train and test the Classifier.

(b) *Clustering*: We use the data collected in October for training and report our results on test data from November. For the training set, we employed three human annotators to manually label 800 clusters, randomly selected from the top-20 fastest-growing clusters according to hourly message volume at the end of each hour in October 2015. Similarly for the testing set, three human annotators labelled 800 clusters, randomly selected from the top-20 fastest-growing clusters according to hourly message volume at the end of each hour in November 2015. The agreement between annotators was calculated using Cohen's kappa (kappa = 0.791), which indicates an acceptable level of agreement.

Evaluation To evaluate the performance of the event identification task, we implement two well-known information retrieval metrics, namely, $Precision@K$ and *NDCG*. $Precision@K$ reports the fraction of correctly identified events out of the top-K selected clusters, averaged over all hours. Where as the normalized discounted cumulative gain *NDCG* metric ranks the top events relative to their ideal ranking as well as *NDCG* supports graded judgments and rewards relevant documents in the top ranked list.

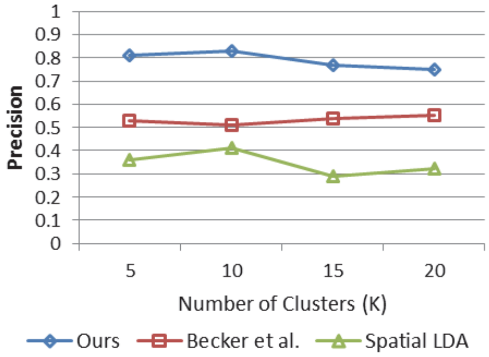

Figure 2: $Precision@K$ for event identification methods.

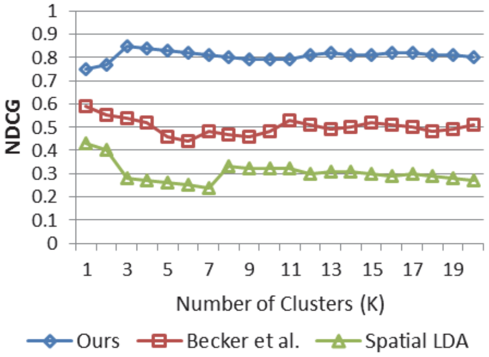

Figure 3: *NDCG* at K for three event identification methods.

Experimental Results

We compare the output of our framework over strong methods (Spatial LDA (Pan and Mitra 2011) and unsupervised [Becker et al.2011] method) using $Precision@K$ and *NDCG* measures. Figures 2 and 3 show the results of various event identification methods using our test corpus.

According to Figures 2 and 3, our proposed framework is effective and outperforms other approaches both in the *NDCG* and $Precision@K$ evaluation measures. In fact our framework discovers much more real-world events than other approaches such as the refugee crisis, disasters and terrorist attacks (e.g. Paris attacks, Beirut bombings, etc.), war against ISIS as well as many other events and stories.

In general, Latent Dirichlet Allocation (LDA) is known to achieve promising results in modeling text collections such as news articles. However, it fails to achieve such results in our case due to the fact that tweets are short and a collection of tweets per hour may contain many more topics than in news articles. This explains why the (Pan and Mitra 2011) approach performance is worse than the results reported in this paper. In addition, the poor performance of (Becker, Naaman, and Gravano 2011) model has reasonable explanations; first the summarization method performs poorly with a morphologically rich language such as Arabic. Secondly, classification after clustering has a crucial impact on performance with short time windows as many of the small clusters are filtered out since they do not exceed the predefined thresholds and are considered non-relevant events (noise). This eliminates many of them together with noise, which confuses the scoring and ranking of event detection.

Conclusion

In this paper we have presented an integrated framework for detecting real-world events reported in Arabic on Twitter. The event identification was performed in several stages: data collection, preprocessing, classification, feature selection, clustering and summarization. Our experiments suggest that our framework yields better performance than many leading approaches in the real-time event detection. This work can be used in event management, intelligence gathering and decision-making including understanding and tracking terrorist groups, such as Al-Qaeda and ISIS. In future work, we aim to investigate more features, such as network features and rank these features. The detection of rumors in Arabic microblogs will also be considered.

References

Alsaedi, N., and Burnap, P. 2015. Arabic event detection in social media. In *CICLing'15*, 384–401.

Alsaedi, N.; Burnap, P.; and Rana, O. 2015. Identifying disruptive events from social media to enhance situational awareness. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Atefeh, F., and Khreich, W. 2015. A survey of techniques for event detection in twitter. *Comp. Int.* 31(1):132–164.

Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond trending topics: Real-world event identification on twitter. In *ICWSM'11*.

Diab, M.; Hacioglu, K.; and Jurafsky, D. 2004. Automatic tagging of arabic text: From raw text to base phrase chunks. In *NAACL'04*.

Imran, M.; Castillo, C.; Diaz, F.; and Vieweg, S. 2015. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys* 47(4):1–67.

Pan, C., and Mitra, P. 2011. Event detection with spatial latent dirichlet allocation. In *JCDL'11*.

Petrović, S.; Osborne, M.; and Lavrenko, V. 2010. Streaming first story detection with application. In *NAACL'10*.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24(5):513–523.

Sayyadi, H., and Raschid, L. 2013. A graph analytical approach for topic detection. *ACM Transactions on Internet Technology* 13(2):4:1–4:23.

Schulz, A.; Schmidt, B.; and Strufe, T. 2015. Small-scale incident detection based on microposts. In *HT'15*.

Walther, M., and Kaisser, M. 2013. Geo-spatial event detection in the twitter stream. In *ECIR'13*.

Weng, J., and Lee, B. 2011. Event detection in twitter. In *ICWSM'11*.