Detecting Significant Events in Arabic Microblogs using Soft Frequent Pattern Mining

Jehad H. Zendah, Ashraf Y. Maghari

Abstract— nowadays, people use microblogs as a main platform to write about events that occur in their environment. Many researches have been conducted for event detection on the English language, but, Arabic context has not received much research. Furthermore, existing approaches rely on platform dependent features such as hashtags, mentions, or retweets, which make their approaches less efficient when these features are not presented. Further, some approaches which depend only on bursty or frequently used words, detect general viral topics instead of event related topics. In this work, we present a new approach for detecting events written in Arabic using frequent event triggers. The approach first identifies the part of speech tags of a sentence and then analyze them to extract event triggers. A soft frequent pattern mining method is applied to find co-occuring event triggers. The approach has been evaluated using a subset of the Evetar dataset. We divided the data into timely constrained windows to mimic the data stream behavior. Two experiments of different time intervals were conducted, 6-hours and one-day time intervals. We achieved an average F-meaure of 0.644 and 0.717. The results show that our approach outperformed some widely known approaches and it was comparable with others.

Index Terms—Event Detection, Event Trigger, Soft Frequent Pattern Mining.

I INTRODUCTION

Nowadays, Microblogs have become the main virtual environment for connecting people and sharing digital content. It allows users to post and share short text, images, and/or short length videos of any type. This content is delivered to a network of followers or virtual friends of the content creator at relatively no time. Content type depends on the user interests and situation. At the occurrence of an event, users post details about the event with their friends. With the instant delivery, events news usually spread faster and reach wider audience in microblogs compared to mainstream media [1].

Events are real-world occurrences that take place in a certain geographical location over a certain time period [2]. Capturing information about an event can help in many aspects. For example, it can help in accelerating the crisis response when the information about disastrous events are retrieved at the time of its occurrence, also, it can help people to easily track occurring events.

In Microblogs context, Dou, W., et al. [3] defined an event as "An occurrence causing change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location". Event Detection (ED) definition depends on the task held by the researcher. In general, ED is the process of discovering and identifying new or previously unidentified events from a set of documents [4].

Event detection from text has been addressed by many researchers; they employed different techniques from many fields such as machine learning, data and text mining, and natural language processing. The first event detection research program is the Topic Detection and Tracking (TDT) conducted by [2]. The techniques introduced in TDT are used to monitor different newswire sources so that users can be aware of occurring events. They were applied on full text of well written news articles, however, with the emergence of microblogs, new challenges are introduced when using these techniques.

For example, content generated by microblogs users is constrained to be very small, thus if the traditional Term Frequency-Inverse Document Frequency (TF-IDF) is used in such short documents, it will result in a sparse vector issue. Microblogs posts are very noisy source of data, according to [5], microblogs' posts do not always refer to an actual event or a subject of importance, but most of the time the posts contain meaningless or uninteresting daily life activity. In addition, the content is generated by anyone, thus, grammatical errors, informal words, or incomplete context will be introduced.

Event detection techniques are classified based on the event type, the detection task, and detection method [6]. The type of event can be specified or unspecified, where the specified event detection techniques require prior knowledge about the event –mainly a related keyword or a query. On the other hand, unspecified event detection techniques do not require any prior knowledge, such techniques require an ongoing monitoring and analysis for the incoming documents to find an increase number of keyword appearance-count, which can be an indicator of a potential event. Based on the detection task, the detection process can be used for new or retrospective events.

Many existing approaches of event detection are tested on

English data. For the best of our knowledge only few researches have been conducted for the Arabic Language [7]. The Arabic language introduces many challenges in the Text mining and Natural Language Processing (NLP) fields, this is due to its vocabulary richness, and its morphological and orthographic nature [8]. These challenges are inherited in the event detection problem. Other existing approaches utilize platform specific features such as hashtags, retweets, and followers and external knowledge to enhance event detection [9, 10]. The problem with these approaches is that when these attributes do not appear for different reasons in the process, the accuracy of the event detection will be affected. In addition to these limitations, most of the approaches found in the literature depend on the burst behavior of specific words, but it is not true that every word that shows burst is related to an event. For example, in the Arabic context, users usually post praising words to Allah such as "Subhan Allah Wa Behamdeh" "سبحان الله وبحمده". These words will sometimes show bursts, but in reality, they do not belong to any event.

In this paper we propose an approach for detecting significant events from Arabic microblog posts that tackle these challenges. Our approach relys on extracting event triggers from post text using pre-defined rules applied to the Part of Speech (POS) tags of the tweet. This process is essential to separate posts that may contain event occurrence from posts that do not. A Soft Frequent Pattern Mining Approach is applied to the posts containing event trigger. The resulted cluster are treated as detected event.

The proposed approach targets the Arabic content to be a first stage for early event reporting, situational awareness and summarization for Arabic audience. Arabic event detection can be very useful in crisis response applications, as many places in the Middle East have conflicts, thus different types of events can occur.

The rest of this paper is organized as follows: section 2 reviews the event detection techniques. Section 3 explains the methodology, section 4 demostrates the experements and the evaluation, and the conclusion is presented in section 5.

II LITERATURE REVIEW

Many approaches of event detection in microblogs use platform specific features which make the approach accuracy dependent on these features and cannot be ported to other platforms. Also, other approaches focus on finding a burst of words in the stream to identify hot topics. These approaches detect a topic without any consideration if that topic is an actual event or a viral general topic. In addition to that, considerable research has been done and tested on English data but only few consider the Arabic language. In the following sections we review some of these researches.

Alsaedi, N. and P. Burnap [7] proposed a novel Arabic event detection framework from Twitter dataset. In the framework, the data undergoes a preprocessing step to enhance the data quality. A Naïve Bayes classifier is used to distinguish event-related tweets from irrelevant tweets. The classifier is trained on 1500 tweets and their terms are used as features. Tweets are represented by a set of features which include temporal, spatial, and textual features such as retweet ratio, mention ratio, hashtag ratio, tweet sentiment, etc. Tweets are then clustered together to distinguish events using an online clustering algorithm. The average weight of each term in all document in a cluster is used as the centroid of the cluster. The approach output was evaluated by splitting the dataset into days and then calculate the average value of precision which was 80.24% for disruptive events. Our approach uses a same schema by filtering tweets before further processing.

Using the terms in a small dataset to train the classifier will have two drawbacks as stated in [11], first, it could decrease classifier accuracy as the keywords introduced in the dataset are subjective to specific events and can lead to undesirable result when new events emerges, second, using the bag of word will result in vector sparseness especially when used in dynamic and rapid changing corpus. Our approach, however, uses a set of rules that examine the tweet's syntax and determine if it contains an event trigger or not. Tweets that do not have event trigger are filtered out.

In [12] an approach for event detection using multimodal factor analysis model is proposed. The approach depends on two feature sets. The hashtags' bag of words created from all the tweets containing the hashtag and geolocation vector containing the latitude and longitude values of all tweets containing the hashtag. A probabilistic generative modal is used to fuse these features and an expectation maximization algorithm is derived for finding the maximum likelihood estimates of the model parameters. The approach assumes that hashtags are used during event occurrence and tweets containing these hashtags are geographically closed together.

In [13] an unsupervised approach for event extraction out of Arabic tweets is presented. The approach tags the event expression and the related entities and link them to the knowledge base. Event expression are identified using a set of rules based on the guidelines provided by the Arabic Annotation Guidelines for Events (Consortium, 2005). This approach processes each tweet independently from the other, thus it will fail to identify significant events. Our approach works on the burst behavior of event triggers, thus only trending/significant events are detected.

In [9] an approach based on a multiple assignment graph partitioning algorithm is introduced where event is represented by a cluster of related words. The authors address the problem of message posting delays which lead to event attributes being scattered in different timestamps, thus the significance of event-related words will be decreased as time goes on. Words are modeled using three Twitter-based information theoretic metrics. The first metric is Conditional Word Tweet Frequency-Inverse Trend Word Tweet Frequency (CWTF-ITWTF), which is a time varying measurement similar to the popular IDF-TF. The objective of this measurement is to decrease the weight of trendy ongoing events.

The second metric is Word Frequency-Inverse Trend Word Frequency (WF-ITW), which is a time varying measure that consider the frequency of keywords. Lastly, Weighted Conditional Word Tweet Frequency-Inverse Trend Weighted Conditional Word Tweet Frequency (WCWTF-ITWCWTF). This measure depends on features from Twitter such as the number of followers and the number of retweets to find the importance of a keyword. A fuzzy time series signal is produced from the three metrics. The approach is evaluated over a manually collected dataset using Twitter stream API. A time window of size 6 hours is used over a month time horizon. To create a ground truth, for each time window the most frequent keywords are extracted and presented to experts to annotate them. The evaluation measurements used are keyword recall/precision and F1-Score. This approach cannot be used in microblogging streams that do not produce these features, however, our approach does not consider any Twitter based features. It also depends only on the bursty pattern of a set of keywords thus the resulted detected event can be a trending topic and not a real-world event. The approach requires different parameter tuning to achieve good F1-Score.

In [14] a generic framework for event detection that depends on dynamic multivariate graph is presented. A user-to-user undirected graph is built where vertices are represented as users, and edges are represented as the follow relationship between the users. Every vertex contains a textual feature vector of domain-specific keywords. The approach focuses on the search of evolving subgraphs over time with anomalous features. The evaluation metrics used are false positive rate (FPR), true positive rate (TPR).

Other approaches depend on frequently used words as the approach presned in [10] depends on clustering wavelet signals. A signal is built for each word using wavelet analysis to reduce space and storage. Auto-correlation is calculated for each signal. Signals that produces skewed auto-correlation are identified as insignificant words and then removed. Similarity between words is calculated using the cross-correlation between every pair of words. Similar words are determined using a threshold value where words that have a similarity higher than the threshold are clustered together. The approach is evaluated over a manually collected dataset from Twitter where non-English characters are removed from the text. The evaluation is conducted manually considering only precision, as the authors cannot enumerate all events that occurred at the time of collection, thus recall is discarded. Different experiments with different configuration are used and the best result achieved was 16.7%. Clustering based on only a pair of words will produce generic events or topics [15]. For example, if a bombing event occurred in the same country with different locations, then using such algorithm will detect event that does not differentiate between the two different locations.

In [15] a novel method called Soft Frequent Pattern Mining (SFPM) is introduced. The method is used to tackle the problem of using patterns of only pairs of terms for event detection. The method uses the same concept of the Frequent Pattern Mining (FPM) technique in which it examines the simultaneous co-occurrence patterns of degree greater than two, however, it is less strict than FPM as it does not require all terms in the pattern to be frequent in the same document, but only a large subset of the terms is frequent in same document. The approach consists of two main components. Term selection in which a fixed number of terms are selected for grouping. The selection process depends on the existence of a randomly collected tweets called reference corpus. The likelihood of appearance is estimated for each term in the reference corpus and the incoming tweets corpus. The ratio of the likelihoods of appearance is then computed. Terms with the highest ratio will be more significant as the term has a frequency higher than usual in the two corpora. The second component is the implementation of the algorithm itself on the selected top terms. Using a static reference corpus will result in a biased behavior for the term selection algorithm, as new emerging term will produce lower ratio, thus it will not be selected.

In [1] an approach based on the traditional FP-Growth method is presented. FP-Growth produces the most frequently used combinations of words that co-occur together in a tweet. Determining what is most frequent depends on a fixed threshold value. On the other hand, the approach introduces a dynamic procedure for calculating the threshold, so that it can handle the dynamic nature of words size over time. The procedure depends on a combination of statistical values which are the average and median of the words' frequencies. A preprocessing step is performed for text tokenization and removing stop words, mentions, URLs, and hashtags. A post processing step is also performed to eliminate duplicate patterns. Duplicate patterns are determined by calculating the cosine similarity between patterns with a threshold of 0.75. The approach was tested using two datasets manually collected by querying the Twitter stream API using a set of keywords that identify two events, the UK General Elections 2015 and the Greece Crisis 2015.

In [16] a multiple source approach that collect data from twitter stream and newswire websites is introduced. Every source is considered as an independent stream. Every stream undergoes two stages. First a weighted graph is built in which nodes represent words and edges represent the number of documents in which the two connected words co-occur together. A pruning process is conducted on the graph to keep emerging and important words. The multiple sources are merged by merging the pruned graph. Events are detected using voltage-based clustering algorithm on the resulted graph. The approach was tested using two sources Twitter and Tumblr and achieved F-Measure of 0.897.

The performace of the approaches that depend on platform-specific features are affected by the absence of theses features. Futhermore, the approaches that depend only on the presence of frequent keywords are very likely to detect general topic instead of real events. However, in this paper we introduce an approach that depends on the textual features of the sentence instead of platform specific features. Moreover, the approach depends on frequent event triggers which enable it to detect real events.

III METHODOLOGY

خطأ! لم يتم العثور على مصدر المرجع. shows the steps of our approach for event detection, i.e. data collection, data preprocessing, event triggers extraction (applying a set of predefined rules in the extracted Part-of-Speech tags), and finally event detection (applying The Soft Frequent Pattern Mining algorithm on the top event triggers). These steps are discussed hereafter.

A Data Collection

In this step, we collect the data that need to be analyzed. Usually, tweets are collected from the Twitter stream, but in our case, an existing dataset that support our task is used. Practically, Twitter stream cannot be analyzed at once, thus the incoming data will be processed in parts in chronological order. We call these parts timely constrained windows.

B Data Preprocessing

This is a preliminary step required for preparing the dataset. We perform tokenization on every tweet. Then a cleaning step is performed, in which we remove Latin alphabets, pecial characters, emoticons and urls. We also remove hastags with maintaining its content tweets. Tweets with less than two words are removed. After that a normalized step followed by stemming are performed to enhance Arabic words similarity. Part-of-Speech (POS) tagging is performed on every processed tweet. Lastly, stop words are removed from the dataset.

C Event Trigger Extraction

An event trigger is a term or a group of terms that represent the event itself. In our approach we use event triggers as an indicator for the occurrence of an event in a tweet. Also, it represents the important words in the text. It helps us shortening the mining process and extract real events instead of popular topics. A set of pre-defined rules are used to extract event triggers as follows:

Verb Phrase (VP)

Rule 1.1: If it contains a Verb in base form (VB), Verb in past tense form (VBD), Verb in non-3rd person singular present form (VBP) or Verb in past participle form (VBN) tag followed by a Noun (NN) tag then we consider both tags as an event trigger. As shown in Table 1

Rule 1.2: If it contains (VB), (VBD), (VBP) or (VBN)

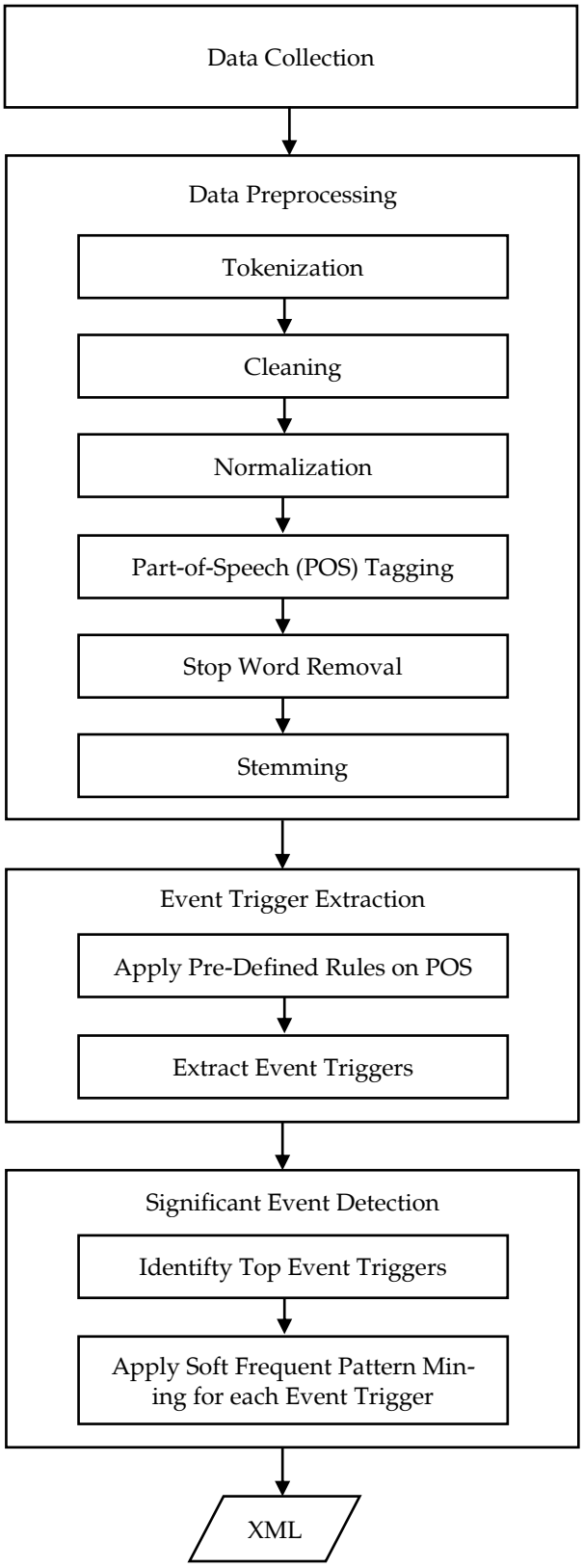

Figure 1: Steps used to detect significant Arabic events

Table 1
Example shows the extraction of event trigger using rule 1.1

Tweet	قضت محكمة الجنايات ببراءة خبير هندسي في وزارة العدل
Event Trigger	قضت محكمة
POS Tags	(DTNNS)/(**محكمة NN**)/(**قضت VBD**)/(الجنايات DTJJ)/(ببراءة NN)/(خبير JJ)/(في IN)/(وزارة NN)/(العدل DTNN)/(هندسي
Rule	1.1

Table 2
Example shows the extraction of event triggers using rule 1.2

Tweet	البحرين: ضبط مطلوبين متورطين في التفجير بالعكر الشرقي بقية الموضوع اضغط هنا
Event Trigger	ضبط مطلوبين
POS Tags	(IN)/(متورطين VN)/(**مطلوبين JJ**)/(**ضبط VBD**)/(في DTJJ)/ (بالعكر DTJJ)/ (التفجير DTNN)/ (الموضوع DTNN)/(بقية NN)/(الشرقي DTJJ)/(اضغط RB هنا)
Rule	1.2

Table 3
Example shows the extraction of event triggers using rule 2.1

Tweet	قتيلان على الأقل في اطلاق نار خلال عملية احتجاز رهائن شرق باريس
Event Trigger	إطلاق نار، عملية إحتجاز
POS Tags	(IN)/(الأقل DTNN)/(على IN)/(قتيلان VBD)/(في **NN**)/(**اطلاق NN**)/(**نار NN**)/(خلال NN)/(رهائن NN)/(**احتجاز NN**)/(**عملية** NN)/(شرق باريس NNP)
Rule	2.1

Table 4
Example shows the extraction of event triggers using rule 2.2

Tweet	ألمانيا تحذر من انقسام المجتمع عقب هجوم شارلي إبدو
Event Trigger	ألمانيا تحذر
POS Tags	(NN)/(من IN)/(**تحذر VBP**)/(**ألمانيا NNP**)/(انقسام DTNN)/(المجتمع NN)/(عقب NN)/(هجوم JJ)/(شارلي JJ)/(إبدو
Rule	2.2

tag followed by Adjective (JJ) tag then we consider both tags as an event trigger. As shown in Table 2

Rule 1.3: If the above rules does not apply then we consider (VB), (VBD), (VBP) or (VBN) tag as the event trigger.

Noun Phrase (NP)

Rule 2.1: If it contains Noun (NN) tag followed by (NN) or Singular Proper Noun (NNP) tag then we consider both tags as an event trigger. As shown in Table 3

Rule 2.2: If it contains a Noun with (NN) or (NNP) tag followed by a Verb with (VB), (VBD), (VBP) or (VBN) tag then we consider both tags as an event trigger. As shown in Table 4.

After identify all event triggers in the dataset, a list of the event triggers and their frequencies is maintained. As the approach focuses only on significant events, we assume a significant event is presented by the highest usage of event triggers combination, thus we keep event trigger of high frequencies and remove the rest. For simplicity, the average of all frequencies is used as the threshold value to determine the frequent event triggers.

D Significant Event Detection

In this step we generate groups of similar event triggers, where each group represents an event. We use an adapted version of the soft frequent pattern mining algorithm [15] to cluster event triggers that co-occur frequently in the documents. Note that the terms are not necessarily frequent in the same document.

Originally, Petkos et al. [15] extracts important words by comparing the words in dataset with a reference corpus. The problem with this solution is that maintaining a reference corpus requires a lot of efforts, also new event's related-words that do not appear in the reference corpus will be considered not important, thus identifying important words is very biased. However, in our work, we select important words by selecting event triggers using rules that depends on the part-of-speech tags.

Soft Frequent Pattern Mining (SFPM) is derived from the concept of Frequent Pattern Mining (FPM). FPM is the process of finding frequent items in a set of transactions [17]. An item is said to be frequent if its frequency is above a predefined threshold. A frequent pattern is a set of items that co-occur together in the same transaction and their frequency is above a threshold. We consider the task of event detection as a frequent pattern mining problem. When an event occurs, a group of users will start tweeting about it using similar words pattern.

We assume that an event is represented by a set of event triggers not just only one. This is due to the difference in writing style between users, which make the POS tagger produce different tags, thus different event triggers for the same event. In addition to that, the POS tagger may incorrectly tag a set of words that hold true to the rules shown. In our work, this incorrectly captured features still have high frequency in case of events, thus it does not affect the detection process.

To formulate the algorithm, suppose we have a set of tweets T of size N, and a K number of event triggers (ET) where all tweets in T contain at least one event trigger. Our task is to group similar event triggers together and retrieve their common tweets as the detected event. Thus, the objective of the algorithm is to produce a set of grouped event triggers.

Initially, every event trigger ET is treated as a single event. To be able to merge ETs, a numerical representation is needed. A vector DS of size N is calculated for each ET where Ds(n)=1 when ET appears in the nth tweet and Ds(n)=0 otherwise. The popular Cosine similarity measure is

used for comparison, where two event triggers are merged when their calculated similarity is above a threshold value. After merging the two event triggers, a vector summation is performed on both Ds vectors, and the newly created vector is assigned to the newly created group.

In a single full iteration, if no event trigger is merged, then the produced group of event triggers is treated as an event. The process is repeated for all event triggers that are not assigned to any group and the algorithm is terminated when there is no further merge. Algorithm 1 shows a pseudocode of the adapted SFPM.

Algorithm 1 : Adapted Soft Frequent Pattern Mining

```
Input: List of Event Triggers (ET)
Output: Sets of Grouped Event Triggers
FOR i=1 to ET.Count
  Calculate ET[i].Ds
END FOR
events = ET
isRepeat = TRUE
WHILE (isRepeat) Do
  Temp = events
  isAssigned = Integer Array of Size Temp.Count
  newEventSet = Empty Object
  isRepeat = FALSE
  FOR i=1 to Temp.Count - 1
    IF isAssigned [i] = 1 THEN
      Skip
    END IF
    FOR j=i+1 to Temp.Count
      IF Similarity(Temp[i].DS, Temp[j].DS)
                          >θ (|Temp[i].DS|) THEN
        isAssigned[j] = 1
        Temp[i] = Merge(Temp[i], Temp[j])
        isRepeat = TRUE
      END IF
    END FOR
    newEvents.ADD(Temp[i])
  E ND FOR
  events = newEvents
END WHILE
```

IV Experiments & Evaluation

A DataSet

To test the correctness of our approach of event detection, we need to find a dataset for such task. Evetar [18] is an Arabic dataset targeted for the event detection task. It contains a total of 590,066,789 tweets and covers 66 significant events. It has been collected in a one-month period using Wikipedia's Current Events Portal at that time. An event is represented by a set of tweets that are relevant to that event during a time period in which the event occurred. Originally, the dataset contains only the IDs of the tweets and we are tasked to fetch the actual content from Twitter.

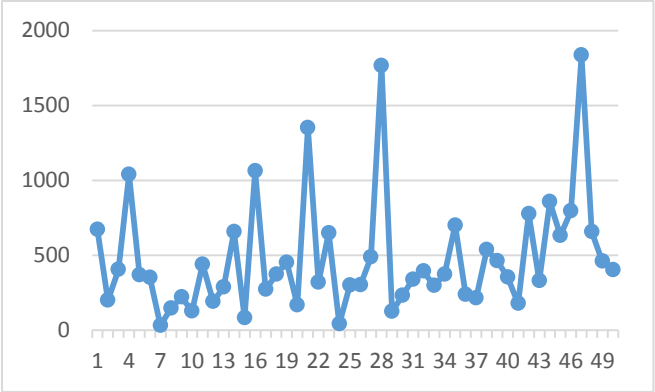

Figure 2: Distribution of events over retrieved dataset

This is due to some privacy concerns. Due to the large volume of the original dataset, we selected a sample of 134,069 tweets that cover the same number of events covered by the original dataset. This sample is provided by the dataset authors. We developed a small tool that uses Twitter API to fetch the content of each tweet. After iterating over the dataset, we were able to fetch only 59,732 tweets that cover 50 significant events, as most of the tweets were deleted or their authors' accounts were suspended, or deleted. Only 23,973 tweets are labeled and the remaining do not represent an event or do not belong to any of the covered events. Figure **2** shows the retrieved event labeled and the corresponding number of tweets.

B Measurements

We used the same measurements introduced in [19]. These measurements are described as follows:

- **Event Recall**: In general, recall is a measurement used in text retrieval. It is the fraction of relevant instances that have been retrieved over the total amount of relevant instances [20]. A detected event is a cluster of tweets. A cluster is said to be true positive if its containing tweets cover any of the reference events. As a cluster may contain hundreds or thousands of tweets, it is very likely to find tweets that do not belong to the event. Thus, the proportion of tweets in the cluster that are part of a single reference event is computed. If the proportion is greater than a threshold then it is true positive. We used the threshold value of 0.5 as recommended by [19]. Thus recall in event detection is calculated as follows:

$$recall = \frac{number\ of\ covered\ events}{number\ of\ reference\ events}$$

- **Event Precision**: Precision is the fraction of relevant instances among the retrieved instances [20]. Precision of event detection is calculated as

follows:

$$precision = \frac{number\ of\ covered\ events}{number\ of\ detected\ events}$$

- **F-Measure**: It is the harmonic average of the precision and recall. It is used to determine the accuracy of classification problems [20].

$$f - measure = \frac{2 * precision * recall}{precision + recall}$$

C Experiments Setup

We conducted our experiments over Evetar to compare our results with other approaches of event detection. We developed a system that implements our approach. In reality, Twitter data comes as a stream of tweets in chronological order, thus it is not feasible to apply the event detection approach on the whole dataset. To mimic this behavior, we split the tweets into chronologically ordered chunks/subsets. The number of subsets depends on the specified time interval. We choose two-time intervals, a one-day time interval, and a 6-hours time interval. Unlike other approaches, long time intervals are selected because of the small number of tweets retrieved from the dataset IDs.

D Results

After running our system on the resulted subsets of the 6-Hours interval we achieved the results shown in Figure **3**. We can observe that at window 15 the lowest f-measure is achieved, this is due to the bad distribution of events in that window where most events have lower number of tweets. Consequently, frequencies of event triggers belong to these events are low too. Thus, using the average as a threshold value eliminates these event triggers. This can be a limitation of our threshold value selection, but our objective in this work is to detect significant events which are the ones with the highest frequencies. On the other hand, subset 19 produced the highest f-measure. The calculated average of recall, precision and f-measure is 0.607, 0.684, and 0.644.

The one-day time interval achieved the results shown in Figure **4**. In subset 1 we achieved the highest f-measure with a value of 0.888. We were able to detect events that have a lower number of tweets because events with high number of tweets produced different event triggers, thus the frequency was distributed among these event triggers, which made the average value had better effect in the results. Table 5 shows a sample of tweets of the same event that produce different event triggers. The lowest f-measure value is 0.533 achieved in subset 11. The calculated average of recall, precision and f-measure is 0.654, 0.793, and 0.717 respectively. Table 6 shows the average measurements compared with three implemented approaches on Evetar [10, 21, 22].

Our approach outperformed both EDCoW and Peaky Topics, however, compared to MABED, we achieved better precision, but there was a wide difference in recall in favor of MABED. This is because the fetched dataset was incomplete and event-related tweets are lost during the fetch process. Thus, our approach failed to detect events with lower frequencies. This is acceptable as our objective is to detect significant events. Overall, our experiments show that having wider time intervals produce better results. Our findings coincide with both [9, 23]. Since wide intervals can cover enough tweets about an event, all event triggers related to this event will have higher frequent pattern.

V CONCLUSION

In this paper, we have presented a Soft Frequent Pattern Mining Based approach that uses event triggers for detecting significant Arabic events from text founded in microblogs. Our approach is based solely on the textual features of the text without relying on any platform-specific features such as hashtags, mentions, retweets. It depends only on the event triggers extracted from the text. Experimental results indicate that the approach outperformed some widely known approaches and comparable with others. It also detects real events with proper accuracy instead of general viral topics. In the future, we will enhance the process of extracting the frequent event triggers using dynamic threshold value. In addition, we will adapt our approach to make it applicable for real-time uses.

REFERENCES

[1] Alkhamees, N. and M. Fasli. *Event detection from social network streams using frequent pattern mining with dynamic support values*. in *Big Data (Big Data), 2016 IEEE International Conference on*. 2016. IEEE.

[2] Allan, J., *Topic detection and tracking: event-based information organization*. Vol. 12. 2012: Springer Science & Business Media.

[3] Dou, W., et al. *Event detection in social media data*. in *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*. 2012.

[4] Allan, J., *Introduction to topic detection and tracking*, in *Topic detection and tracking*. 2002, Springer. p. 1-16.

[5] Hurlock, J. and M.L. Wilson. *Searching Twitter: Separating the Tweet from the Chaff*. in *Icwsm*. 2011.

[6] Atefeh, F. and W. Khreich, *A survey of techniques for event detection in twitter.* Computational Intelligence, 2015. **31**(1): p. 132-164.

[7] Alsaedi, N. and P. Burnap. *Arabic event detection in social media*. in *International Conference on Intelligent Text Processing and Computational Linguistics*. 2015. Springer.

[8] Farghaly, A. and K. Shaalan, *Arabic natural language processing: Challenges and solutions.* ACM Transactions on Asian Language Information Processing (TALIP), 2009. **8**(4): p. 14.

[9] Doulamis, N.D., et al., *Event detection in twitter microblogging.* IEEE transactions on cybernetics, 2016. **46**(12): p. 2810-2824.

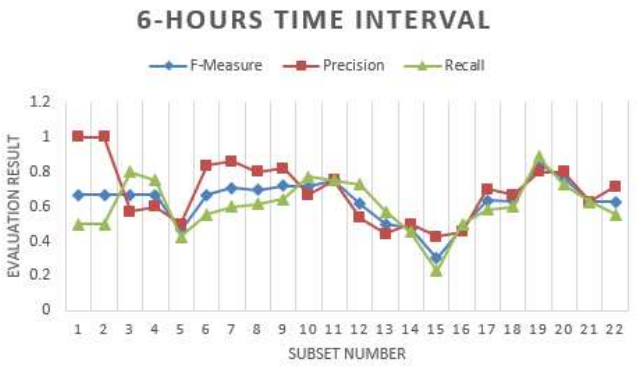

Figure 3: The results produced by dividing the dataset with 6-hours time interval

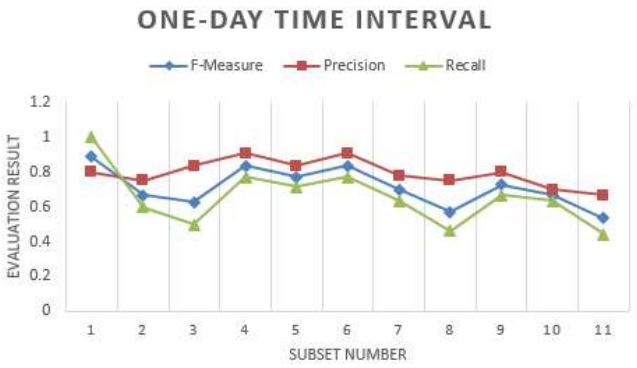

Figure 4: The results produced by dividing the dataset with one-day time interval

[10] Weng, J. and B.-S. Lee, *Event detection in twitter.* ICWSM, 2011. **11**: p. 401-408.

[11] Wang, X., L. Tokarchuk, and S. Poslad. *Identifying relevant event content for real-time event detection*. in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. 2014. IEEE.

[12] Yılmaz, Y. and A.O. Hero, *Multimodal Event Detection in Twitter Hashtag Networks.* Journal of Signal Processing Systems, 2016. **90**(2): p. 185-200.

[13] Mohammad, A.-S. and O. Qawasmeh, *Knowledge-based Approach for Event Extraction from Arabic Tweets.* International Journal of Advanced Computer Science & Applications, 2016. **1**: p. 483-490.

[14] Shao, M., et al. *An efficient approach to event detection and forecasting in dynamic multivariate social media networks*. in *Proceedings of the 26th International Conference on World Wide Web*. 2017. International World Wide Web Conferences Steering Committee.

[15] Petkos, G., et al. *A soft frequent pattern mining approach for textual topic detection*. in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*. 2014. ACM.

[16] Katragadda, S., R. Benton, and V. Raghavan, *Framework for real-time event detection using multiple social media sources.* 2017.

[17] Aggarwal, C.C. and J. Han, *Frequent pattern mining*. 2014: Springer.

[18] Almerekhi, H., M. Hasanain, and T. Elsayed. *Evetar: A new test collection for event detection in arabic tweets*. in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2016. ACM.

[19] Petrovic, S., *Real-time event detection in massive streams.* 2013.

[20] Sammut, C. and G.I. Webb, *Encyclopedia of machine learning*. 2011: Springer Science & Business Media.

[21] Guille, A. and C. Favre. *Mention-anomaly-based event detection and tracking in twitter*. in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. 2014. IEEE.

[22] Shamma, D.A., L. Kennedy, and E.F. Churchill. *Peaks and persistence: modeling the shape of microblog conversations*. in *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. 2011. ACM.

[23] Gaglio, S., G.L. Re, and M. Morana. *Real-time detection of Twitter social events from the user's perspective*. in *Communications (ICC), 2015 IEEE International Conference on*. 2015. IEEE.

Table 5
Tweets of belong to the same event produces different event triggers

Tweet	بي بي سي: غضب إسرائيلي وأمريكي عارم على توقيع عباس طلب الانضمام للمحكمة الجنائية الدولية: توقيع رئيس السلطة ال
Event Trigger	توقيع عباس، توقيع رئيس

Tweet	نتنياهو في رده على توقيع الرئيس عباس على معاهدة روما يقول ان السلطة هي من يجب ان تقلق من المحكمة الجنائية الدولية لتحالفها مع حركة حماس
Event Trigger	رد توقيع، توقيع عباس

Tweet	عباس يوقع اليوم طلب الانضمام الى المحكمة الجنائية الدولية بعد رفض مشروع القرار الفلسطيين
Event Trigger	يوقع طلب، رفض مشروع

Table 6
Summary of the results achieved by our approach compared to other approaches applied to Evetar.

Approach	Recal	Precission	F-Measure
EDCoW	0.15	0.09	0.11
Peaky Topics	0.80	0.11	0.19
MABED	0.92	0.61	0.73
Our Approach (6-Hours Interval)	0.607	0.684	0.644
Our Approach (1-Day Interval)	0.654	0.793	0.717