IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

Segmentation Based Event Detection

Zeenath MT[1], Balu John[2]

PG Student, Department of Computer Science and Engineering,

Government Engineering College, Palakkad, Kerala, India[1]

Associate Professor, Department of Computer Science and Engineering,

Government Engineering College, Palakkad, Kerala, India[2]

Abstract: Twitter is a social networking and microblogging service on which users interact with messages. It is one of the best examples of microblogging and has a 280-character limit for a tweet. Registered users on twitter can post, like, and retweet tweets, but unregistered users can only read them. It is used not only to communicate with friends but also to share real-world events. Event detection is a major research area in text mining. Social media data (specifically, twitter data) is easily available. Twitter is a major source of information about real-world events. In twitter hashtags and word limit ensures the concise representation of real-world events. In this work, a segmentation based model is used to detect real-world events. Hashtags are the most important segment in the event detection process. The method of event detection is to split each tweet and hashtags into segments. From these segments extract the bursty segments. Then bursty segments are clustered based on the similarity measures. Finally, these clusters are summarized to produce final event. The key features of the event detection system are hashtags, retweet count, user popularity, and follower count. Here hashtags are more important and giving more weight to improve the performance of the model. The event detection system uses a Wikipedia title file for indexing the segments. Events2012 dataset is used for event detection. The results show the events are real-world in most of the cases.

Keywords: Microblogging, Hashtags, Natural language processing, Segmentation.

I. INTRODUCTION

Detecting real events from tweets is a very difficult task. About 500 million short messages are tweeted every day. So the task of event detection, in order to generate real-world events from the tweets, is more complicated. Tweets are mainly classified as "pointless babbles", "conversational", "self promotion", "pass-along value", "news" and "spam". In this 40\% are Pointless babbles that are personal insight that may be interesting to family and friends. 37\% are Conversational that include Questions, polls, back and forth dialog in an almost instant message fashion. 9\% are Pass along values that are Re-tweets passed along from other Twitter members. 6\% are Self-promotion that are Tweets about members products, services, shows, or companies. 4\% are Spam. 4\% are News from mainstream media sources like CNN. These pointless babbles, spam, and self-promotions are discarded for event detection. The tweets may contain noisy data, informal writing, grammatical errors, and a large volume of data coming at very high velocity. These are insignificant in the task of event detection.

Currently, many efforts have been taken for different event detection approaches. Event detection is not a new topic of research, but rather an area on which extensive research has been done over this decade. Event detection methods are classified into statistical, probabilistic, artificial intelligence and machine learning, and composite. Most event detection methods fit by one of these methods, but several methods combine two or more. statistical event detection is the simplest and most computationally straightforward method. Probabilistic event detection methods based on the probability of event occurrence and other related probabilities. In the composite method of event detection, it uses Bayesian Gaussian Process (BGP) models. It is an instance, combines probabilistic and machine learning methods.

The basic idea of event detection as shown in Figure 1. The input of the system is tweet streams, and output is the real-world events. The Twitter API is used to collect raw tweets from the Twitter platform. So the data are collected using Twitter Application Program Interface(API). Raw tweet module receives the filtered tweet data in the form of raw tweets from the Twitter platform, which comes directly from the Twitter API. The pre-processing module does the filtering of spam tweets through the classification procedures. This module also performs the removal of special characters and separation marks and followed by standardization of data, which involves upper- and lower-case conversion. The Twitter analysis module extracts for processing and analyzing the incoming pre-processed tweets, which reflects the required type of event. In the final module, bursty keywords are extracted by discarding the non-bursty keywords. Event detection has drawn important attention of researchers to automatically extract, understand, and summarize the happenings of an event in different fields. In event detection, any input message by a user posted to a social network or on twitter can be considered as his observation on a real-world happening at a certain location and time. This type of observation is called

DOI 10.17148/IJARCCE.2020.9656

an event element. Twitter is important for helping to spread important news. Emerging events are streamed on social network and it is real-time in nature. These emerged events are typically driven by breaking news and general topics that attract the attention of a large fraction of social media users. Thus, event detection has high importance and significance to news reporters and analysts. For example, the announcement of Sushant Singh Rajput's suicide on June 14, 2020, twitter was immediately flooded with an enormous volume of connected discussions and comments. Event detection is also important for online marketing professionals and opinion tracking companies, as emerging events capture the general public attention. Event analysis requires real-time event detection for a live stream of data from the social media platform where topics of discussion shift dynamically with time. One of the challenges in the event detection technique is Domain Dependence, which is suitable for one domain might not be the same for the other domains and it is extremely situational-dependent. The next challenge is the time constraints. An extreme timeline constraint is a timeline in which the event detection method should be able to identify events correctly. The other challenge is voluminous data, which is a huge volume of data that requires a high-powered computing algorithm and immense storage space to store, access, filter, and process all data.
This paper is organized as follows: Section II describes the details of the related works done in the field of event detection. Section III explained about methodology. Section IV contains the experimental setup and outputs obtained from the framework. Section V gives brief concluding comments

II. LITERATURE SURVEY

The chapter discusses different research works done in the area of event detection. A comparison of the proposed system with the previous works is also done. The problem definition and its brief explanation are also included. As mentioned earlier, the four major event detection methods are statistical, probabilistic, artificial intelligence and machine learning, and composite. Artificial intelligence and machine learning methods become popular with the emergence of different strongest neural network and machine learning models. Composite methods become popular with a combination of probabilistic and machine learning methods. Numerous works have been done for implementing an event detection system are based on normal machine learning techniques. The works considered for comparison in this survey are based on artificial and machine learning tasks.

The event detection task proposed by Amosse Edouard [1] is Graph-based Event Extraction from Twitter. It is an unsupervised approach to detect open domain events on Twitter, where the stream of tweets is represented through temporal event graphs, modeling the relations between Named Entities (NE) and the terms that surround their mentions in the tweets. The event extraction is carried out by six steps. The steps are, Tweet Preprocessing, Named Entity Recognition and Linking, Graph Generation, Graph Partitioning, Event Detection, and Event Merging. NE mentions in the tweets are extracted using the Twitter-specific Named Entity Recognizer (NER) tool. The graph is modeled by the terms surrounding the mention of a NE in a tweet. So, this NE in a tweet defines its context, thus rely on the NE context to create event graphs. An event graph is generated to represent the relationships between terms in the NE contexts.
Then apply graph theory to partition the graph into sub-graphs, which will be considered as event candidates. Tweets related to the same events usually share a few common keywords. In graph partitioning, a good partition of a graph can be obtained by separating high ranked vertices from low-ranked ones, if the nodes in the graph have distinguishable values. When a new event is found as duplicate, then merge it with the previously detected event.

Alberto Tonon [2] proposed a system to detect events such as terrorist activity and natural disasters by analyzing Twitter posts. This system is developed for the Swiss Armed Forces. The system extracts a structured representation from the tweet's text using Natural Language Processing (NLP) technology, which it then integrates with DBpedia and WordNet in an RDF knowledge graph. The objective of armasuisse was to detect events with complex descriptions, so depart from statistical and IR approaches and use semantic search instead.

It uses natural language processing to associate each tweet with a set of quads of the form (subject, predicate, object, location), describing who did what to whom and where; any of these components can be empty, which is denoted as x. Then associate with each tweet a set of entities whose role (subject or object) in the tweet could not be determined. Subjects, objects, locations, and entities are matched to DBpedia [3], a knowledge base extracted from Wikipedia, and predicates are matched to verb synsets in WordNet [b4], an extensive lexicon. Thus, DBpedia and WordNet provide us with a vocabulary and background knowledge for describing complex events.

In this literature, the procedure is carried by using three components, that is NLP, Semantic Analysis, and Event Detection. The Natural Language Processing component is used to analyses the tweets’ text. The NLP component and extracts the quads and entities from these tweet text. The quads and entities are independent of the complex event descriptions. The Semantic Analysis component converts the quads and entities into RDF, which is then analyzed and filtered using the user’s event descriptions. The output of RDF is a set of time series, each time series consisting of a

DOI 10.17148/IJARCCE.2020.9656

summary and a set of tweets. Finally, the Event Detection component uses an outlier detection algorithm to extract from each time series that correspond to the actual events. The resulting events and their summaries are finally reported to the user.

AJ McMinn[5] proposed the use of NEs for the efficient and effective detection and tracking of events on Twitter. Then conjecture that named entities are the building blocks of events; the people, places, and organizations involved are crucial in describing an event. This real-time approach[5] identifies bursty named entities and uses an efficient clustering approach to detect and break events into individual topics, each of which describes a different aspect of an event.NEs play a key role in describing events, such as the people involved, or the location where the event took place. Without this information or some other contextual clue, it is unreasonable to expect a person or machine to determine the specifics of an event.

III. SEGMENTATION BASED EVENT DETECTION SYSTEM

This section gives the system architecture and the overall implementation details of the proposed system. How well the proposed segmentation-based event detection system works on the Event2012 dataset is also explained. The proposedsystem detects real-world events on the Event2012 dataset. The detected events include information about Sports, Politics, Entertainment, Science Technology, etc. Here, develop a segmentation-based event detection system from tweets to detect real-world events.
Figure 1 shows the basic architecture of the proposed system. The overall system architecture can be considered as four main modules namely:

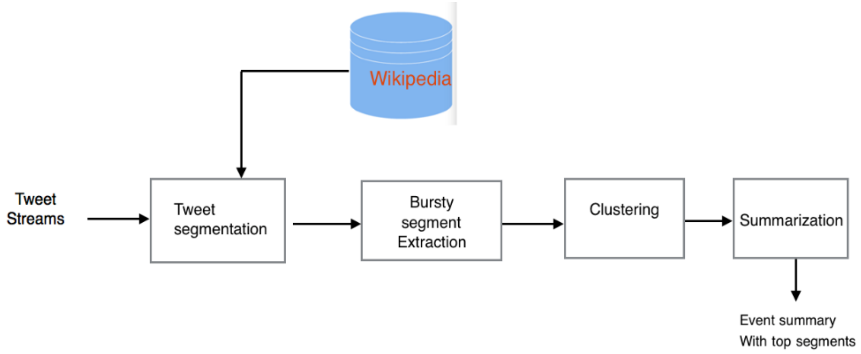

Fig. 1 System Architecture

- Tweet segmentation module
- Bursty Segment Extraction module
- Bursty Segment Clustering module
- Event Summarization module

A. Tweet Segmentation Module
In the tweet segmentation module, the tweets are segmented using a dataset called Wikipedia page titles dataset. This dataset contains all the title of every article on Wikipedia. This module performs splitting a given tweet into non-overlapping meaningful segments. The generated segment can be a unigram or a multigram. This segment (or unigram or multigram) contains much more specific information. For example, the segment [vice presidential debate] is much more informative then [vice], [presidential], and [debate] separately. While segmenting the tweet, give some important to following 3 components:

- Tweet text
- Name mention
- Hash tags

Tweet text is the content of the tweet. The tweet text is considered as a user write in a tweet except for URL links, name mentions, and hashtags. From these segments only keep those segments are present on the Wikipedia page. So, this ensures only named entities or meaningful segments are kept. Remaining segments are removed otherwise increase the noise in the event detection process. One of the examples for tweet text is Photography, cryptocurrency, etc. The name mentions in a tweet is to mention a person by their user name. For example, @msdhoni user mention is used for Mahindra Singh Dhoni. So these name mentions are replaced by their actual name and this name is considered as a segment.
The most important component in the event detection system is hashtags. Hashtags contain more information in a condensed manner. The related tweets carry the same hashtag. For example, the hashtag #InternationalWomensDay will

DOI 10.17148/IJARCCE.2020.9656

be segmented as [international women's day]. Hashtags do not contain any capitalization of letters. It would be considered as a unigram when segmenting them. Hashtags do not contain whitespace or punctuations.

B. Bursty Segment Extraction Module

Since there are hundreds of thousands of unique tweets are streamed per day. So processing millions of segments is computationally expensive task. Then clustering all of them for detecting events also be a computationally expensive task. So, once the tweets are segmented, find out the bursty segments that are related to an event and discard the remaining segments. Let Nt denote the number of tweets within the time window t and fs,t be the number of tweets containing segment s in t. The probability of observing s with a frequency fs,t can be considered as a Binomial distribution B(Nt, ps) where ps is the expected probability of observing segment s in any time window. Since the number of tweets within the time window is very large in the case of tweets. If a segment has fs,t is greater than or equal to Expectation value, it will be called a bursty segment, while a segment with fs,t is less than the value of Expectation will not be considered bursty and will be discarded. The formula for the bursty probabilityPb(s,t) for segment s in time window t:

$Pb(s, t) = S((fs,t) - (E[s|t] + \sigma[s|t])/\sigma[s|t])$

retweet is a copy of a tweet created by another user, that is when someone republishes or forwards a post to their own Twitter followers Then find the tweet retweeted by many users might be related to an important event and can be used to provide more weight to segments in retweets. So define retweet count of a segment s in t as srcs,twhich is the sum of retweet counts of all tweets. A tweet by someone who has millions of followers might also be more important as compared with someone who has very few followers. Giving more weight to such tweets will ensure that spam or self-promoting tweets are filtered out. Segment follower count of a segment s in t as sfcs,t which is the sum of follower count of all users using this segment in t. Combining these formulas, the burst weight wb(s, t) for segment s in t is as follows:

$wb(s, t) = Pb(s, t)\log(us,t) * \log(srcs,t)\log(sf\ cs,t)$

The score of segments are calculated using bursty probability (pb) and follower count (fc) retweet count (rc) and count of unique users using them (u):

$Scores = pb(s) * \log(us) * \log(rcs) * \log(f\ cs)$

C. Bursty Segment Clustering module

This module used to cluster the bursty segments and removing non-bursty segments or non-event cluster. In twitter, topics are fast-changing and dynamic. So the time window is split into M sub-windows t = {t1, t2, ..., tM }. The tweet frequency of segment s is ft(s,m) in the sub-windowtm and Tt(s,m) be the combination of all the tweets in the sub-window tm. In clustering, the tf-idf similarity of the set of tweets T1and T2 is calculated. The K-Mean clustering algorithm is used to cluster the favor segments. This clustering algorithm is applied to the bursty segments. The clustering is done by all bursty segments and is considered as nodes and initially, all nodes are disconnected. Then an edge is added between segments sa and sb only if k-Nearest neighbors of sa contains sb and vice versa.After adding all possible edges from segment to segments, all the connected components of the graph are considered as candidate event clusters. Those segments that do not have any edges are removed from further processing. In the event cluster some segments like [sunday dinner], [sundaynight], [every sunday], [sunday funday], and [next sunday], these segments are bursty on specific days of the week. Sothis type of events are filtered using external knowledge base like Wikipedia page database.

D. Event Summarization

Here a list of segments that are associated with an event cluster might not provide all the information regarding an event. So the task of summarization is very important in this scenario. For summarization, the TextRank algorithm is used in the event clusters obtained in the previous step. It is an unsupervised graph-based approach for text summarization. The algorithm takes input as multiple documents and provides a summary of it. This summarizer uses a graph-based method that computes the pairwise similarity between two segments and makes the similarity score between two segments. The final score of a segment is computed based on the weights of the edges that are connected to it. In this model, we have a connectivity matrix based on intra segment cosine similarity which is used as the adjacency matrix of the graph representation of segments. Then the segments are ranked according to their similarities.

VI. RESULTS AND DISCUSSION

The experimental setup, evaluation, and results obtained for the proposed system are discussed in detail in this section. The major aspects used in the implementation of the proposed system are explained here.

A. Dataset

As mentioned before, this proposed system aims to detect real-world events from a given user tweet. In the tweet segmentation phase Wikipedia page titles dataset is used. This dataset contains 8,007,358-page titles and 4,342,732 distinct entities that appeared as anchor text. McMinn [6] created a Twitter corpus called Events2012 containing tweets from Oct 10 - Nov 7, 2012. After all, filtering is done, the corpus contains over 120 million tweets. The Events2012

DOI 10.17148/IJARCCE.2020.9656

corpus also contains a list of 506 events detected in the corpus distributed among 8 categories. This corpus is used to estimate the segment probabilities ps in the bursty segment extraction phase and also used to evaluate the performance of the proposed model. Both Wikipedia Page Titles and the tweets in the Event2012 corpus were preprocessed. Consider an example from [7],
Tweet text: Amanda Todd took her own life due to cyber bullying RipAmandaTodd NoMoreBullying
Segmentation: [amanda todd], [cyber bullying], [rip amanda todd], [no more bullying].

B. Results

Most of the previous work defines another measure called Duplicate Event Rate (DERate) as a percentage of events that have been identically detected among all realistic events. Precision is the fraction of the detected events that are related to a realistic event. We use these definitions of precision and DERate in our evaluation. Here, we did not use recall as a measure to evaluate the results found by the model because of the lack of an exhaustive list of events in the Events2012 dataset. Although Events2012 dataset has provided a list of 506 events detected by their [6] model within the period of Oct 10 - Nov 7, 2012. The proposed model finds 48 events within a period of Oct 11 - Oct 17, 2012, that were not reported by them [6]. Instead of recall, use No. of events, which is the number of realistic events detected, as a measure to evaluate the performance of the model. Several parameters that affect the performance of the proposed method like no.of sub window M, time window size, hashtag weight H, number of neighbors k, and threshold value T. The time window is set to be of 24 hours which contains M = 12 sub-windows of 2 hours each. Then the hashtag weight is set H = 3, k = 3 neighbors, and threshold T = 4 in the event detection system. In the tweet segmentation section, the value of hashtag weight H is denoted by how many times the frequency of a hashtag is multiplied. A lower value of H cause noisy segments from tweet text and affect the accuracy of the event detection model. The higher value of H would not allow other frequently used segments in the tweet text to become bursty and again reduce the accuracy. The value of hashtag weight H = 3 produces the best results. In the event clustering section, the threshold T was used for deciding if a candidate event cluster is a realistic event or not. The observation on the value of T, the higher value get event would be considered as realistic. On experimenting with different values of T from 2,3,4, and 5, the optimal results at T = 4.

IV. CONCLUSION

In this era, social media plays an important role in people's life. One of the social media, Twitter has accomplished a serious increase in both users and the volume of information. Tweets being short and containing noisy data in large volume produce challenges on event detection task. The key features of segmentation-based event detection systems are hashtags, retweet count, user popularity, and follower count. The hashtag weight takes apart an important role in a segmentation-based event detection system. So, giving more weight to hashtags obviously improve the performance of the system. The bursty segment extraction and bursty segment clustering techniques are used to extract clusters related to ongoing real-world events. The proposed system undergoes mainly based on the four steps. The first step is tweet segmentation, segment all the tweet text into uni gram or multi gram. Hashtag weight is more important in this step. Bursty segment extraction is based on the segment probability. Then the bursty segments are clustered using the clustering algorithm and here threshold value is more important. The final step is to summarize these event clusters to detect the event.

REFERENCES

[1] A. Edouard, E. Cabrio, S. Tonelli, and N. Le Thanh, “Graph-based event ex-traction from twitter,” 2017

[2] A. Tonon, P. Cudr e-Mauroux, A. Blarer, V. Lenders, and B. Motik, “Armatweet: detecting events by semantic tweet analysis,” in European Semantic Web Conference, pp. 138–153, Springer, 2017.

[3] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes,S. Hellmann, M. Morsey, P. Van Kleef, S. Auer,et al., “Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia,”Semantic web, vol. 6,no. 2, pp. 167–195, 2015.

[4] G. A. Miller, “Wordnet: a lexical database for English,” Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.

[5] A. J. McMinn and J. M. Jose, “Real-time entity-based event detection for twitter,” in International conference of the cross-language evaluation forum for european languages, pp. 65–77, Springer, 2015.

[6] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, “Building a largescale corpusfor evaluating event detection on twitter,” in Proceedings of the 22nd ACM in-ternational conference on Information Knowledge Management, pp. 409–418,2013.

[7] Keval M. Morabia, Neti Lalita Bhanu Murthy, Aruna Malapati and Surender S. Samant” SEDTWik: Segmentation-based Event Detection from Tweets using Wikipedia”, Association for Computational Linguistics: June 3 - 5, 2019.

[8] C. Li, A. Sun, and A. Datta, “Twevent: segment-based event detection from tweets,” in Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 155–164, 2012.

[9] R. Cox, “Regular expression matching can be simple and fast (but is slow injava, perl, php, python, ruby,...),”URL: http://swtch. com/rsc/regexp/regexp1.html, 2007.

[10] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. Mc-Closky, “The stanford corenlp natural language processing toolkit,” in Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55–60, 2014.

[11] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using network x,” tech. rep., Los Alamos National Lab.(LANL), LosAlamos, NM (United States), 2008

DOI 10.17148/IJARCCE.2020.9656