Road Traffic Event Detection Using Twitter Data, Machine Learning, and Apache Spark

Ebtesam Alomari¹, Rashid Mehmood², and Iyad Katib¹
¹Faculty of Computing and Information Technology, ²High Performance Computing Center
King AbdulAziz University
Jeddah, Saudi Arabia
E: Alomari0011@stu.kau.edu.sa, {RMehmood, IAKatib}@kau.edu.sa

Abstract—Road transportation is the backbone of modern societies, yet it costs annually over a million deaths and trillions of dollars to the global economy. Social media such as Twitter have increasingly become an important source of information in many dimensions of smart societies. Automatic detection of road traffic events using Twitter data mining is one such area of a great many applications and enormous potential, albeit facing major challenges concerning the management and analysis of big data (volume, velocity, variety, and veracity). Various approaches on the subject have been proposed in recent years, but the methods and outcomes are in their infancy. This paper proposes a method for automatic detection of road traffic related events from tweets in the Arabic using machine learning and big data technologies. Firstly, we build and train a classifier using three machine learning algorithms, Naïve Bayes, Support Vector Machine, and logistic regression, to filter tweets into relevant and irrelevant. Subsequently, we train other classifiers to detect multiple types of events including accident, roadwork, road closure, road damage, traffic condition, fire, weather, and social events. The results from the analysis of one million tweets show that our method is able to detect road traffic events, as well as their location and time, automatically, without any prior knowledge of the events. To the best of our knowledge, this is the first work on traffic event detection from Arabic tweets using machine learning and the Apache Spark big data platform.

Keywords—Twitter data analysis, Smart transportation, Event detection, Smart cities, Machine learning, Text mining, Big data analytics, Apache Spark, MongoDB, Naïve Bayes, Support Vector Machine (SVM), Logistic Regression

I. INTRODUCTION

Road transportation is the backbone of modern cities and societies, yet it costs annually, 1.25 million deaths and 20-50 million people injured across the globe [1]. Moreover, road traffic congestion is one of the most significant problems in modern cities. The annual cost of congestion to the US economy alone exceeds $305 billion [2]. The increasing number of vehicles, social events, lane closures, roadworks, adverse weather, and other unexpected incidents have a negative impact on traffic flow and cause traffic congestions. Therefore, those causes, namely events (incidents), should be detected in an efficient and timely manner in order to support decision making and set management strategies to reduce or eliminate congestion.

Smart cities provide “state-of-the-art approaches for urbanization, having evolved from … knowledge-based economy … digital economy and intelligent economy. The notion of smart cities can be extended to smart societies … digitally enabled, knowledge-based societies, aware of and working towards social, environmental, and economic sustainability” [3]. In smart cities and societies, a large amount of diverse information is produced daily by heterogeneous sources including GPSs, cameras, smartphones as well as user-generated content from social media. Such data offers the potential for developing novel solutions that will support decision making for smart transportation. In recent years, several approaches related to transportation in smart cities have been proposed, e.g., autonomic transportation systems [4] and intelligent disaster management [5].

Social media such as Twitter and Facebook are a relatively inexpensive and conveniently available source of information comparing to physical sensors that cost greatly to install at a large scale to monitor the traffic flow. Twitter is one of the most popular microblogging media used for communication and sharing personal status, events, news, etc. Twitter allows users to post short text messages called tweets. A massive amount of real-time data is posted by millions of users on various topics including transportation and real-time road traffic.

Moreover, Twitter has been adopted as a powerful data source in smart transportation. In recent years, there has been an increasing amount of literature on the use of Twitter as a sensor for traffic monitoring [6], flow forecasting [7], congestion estimation [8], and event detection [9]. These approaches show great potential for this area, albeit face major challenges. From the data mining perspective, event detection from unstructured, rapidly evolving tweets is a challenging task. The Twitter data has all the characteristics of big data, i.e., volume, velocity, variety, and veracity. Therefore, the management and analysis of Twitter data for event detection purposes is a major challenge. Advanced techniques and efficient approaches for data mining are required to extract useful information, monitor the changes and predict future observations [10].

Another dimension of the automatic event detection domain is the language of the tweets. Many researchers have attempted using social media information to monitor road traffic in different countries by analyzing text from different languages such as Japanese [11], Italian [12], and Chinese [13]. Our interests lie in detecting events from tweets in Saudi Arabia which has its own challenges due to the dialectical Arabic which is used mostly in everyday tweeting compared to the formal Modern Standard Arabic (MSA).

Another research gap is the limited use of big data technologies in the automatic detection of road traffic events from Twitter data in the Arabic language. Particularly, to the best of our knowledge, no work exists that uses big data technologies for automatic event detection of road traffic events from tweets in the Arabic language.
To sum up, several approaches to automatic event detection from Twitter data have been proposed in recent years, but the methods and outcomes are in their infancy. This work aims at detecting traffic-related events to enable smarter transportation. To this end, we propose a method for automatic detection of road traffic related events from tweets in Saudi dialect using machine learning and big data technologies. Firstly, we build and train a classifier using three machine learning (ML) algorithms, Naïve Bayes, Support Vector Machine (SVM), and logistic regression, to filter tweets into relevant and irrelevant. Subsequently, other classifiers were trained to detect multiple types of events including accident, roadwork, road closure, road damage, traffic condition, fire, weather, and social events. The results show that our method is able to detect road traffic events, as well as their location and time, automatically, without any prior knowledge of the events. Subsequently, the detected events are validated by searching in official sources such as newspapers websites. The classification accuracy is also evaluated using four widely used metrics; precision, accuracy, recall, and F-score.

The big data platform that we have used is Apache Spark. It is a powerful in-memory distributed computing platform that enables batch and streaming processing with extensive support for many machine learning algorithms for text mining and other applications [14].

The paper is organized as follows. Section 2 reviews the related work. Section 3 describes the methodology. Section 4 provides results. Section 5 concludes and gives future directions.

II. LITERATURE REVIEW

In this section, we review some notable literature related to social media based event detection. First, we review the works on traffic event detection in languages other than Arabic. Subsequently, we discuss the existing works about detection of various events (not necessarily traffic events) from Arabic social data. These two sections do not include any works that use big data. Finally, we review the works on traffic event detection that use big data technologies.

A. Traffic event detection using social media

In recent years, researchers had proposed many different approaches in online event detection from social media. Agarwal et al. [9] focused on identifying complaints reported (tweets) in road irregularities and bad road conditions. After extracting the important information; such as the problem and the location, they applied rule-based classifier and categorized them into useful, nearly-useful and irrelevant complaint reports. Sakaki et al. [11] mainly focus on detecting heavy-traffic information and weather information. They classified Japanese tweets into positive (event-related) and negative (not related to events) classes using Support Vector Machine (SVM) classifier. Furthermore, Klaithin and Haruechaiyasak [15] extracted information related to traffic using lexicon-based and rule-based techniques. These applied machine learning classifier based on Naïve Bayes Model to classify Thai tweets about traffic into six categories include accident, announcement, question, orientation, request, and sentiment. The trained the model using 4,637 tweets. Kumar et al. [16] detected road hazard by applying a trained language model to classify the tweets as having negative or non-negative sentiment. All tweets that expressed negative sentiment are considered as a tweet with road hazard information.

Additionally, they used three machine learning methods: Naïve Bayes, K-nearest-neighbor and the Dynamic Language Model (DLM). Moreover, D’Andrea et al. [12] collected real-time Italian tweets and classified them after applying text mining techniques. The tweets are classified into three classes namely, traffic due to an external event, traffic congestion or crash, and non-traffic. Other works on traffic event detection using social media include [17], [18], [19], [20], [21] and [22].

B. General event detection from Arabic Tweets

The amount of research about analyzing Arabic social information for event detection is considerably limited compared to what is done in other languages. AL-Smadi and Qwassmeh [23] used an unsupervised rule-based technique to extract events about technology, sports, and politics out of Arabic tweets. Furthermore, Alsaedi and Pete [24] proposed a framework for detecting disruptive events from Arabic tweets. The tweets are classified into event and non-events using a Naïve Bayes model. Also, they applied an online clustering algorithm to identify the topic of an event. Moreover, they extended their work and used the clustering algorithm to detect the riots events [25].

Other researchers [26] trained classification algorithms by using the training matrix that contains the selected terms and their corresponding TF-IDF (Term Frequency-Inverse Document Frequency) weights. They tested several algorithms. The results show that SVM was promising in terms of accuracy. However, the model was trained on a small dataset about 3700 Arabic tweets to detect one type of events which is a high-risk flood. Moreover, none of the above-discussed approaches for event detection from the Arabic text used big data platforms. Furthermore, their main focus was not on traffic events such as traffic jam.

C. Traffic Events detection using big data technologies

Alomari and Mehmood [27] used SAP HANA, which is an in-memory processing platform to analyze Arabic tweets related to traffic congestion in Jeddah city. In addition, they extracted the top causes of traffic congestion. Furthermore, they extended the work and proposed sentiment analysis approach for traffic events [28]. However, their approach was dictionary-based. They did not use machine learning techniques. Salas et al. [29] propose a framework for the real-time detection of traffic events from tweets in English language using Apache Spark and Python machine learning algorithms. Additionally, they used the SVM classification algorithm and classified the tweets into traffic and non-traffic related tweets.

Suma et al. [30] have analyzed tweets to detect events related to road traffic. They built a classification model to classify the tweets into traffic-related and non-traffic-related by using logistic regression with stochastic gradient descent. To detect events, they identify the most frequent terms among the traffic-related tweets. They improve the methodological and event detection aspects of their work in [31].

All of these approaches ([29], [30], [31]) used supervised classification algorithms on Apache Spark platform. However, they analyzed tweets in the English language. Lau [13] used the Latent Dirichlet Allocation (LDA) topic modeling module for unsupervised topic mining. After labeling the messages using a list of common traffic event keywords, they implemented ML classifier using Spark Machine Learning (MLib) library.

1889
However, they focused on three traffic events: traffic jams, poor road conditions, and traffic restrictions and analyzed the Chinese language data. Therefore, there is a need for an efficient and scalable approach mainly designed for the Arabic Language to address the challenges arising from Arabic big social data.

III. METHODOLOGY

Fig. 1 illustrates the proposed architecture for automatic traffic event detection from Arabic tweets using supervised ML algorithms and Apache Spark. It comprises six main components: (1) Data collection and storage component, (2) Data pre-processing component, (3) Feature extractor component, (4) Tweet filtering component, (5) Event detection component, and (6) Validation and results visualization component.

First, the data are collected using Twitter API, and the fetched JSON objects are stored in MongoDB. After removing the duplicates, we split the tweets into a labeled and unlabeled dataset. The authors manually tag each tweet in the labeled set with an appropriate label (1 for relevant, 0 for irrelevant). Second, we apply pre-processing steps to remove noise and prepare the data for classification. The output of this component is a list of normalized and cleaned tokens. Third, we extract the features and use TF-IDF as a feature vectorization method to reflect the importance of a term to a document (tweet) in the whole collection (tweets list). Fourth, the labeled tweets are used to build and train a classifier to filter tweets into relevant and irrelevant. Three models are built using three different supervised classification algorithms. Then, the four widely used evaluation metrics; precision, accuracy, recall, and F-score are used to evaluate the models and select the best algorithm. After that, we use the trained model that achieves higher performance than the others to filter out the irrelevant tweets. Fifth, part of the relevant tweets are manually labeled and used to build and train other classifiers to classify events. The trained classifiers are evaluated and then used for event detection. Finally, we visualize the results and validate the effectiveness of the classifier by searching in the official sources such as the newspaper website.

Moreover, we use Apache Spark platform, which is a distributed in-memory computing platform to handle the huge volume of unstructured data in twitter platform for event detection. Besides, we use Python Machine Learning (Spark ML) package, which provides high-level machine learning APIs built on top of Spark DataFrame. A DataFrame is a distributed collection of data organized into named columns. It is conceptually equivalent to a table in a relational database. DataFrames can be used with Spark SQL. Additionally, it can be constructed from different sources such as Hive tables, structured data files, external databases, or existing RDDs.

A. Data collection

Tweets are collected via Twitter REST API using geolocation filtering to obtain tweets posted in Saudi Arabia.
In addition, we collected tweets in hashtags that usually used to post about events in cities such as ‘#Riyadh_now’ (meaning ‘Riyadh_now’, ‘钽녕inals的语言’ = ‘‘#Riyadh_now’). We collected all Arabic tweets in the period between 23 September-1st October 2018.

Since our data required scalable and flexible schemas based storage, we selected NoSQL databases instead of the relational databases. The collected tweets are stored in MongoDB, which is a document-oriented database suitable for storing and managing Big Data-sized collections of documents like text. The fetched JSON objects from Twitter API are inserted into the database. Further, the Tweets object contains several attributes including (i) ‘created_at’, which represents the time when the tweet was posted and (ii) ‘full_text’ contains the message content. After that, we checked the redundancy and removed duplicate tweets (retweets). The total number of tweets after removing the duplicates is about 1 million.

B. Pre-processing

Pre-processing the text is an essential task since the Arabic morphology is rich and the Arabic dialectal text usually has typos or grammatical mistakes. Also, it is a critical step to reduce the amount of noise before classification because performing analysis directly on dialectal text may lead to poor results.

Algorithm 1 summarizes the main pre-processing steps. First, sparkConnector is used to connect to MongoDB. Then, the tweets are loaded and saved in Spark DataFrame. The next step is iterating over the tweets to remove all numbers, English alphabets and punctuations such as commas (,), period (.), semi-colons (;), colons (:), question marks (?), and so forth. Likewise, we strip the Arabic question mark (؟) and Arabic semi-colons (؛) so forth. Likewise, we strip the Arabic question mark (؟) and Arabic semi-colons (؛) and normalized to bare Alif (א). Also, the letter (א) pronounced Yaa is normalized to dotless Yaa (ג) and normalized to bare Alif (א). Taa marbutah to (ـ) and normalized to bare Alif (א).)

Furthermore, we check the result of the pre-processing phase before starting the classification. If the remaining number of tokens is equal to zero, the tweet is excluded from the analysis. Fig. 3 shows the steps applied to a sample tweet. The English translation for the tweet is: "#Riyadh_now abnormal congestion at the intersection of prince Fahad St. and University St.!!! Morning @Ruh_Rd". The removed punctuation, diacritics, and English words are highlighted in red color. The word ‘صلاة’ ends with Fatha Tanween (ـ), which is one of the Nunation diacritics. The normalized tokens are highlighted in green while the removed stop words are highlighted in red. All the discussed pre-processing implementation steps in this subsection are specific to the Arabic language except tokenization since it is based on splitting a string by white space regardless of the language.

C. Feature Extraction

We use Feature Extractors algorithms provided in Spark ML package. We apply TF-IDF (Term Frequency-Inverse Document Frequency), which is a measure of how important a word is to a document (tweet). The TF-IDF is merely the product of TF and IDF. The TF(t, d) is the frequency of the appearance of term t in document d while the IDF is a numerical measure of how much information a term provides. The IDF is calculated using the following equation:

$$IDF(t, D) = \log \frac{\|D\| + 1}{DF(t, D) + 1}$$

where \(|D|\) is the total number of documents in the collection D. Document Frequency DF(t, D) is the number of documents where the term t appears.

$$TFIDF(t, d, D) = TF(t, d) \cdot IDF(t, D)$$

To generate the term frequency (TF) vectors, we used CountVectorizer algorithm. The algorithm gets the list of tokens in ‘Tokens’ column as input and then converts them into vectors of token counts. Then, the resultant term frequency vectors are passed to the IDF algorithms. After that, the IDFModel will rescale the feature vectors, and the output will be stored in a new column named ‘Features’. This column is passed as input for classification algorithms.

D. Classification (Tweet Filtering)

Since not all the collected tweets are relevant to traffic, we filter the tweets before detecting events. So, we build a classifier to filter out the irrelevant tweets to traffic. We used machine learning algorithms in the Spark ML package.
To train the events classifier, the authors manually label parts of Naïve Bayes, SVM, and logistic regression algorithms. The models are trained on the training set. To find the best algorithm, we evaluate them over the testing set. The common statistical metrics, such as precision, accuracy, recall, and F-score are used to evaluate the trained classifier. To clarify the meaning of these metrics, we refer to traffic-related tweets as positive class and none related as negative class. The following four classes are used in these metrics: (i) True Positive (TP) for the positive tweets that correctly predicted as positive, (ii) True Negative (TN) for the negative tweets that correctly predicted as negative, (iii) False Positive (FP) refers to the tweets that labeled as negative but predicted as positive, and (iv) False Negative (FN) for the tweets that labeled as positive but predicted as negative. The corresponding equations for each metric are listed below. The accuracy is calculated by Eq. (3), Precision (Positive Predictive Value) by Eq. (4), Recall (True Positive Rate) by Eq. (5) and F-Score by Eq. (6).

\[
\text{acc} = \frac{TP+TN}{TP+FP+FN+TN} \tag{3}
\]

\[
\text{PPV} = \frac{TP}{TP+FP} \tag{4}
\]

\[
\text{TPR} = \frac{TP}{TP+FN} \tag{5}
\]

\[
F(\beta) = \left( 1 + \beta^2 \right) \frac{\text{PPV}.\text{TPR}}{\beta^2.\text{PPV}+\text{TPR}} \tag{6}
\]

E. Event Detection

For event detection, we build and train classifier using the Naïve Bayes, SVM, and logistic regression algorithms. To train the events classifier, the authors manually label part of the filtered data from the previous step into eight event categories, which are Fire, Weather, Social Events, Traffic Condition, Roadwork, Road Damage, Accident, and Road Closures. Traffic condition category includes negative and positive tweets about the traffic condition. For Fire events, all tweets about fires are included under this category even though it is not a vehicle fire because it may affect negatively on the traffic and cause congestion. Furthermore, for the social event, we focus only on the events that could affect the traffic (e.g., carnival, national day).

During our analysis, we notice that some event types have a large number of tweets compared to the other. So, we divided them into small-scale and large-scale events based on the number of tweets. The small-scale events are Traffic Condition, Roadwork, Road Damage, Accident, and Road Closures. The number of tweets for these events is small compared to Fire, Weather, and Social Events. So, we consider them as large-scale events.

Furthermore, we have a multi-label classification problem, since the classes (event types) are not mutually exclusive and the same tweet can belong to more than one class. For example, the tweet can be about Traffic Condition and Accident at the same time. To address this problem, we treat each label as a separate binary classification problem. Thus, we trained eight binary classifiers. For each event type, we consider the tweets about the event as positive while all the remaining tweets about the other types of events as negative. However, this will lead to imbalance sampling where the number of negative is larger than the positive. To adjust the class distribution and eliminate the effect on evaluation results, we perform undersampling for the negative (majority) class using the random undersampling method to make the data set balanced before evaluation. We prefer undersampling by removing samples from the majority class instead of oversampling by taking repeated samples from the minority class. Since the number of the negative labels is very large compared to the positive where it contains all the tweets about the other event types. Even though undersampling leads to loss of information, in our case, correctly classifying the negative labels is less important than the positive labels. Moreover, after detecting the events, we extract the time of occurrence using the time, and date information from 'created_at' attribute in the tweets object. Furthermore, we extract information about each event including location information using the top frequent terms since people usually refer to the event place using the hashtag. For model evaluation, we use the same evolution method explained in section 4.C. To validate the effectiveness of our event detection approach, we extract the top vocabularies from the tweets of each detected events. Then, we use these vocabularies to search in the official news/newspapers websites to confirm the occurrence of the events. After that, we compare the extracted information by our method including time and location with the real information in the official sources.

IV. RESULTS AND DISCUSSION

A. Results for Tweets filtering

The performance of the three classification algorithms (Naïve Bayes, SVM, and Logistic Regression algorithms) for tweet filtering is measured using the evaluation metrics explained in Eq. (3-6). Fig. 4 shows that SVM is better than Naïve Bayes and Logistic Regression algorithms in term of accuracy, F-score and precision. Furthermore, both SVM and Logistic Regression achieved recall of 90%.

![Evaluation results for tweets filtering](image_url)
Fig. 5. Evaluation results for events classification (a) Accuracy, (b) Precision, (c) Recall and (d) F-score

Fig. 6. Sample of the detected large-scale events per day (Year: 2018)

Fig. 7. Sample of the detected small-scale events per day (Year: 2018)

B. Results for event detection

Fig. 5 illustrates the evaluation results for the binary classification of events. The figure shows the four metrics: Accuracy, Precision, Recall, and F-score, respectively. We compared the result to select the algorithm that achieves higher results for the four metrics. We found that for Road Closures, Accident and Traffic Condition events, SVM worked better than the other algorithm. On the other side, the logistic regression algorithm achieved higher results for Social Event, Roadwork and Road Damage. Besides, SVM and logistic regression algorithms gave similar results for Fire and Weather events.

Moreover, we noticed that the results for Weather and Fire are higher than the other events. We assume that the reason is that our dataset contains only one big fire event (as explained later in this section), and thus we expected that most tweets about it contain similar vocabularies, which make the classification easier. Similarly, most of the tweets related to the weather condition are about rains.

Furthermore, we created charts to show the detected events per day. We divided them into two categories: large-scale and small-scale based on the number of tweets. Fig. 6 shows the large-scale events: Social Events, Weather, and Fire. On the other side, the small scale events including Traffic Condition, Accident, Road Damage, Roadwork and Road Closure are shown in Fig. 7. Moreover, we validated our event detection approach by searching in the official sources. From the tweets of each detected events, we extracted the top vocabularies. Then, we searched in the official news websites and local newspapers websites such as Okaz and Sabq. After that, we extracted the time information from the tweet object and drew charts to show the number of tweets in hours by day.

Fig. 8 shows the hourly number of tweets related to Social event. From the tweets about the Social Event on the 23rd of September, we listed the top vocabularies: ٍ الوطن (national), ٍيوم (day), ِاحتفالات (celebrations), ِسعودي (Saudi). The vocabularies illustrate that the detected event is the Saudi national day celebration where many activities were organized by municipalities in different cities.

The second large-scale detected event is about the weather condition. Fig. 9 shows the number of tweets in hours by day. The highest number of tweets about the weather was on the 27th of September. The top extracted vocabularies about this event are ٍالنور (now), ٍمطر (rain), ٍطويل (Taif), ٍمكة (Makkah). The news reports indicated that there were rains in Makkah region including Makkah and Taif cities on the same date.
Moreover, we detected Fire on the 1st of October. The list of the top vocabularies, which include ﺣﺮﻳﻖ (fire), ﺮﻳﺎﺽ (Riyadh), ﻷﻥ (north), ﻴﻠﮑﺮ (electricity), ﺳﺒﺎﻗ (video), ﺘﺤﻠﻴﻠ (station) indicates that the fire was on Riyadh city. The newspapers illustrated that there was a huge fire broke out at a power plant in Riyadh. As posted in newspaper websites, the Saudi Civil Defense received notification about the fire at 15:00. Fig. 10 illustrates that the number of tweets about Fire increased sharply at 15:00. This matches the event starting time clarified in the newspapers articles.

For small-scale events, we chose the top detected event which is about the traffic condition on the 25th of September. We listed the top vocabularies extracted from the tweets about this event, which include ﺟﺮم (Harammin), ﻷﻥ (Road), ﺑﻮ (Train), ﺱﺎر (Streets), ﺎر (Congestion), ﺷﺎ (Highway), ﺟﺪ (Jeddah). After that, we searched on the newspaper website using these vocabularies. We found that Al-Haramain high-speed railway was inaugurated on the same date. The inauguration ceremony was at the main station in Jeddah and started afternoon during rush hours when students/employee drive to home from work or school. Additionally, the closure of the main roads near the station increased the traffic jam. As shown in Fig. 11 and Fig. 12 the number of tweets on the 25th of September is increased after 12:00 pm. The discussed results above verify the ability of our proposed approach for automatic detection of large-scale and small-scale events, as well as the location and time without prior knowledge about the event.

V. CONCLUSION

In this paper, we focused on detecting road traffic related events to enable smarter transportation. We proposed a method for automatic detection of traffic events from tweets in Saudi dialect using machine learning algorithms and Apache Spark platform. Since the raw text is not suitable as direct input to classification, the text was divided into tokens and normalized after removing numbers, punctuation, diacritic, and non-Arabic words. TF-IDF was selected as weighting schemes and the tokens are converted into a vector of terms.

Furthermore, we trained a classifier to filter tweets into relevant (to traffic) and irrelevant. We used three machine learning algorithms, Naïve Bayes, SVM, and logistic regression. Subsequently, we trained the other classifiers to detect the occurrence of multiple traffic-related events in Saudi Arabia. We extracted information about each event including location information using the top frequent terms. Then, we searched in the official sources such as the newspaper websites to validate our approach. The results showed that our method is able to detect the traffic-related events, as well as their location and time, automatically, without any prior knowledge of the events.
In the future, we will improve the location detection approach to extract the exact location of the event especially if it is not mentioned in the text. Besides, we will develop a sentiment classifier to identify positive and negative tweets. For instance, traffic condition events will be classified into positive (no traffic jam) and negative (traffic jam). We will also improve the design, analysis and data variety aspects of our work. Finally, the proposed methodology can be applied to event types other than transportation, and other areas, because we collect all the tweets (without any filtering) and then build and train a classifier to filter out irrelevant tweets.

ACKNOWLEDGMENT

The work carried out in this paper is supported by the HPC Center at King Abdulaziz University.

REFERENCES