Real-Time Detection of Traffic From Twitter Stream Analysis

Amol jadhao1, Sweety Kumari2, Firdos Khan 3, Shekh Sultan 4, Ruchita Khandge5

1 Information Technology, University of Mumbai, Mumbai, Maharashtra, India
2 Computer engg, Savitribai Phule Pune University, Pune, Maharashtra, India
3 Computer engg, Savitribai Phule Pune University, Pune, Maharashtra, India
4 Computer engg, Savitribai Phule Pune University, Pune, Maharashtra, India
5 Computer engg, Savitribai Phule Pune University, Pune, Maharashtra, India

Abstract—Twitter has received much thoughtfulness recently. In this paper, we present a real-time monitoring system for traffic event detection from Twitter stream analysis. An important characteristic of Twitter is its real-time nature. The system fetches tweets from Twitter by using many search criteria; processes tweets, by using text mining techniques and then performs the classification of tweets. To detect a target event, we devise a classifier of tweets based on features like keywords in a tweet, the number of words, and their context. Users are using Twitter to report real-life events. It focuses on detecting those events by analyzing these text streams in Twitter. The characteristics of Twitter make it a non-trivial task. The traffic detection system was employed for real-time monitoring of many areas of the road network, that allow for detection of traffic events almost in real time.

Keywords—Twitter, Traffic event detection, tweet classification, text mining, social sensing.

I. INTRODUCTION

We use Social Network sites, also called micro-blogging services (e.g., Twitter, Facebook, Google+), have spread in recent years, becoming a new kind of real-time information channel. Social media sites for short messages like as Twitter have become a powerful tool for real-time information sharing on a large scale. The popularity for the characteristics of portability of many social networks applications for smartphones and tablets, easiness of use, and real-time of nature[1],[2]. Twitter currently produces 340,000,000 tweets per day from more than 140,000,000 active users1. Many users post message related to specific real-world events. Current event detection methods in social media streams can be classified into three types: cluster based, model-based, and those based on signal processing, see [2] for summary. An in-depth analysis of how breaking news spread on Twitter is also provided in [3]. The detection of events on Twitter in real-time is challenging for several reasons. Social networks allow people to create an identity and share it in order to build a community.

The user message shared in social networks is called Status Update Message(SUM), and it contains, some part from the text, meta-information like as timestamp, geographic coordinates (latitude and longitude), name of the user, links to other resources, hashtags, and mentions. In this paper, present our method for real-time detection of real-world events.

Social media platforms are widely used for sharing information about the detection of events, such as traffic congestion, incidents, natural ruins (earthquakes, storms, fires, etc.), or other events. An event is defined as a real-world occurrence that happens in a specific time and space [1], [7]. Generally traffic related events, people often share by means of a SUM information about the current traffic situation around them while driving. For this reason, event detection from social networks is also often employed with Intelligent Transportation Systems (ITSs). ITSs provide, real-time information about weather, traffic congestion or regulation, or plan efficient (e.g., shortest, fast driving, least polluting) routes [4], [6], [8]–[14].

Event detection from social networks analysis is a more challenging problem than event detection from traditional media like blogs, emails, etc. In fact, SUMs are unstructured and irregular texts, it contains informal or shortened words, misspellings or grammatical errors [1]. SUMs contain a huge amount of not useful or meaningless information [15], which has to be filtered. According to Pear Analytics, it has been estimated that over 40% of all Twitter2 SUMs (i.e., tweets) is pointless with no useful information for the audience. For all of these reasons, in order to analyze the information coming from social networks or text mining techniques [17], We use to extract important information [18], of data mining, machine learning, statistics and Natural Language Processing (NLP).

Text mining denotes to the process of automatic extraction of important information and knowledge from unstructured text. Text mining is a variation on a field called data mining [2], that tries to find interesting pattern from large databases. Text mining, is also known as Intelligent Text Analysis, Text Data Mining or Knowledge Discovery in Text (KDT), refers
generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. As most information (over 80%) is stored as text, text mining is believed to have high commercial potential value. Knowledge can be discovered from many sources of information, unstructured texts remain the largest readily available source of knowledge. Most of text mining techniques are based on the idea that a document can be faithfully represented by the set of words contained in it (bag-of-words representation [21]). According to this representation, each document j of a collection of documents is represented as an M-dimensional vector

$$V_j = \{w(t_{j1}), \ldots, w(t_{ji}), \ldots, w(t_{jM})\},$$

where M is the number of words defined in the document collection, and w(tji) specifies the weight of the word ti in document j. The simplest weighting method assigns a binary value to w(tji), thus indicating the absence or the presence of the word ti, while other methods assign a real value to w(tji). During the text mining process, several operations can be performed [21], depending on the specific goal, such as: i) linguistic analysis through the application of NLP techniques, indexing and statistical techniques, ii) text filtering by means of specific keywords, iii) feature extraction, i.e., conversion of textual features (e.g., words) in numeric features (e.g., weights), that a machine learning algorithm is able to process, and iv) feature selection, i.e., reduction of the number of features in order to take into account only the most relevant ones.

Data mining and machine learning algorithms (i.e., support vector machines (SVMs), decision trees, neural networks, etc.) are applied to the documents in the vector space representation, to build classification, clustering or regression models. Twitter is a good source of information for real-time event detection and analysis. Twitter has some advantages concluded the similar micro-blogging services. First, tweets are up to 140 characters, enhancing the real-time and news-oriented nature of the platform. In fact, the life-time of tweets is usually very short, So Twitter is the social network platform that is best suited to study SUMs related to real-time events [22]. In this paper, we propose an intelligent system, based on text mining and machine learning algorithms, for real-time detection of traffic events from Twitter stream analysis. These technologies and techniques have been analyzed, tuned, adapted, and integrated in order to build the intelligent system. The most effective among different state-of-the-art approaches for text classification. The chosen approach was integrated into the final system and used for the on-the-field real-time detection of traffic events.

The paper has the following structure. Section II summarizes related work about event detection from social Twitter stream analysis. Section III outlines the architecture of the proposed system for traffic detection, by describing the methodology used to collect, elaborate, and classify SUMs, with particular reference to SUMs extracted from the Twitter stream. Section IV describes the setup of the system. Section V presents the results completed with different classification models and provides a comparison with similar works in the literature. Section VI presents the real-world monitoring application for real-time detection of traffic events. Finally, Section VII provides concluding remarks.

II. RELATED WORK

With reference to current approaches for using social media to extract useful information for event detection, we need to distinguish between small-scale events and large-scale events. Small-scale events (e.g., traffic, car crashes, fires, or local manifestations) usually have a small number of SUMs related to them, belong to a precise geographic location, and are concentrated in a small time interval. On the other hand, large-scale events (e.g., earthquakes or the election of a president) are characterized by a huge number of SUMs, and by a wider temporal and geographic coverage [24]. Consequently, due to the smaller number of SUMs related to small-scale events, small-scale event detection is a non-trivial task. Several works in the literature deal with event detection from social networks. Many works deal with large-scale event detection [6], [25]–[28] and only a few works focus on small-scale events [9], [12], [24], [29]–[31].

Regarding large-scale event detection, Sakaki et al. [6] use Twitter streams to detect earthquakes and typhoons, by monitoring special trigger-keywords, and by applying an SVM as a binary classifier of positive events (earthquakes and typhoons) and negative events (non-events or other events). In [25], the authors present a method for detecting real-world events, such as natural disasters, by analyzing Twitter streams and by employing both NLP and term-frequency-based techniques. Chew et al. [26] analyze the content of tweets shared during the H1N1 (i.e., swine flu) outbreak, containing keywords and hashtags related to the H1N1 event to determine the kind of information exchanged by social media users. De Longueville et al. [27] analyze geo-tagged tweets to detect forest fire events and outline the affected area.

Regarding small-scale event detection, Agarwal et al. [29] focus on the detection of fires in a factory from Twitter stream analysis, by using standard NLP techniques and a Naive Bayes (NB) classifier. In [30], information extracted from Twitter streams is merged with information from emergency networks to detect and analyze small-scale incidents, such as fires. Using NLP techniques and syntactic analysis, traffic information from microblogs to detect and classify tweets containing place mentions and traffic information. The field of natural language processing has produced technologies that teach computers natural language so that they analyze, understand, and even generate text. Some of the technologies [3] that have been developed and used in text mining process are information extraction, topic tracking, summarization, categorization, clustering is concept linkage, information visualization, and question answering. In the following sections we discuss each of these technologies and the role that play in text mining. The system focuses on Crime and Disaster-related Events (CDE) such as shootings, thunderstorms, and car accidents, and aims to classify tweets.
as CDE events by exploiting a filtering based on keywords, spatial and temporal information, number of followers of the user, number of retweets, hashtags, links, and mentions. Sakaki et al. [9] extract, based on keywords, real-time driving information by analyzing Twitter’s SUMs, and use an SVM classifier to filter “noisy” tweets not related to road traffic events.

In this paper, we focus on a particular small-scale event, i.e., road traffic, and we aim to detect and analyze traffic events by processing users’ SUMs belonging to a certain area and written in the Italian language. To this aim, we propose a system able to fetch, elaborate, and classify SUMs as related to a road traffic event or not. However, with respect to our work, all of them focus on languages different from Italian, employ different input features and/or feature selection algorithms, and consider only binary classifications. The proposed system may approach both binary and multi-class classification problems. As regards multi-class classification, we split the traffic-related class into two classes, namely traffic congestion or crash, and traffic due to external event. In this paper, with external event we refer to a scheduled event (e.g., a football match, a concert), or to an unexpected event (e.g., a flash-mob, a political demonstration, a fire). In this way we aim to support traffic and city administrations for managing scheduled or unexpected events in the city. Moreover, the proposed system could work together with other traffic sensors (e.g., loop detectors, cameras, infrared cameras) and ITS monitoring systems for the detection of traffic difficulties, providing a low-cost wide coverage of the cameras) and ITS monitoring systems for the detection of traffic difficulties, providing a low-cost wide coverage of the cameras.

In this section, our traffic detection system based on Twitter streams analysis is presented. The system architecture is service-oriented and event-driven, and is composed of three main modules, namely: i) Fetch of SUMs and Pre processing, ii) Elaboration of SUMs, iii) Classification of SUMs. The purpose of the proposed system is to fetch SUMs from Twitter, to process SUMs by applying a few text mining steps, and to assign the appropriate class label to each SUM. Finally, as shown in Fig. 1, by analyzing the classified SUMs, the system is able to notify the presence of a traffic event. The main tools we have exploited for developing the system are: 1) Twitter’s API, 4 which provides access to the public stream of tweets; 2) Twitter4J, 5 a Java library that we used as a wrapper for Twitter’s API; 3) The Java API provided by Weka (Waikato Environment for Knowledge Analysis) [32], which we mainly employed for data pre-processing and text mining elaboration. We recall that both the “Elaboration of SUMs” and the “Classification of SUMs” modules require setting the optimal values of a few specific parameters, by means of a supervised learning stage. To this aim, we exploited a training set composed by a set of SUMs previously collected, elaborated, and manually labeled. Section IV describes in greater detail how the specific parameters of each module are set during the supervised learning stage. In the following, we discuss in depth the elaboration made on the SUMs by each module of the traffic detection system.

A. Fetch of SUMs and Pre-Processing

The first module, “Fetch of SUMs and Pre-processing”, extracts raw tweets from the Twitter stream, based on one or more search criteria (e.g., geographic coordinates, keywords appearing in the text of the tweet). Each fetched raw tweet contains: the user id, the timestamp, the geographic coordinates, a retweet flag, and the text of the tweet. In this paper, we took only Italian language tweets into account. However, the system can be easily adapted to cope with different languages.

After the SUMs have been fetched according to the specific search criteria, SUMs are pre-processed. In order to extract only the text of each raw tweet and remove all meta-information associated with it, a Regular Expression filter [33] is applied. The meta-information discarded are: user id,
B. Elaboration of SUMs

The second processing module, “Elaboration of SUMs”, is devoted to transforming the set of pre-processed SUMs, i.e., a set of strings, in a set of numeric vectors to be elaborated by the “Classification of SUMs” module. In this same text mining techniques are applied in sequence to the pre-processed. In the following, the text mining steps performed in this module are described in detail:

a) tokenization is typically the first step of the text mining process, and consists in transforming a stream of characters into a stream of processing units called tokens (e.g., syllables, words, or phrases). During this step, other operations are usually performed, such as removal of punctuation and other non-text characters [18], and normalization of symbols (e.g., accents, apostrophes, hyphens, tabs and spaces). In the proposed system, the tokenizer removes all punctuation marks and splits each SUM into tokens corresponding to words (bag-of-words representation). At the end of this step, each SUM is represented as the sequence of words contained in it. We denote the jth tokenized SUM as $SUM_j^T = \{t_{j1}^T, \ldots, t_{jh}^T, \ldots, t_{jH_j}^T\}$, where $t_{jh}$ is the hth token and $H_j$ is the total number of tokens in $SUM_j$.

b) stop-word filtering consists in eliminating stop-words, i.e., words which provide little or no information to the text analysis. Common stop-words are articles, conjunctions, prepositions, pronouns, etc. The authors in [35] have shown that the 10 most frequent words in texts and documents of the English language are about the of the tokens given file. In the proposed system, the stop-word list in Italian language was freely downloaded from the Snowball Tartarus website and extended with other ad hoc defined stop-words. At the end of this step, each SUM is thus reduced to a sequence of relevant tokens. We recall that a relevant token is a token that does not belong to the set of stop words.

$SUM_j^{SW} = \{t_{j1}^{SW}, \ldots, t_{jk}^{SW}, \ldots, t_{jK_j}^{SW}\}$.

c) stemming is the process of reducing each word (i.e., token) to its stem or root form, by removing its suffix. The purpose of this step is to group words with the same theme having closely related semantics. In the proposed system, the stemmer exploits the Snowball Tartarus Stemmer [7] for the Italian language, based on the Porter’s algorithm [36]. Hence, at the end of this step each SUM is represented as a sequence of stems extracted from the tokens contained in it.

$SUM_j^S = \{t_{j1}^S, \ldots, t_{jL_j}^S\}$.

d) stem filtering consists in reducing the number of stems of each SUM. In particular, each SUM is filtered by removing from the set of stems the ones not belonging to the set of relevant stems. The set of $F$ relevant stems

$$RS = \{\hat{s}_1, \ldots, \hat{s}_f, \ldots, \hat{s}_F\}$$

is identified during the supervised learning stage that will be discussed in Section IV. At the end of this step, each SUM is represented as a sequence of relevant stems. We denote the jth filtered SUM as

$$SUM_j^{SF} = \{t_{j1}^{SF}, \ldots, t_{jF_j}^{SF}\}.$$

C. Classification of SUMs

The third module, Classification of SUMs, assigns to each elaborated SUM a class label related to traffic events. So the output of this module is a collection of $N$ labelled SUMs. The parameters of the classification model have been identified during the supervised learning stage. The classifier that achieved the most accurate results was finally employed for the real-time monitoring with the proposed traffic detection system. The system continuously monitors a specific region and notifies the presence of a traffic event on the basis of a set of rules that can be defined by the system administrator. For example, when the first tweet is recognized as a traffic-related tweet, the system may send a warning signal. Then, the actual notification of the traffic event may be sent after the identification of a certain number of tweets with the same label.

IV. SETUP OF THE SYSTEM

As stated previously, a supervised learning stage is required to perform the setup of the system. In particular, we need to identify the set of relevant stems, the weights associated with each of them, and the parameters that describe the classification models. During the learning stage, each SUM is elaborated by applying the tokenization, stop-word filtering, and stemming steps. Then, the complete set of stems is built as follows:
CS is the union of all the stems extracted from the $N_{tr}$ SUMs of the training set. We recall that $SUM_j$ is the set of stems that describes the $j$th SUM after the stemming step in the training set. Then, we compute the weight of each stem in $CS$, which allows us to establish the importance of each stem $sq$ in the collection of SUMs of the training set, by using the Inverse Document Frequency (IDF) index as $wq = IDFq = \ln(N_{tr}/Nq)$, where $Nq$ is the number of SUMs of the training set in which the stem $sq$ occurs [37].

V. EVALUATION OF THE TRAFFIC DETECTION SYSTEM

In this section, we discuss the evaluation of the proposed system. We performed several experiments using two different datasets. For every dataset, we built and compared seven different classification models: SVM, NB, C4.5, $k$NN (with $k$ equal to 1, 2, and 5), and PART. The setup of the system, we recall the employed classification models. Then, we present the achieved results, and the statistical metrics used to evaluate the performance of the classifiers.

A. Description of the Datasets

We built two different datasets, i.e., a 2-class dataset, and a 3-class dataset. For each dataset, tweets in the Italian language were collected using the “Fetch of SUMs and Pre-processing” module by setting some search criteria (e.g., presence of keywords, geographic coordinates, date and time of posting).

1) 2-Class Dataset: The first dataset consists of tweets belonging to two possible classes, namely i) road traffic-related tweets (traffic class), and ii) tweets not related with road traffic (non-traffic class). The tweets were fetched in a time span of about four hours from the same geographic area.

Table II shows some of the most important textual features (i.e., stems) and their meaning, related to the traffic class tweets, identified by the system for this dataset.

2) 3-Class Dataset: The second dataset consists of tweets belonging to three possible classes. In this case we want to discriminate if traffic is caused by an external event (e.g., a football match, a concert, a flash-mob, a political demonstration, a fire) or not. We took into account four possible external events, namely, i) matches, ii) processions, iii) music concerts, and iv) demonstrations. Thus, in this dataset the three possible classes are: i) traffic due to external event, ii) traffic congestion or crash, and iii) non-traffic. The tweets were fetched in a similar way as described before.

More in detail, first, we fetched candidate road traffic-related tweets due to an external event (traffic due to external event class) according to the following search criteria:

— geographic area of origin of the tweet: Italy, parameters set as in the case of the 2-class dataset;
— time and date of posting: parameters set as in the case of the 2-class dataset, but different hours of the same weekend days are used;
— keywords contained in the text of the tweet: for each external event aforementioned, we took into account only one keyword.

B. Employed Classification Models

In the following we briefly describe the main properties of the employed and experimented classification models. The best hyper-plane is the one with the maximum margin, i.e., the largest minimum distance, from the training samples and is computed based on the support vectors (i.e., samples of the training set). The NB classifier [45] is a probabilistic classification algorithm based on the application of the Bayes’s theorem, and is characterized by a probability model which assumes independence among the input features. In other words, the model assumes that the presence of a particular feature is unrelated to the presence of any other feature.

The $k$NN algorithm [50] belongs to the family of “lazy” classification algorithms. The basic functioning principle is the following: each unseen sample is compared with a number
of pre-classified training samples, and its similarity is evaluated according to a simple distance measure (e.g., we employed the normalized Euclidean distance), in order to find the associated output class.

C. Experimental Results

In this section, we present the classification results achieved by applying the classifiers mentioned in Section V-B to the two datasets described in Section V-A. For each classifier the experiments were performed using an \( n \)-fold stratified cross-validation methodology. In \( n \)-fold stratified cross-validation, the dataset is randomly partitioned into \( n \) folds and the classes in each fold are represented with the same proportion as in the original data. The classification model is trained on \( n-1 \) folds, and the remaining fold is used for testing the model. We recall that, for each fold, we consider a specific training set which is used in the supervised learning stage to learn both the pre-processing (i.e., the set of relevant stems and their weights) and the classification model parameters.

To evaluate the achieved results, we employed the most frequently used statistical metrics, i.e., precision, accuracy, recall, and F-score. In fact, in the case of a multi-class classification, the metrics are computed per class and the overall statistical measure is simply the average of the per-class measures. The correctness of a classification can be evaluated according to four values: i) true positives (TP): the number of real positive samples correctly classified as positive; ii) true negatives (TN): the number of real negative samples correctly classified as negative; iii) false positives (FP): the number of real negative samples incorrectly classified as positive; iv) false negatives (FN): the number of real positive samples incorrectly classified as negative. We employed again the classifiers previously introduced and the obtained results are shown in Table VII. The best classifier resulted to be again SVM with an average accuracy of 88.89%.

VI. REAL-TIME DETECTION OF TRAFFIC EVENTS

The established system was installed and tested for the real-time monitoring of several areas of the Italian road network, by means of the analysis of the Twitter stream coming from those areas. The aim is to perform a continuous monitoring of frequently busy roads and highways in order to detect possible traffic events in real-time or even in advance with respect to the traditional news media [55], [56]. The system is implemented as a service of a wider service-oriented platform to be developed in the context of the SMARTY project [23]. In this section, we aim to show the effectiveness of our system in determining traffic events in short time.

The system continuously performs the following operations: i) fetches, with a time frequency of \( z \) minutes, tweets originated from a given area, containing the keywords resulting from CondA, ii) performs a real-time classification of the fetched tweets, iii) detects a possible traffic-related event, by analysing the traffic class tweets from the considered area, and, if needed, sends one or more traffic warning signals with increasing intensity for that area.

VII. CONCLUSION

In this paper, we have proposed a system for real-time detection of traffic-related events from Twitter stream analysis. The system, built on a SOA, is able to fetch and classify streams of tweets and to notify the users of the presence of traffic events. The system is also able to discriminate if a traffic event is due to an external cause, such as football match, procession and manifestation, or not.

These technologies and techniques have been analyzed, tuned, adapted and integrated in order to build the overall system for traffic event detection. Among the analyzed classifiers, we have shown the superiority of the SVMs, which have achieved accuracy of 95.75%, for the 2-class problem, and of 88.89% for the 3-class problem, in which we have also considered the traffic due to external event class.

The best classification model has been employed for real-time monitoring of several areas of the Italian road network. We have shown the results of a monitoring campaign, performed in September and early October 2015. Different criteria for fetched candidate tweets for traffic class. We have discussed the capability of the system of detecting traffic events almost in real-time, often before online news web sites and local newspapers.

ACKNOWLEDGMENT

We would like to thank Fabio Cempini for the implementation of some parts of the system presented in this paper.
REFERENCES


