American-Eurasian Journal of Scientific Research 12 (5): 236-240, 2017
ISSN 1818-6785

DOI: 10.5829/idosi.aejsr.2017.236.240

An Effective Segmentation of Real Time Traffic Alerts by Improving NLP Stickiness Scores

[1]S. Singaravelan, [1]P. Gopalsamy, [1]D. Arun Shunmugam and [2]D. Murugan

[1]Department of Computer Science and Engineering,
P.S.R Engineering College, Sivakasi-626140, Tamilnadu, India
[2]Department of Computer Science and Engineering,
Manonmaniam Sundaranar University, Tirunelveli-627012, Tamilnadu, India

Abstract: We demonstrate the effectiveness that machine learning can bring to improve social media platforms through a case study on Twitter trending topics. Twitter has attracted millions of users to share and disseminate most up-to-date information, resulting in large volumes of data produced every day. However, many applications in Information Retrieval (IR) and Natural Language Processing (NLP) suffer severely from the noisy and short nature of tweets. In this paper, we propose a novel framework for tweet segmentation in a batch mode, called HybridSeg. Out of many issues we face in transportation today, road traffic has become the most crucial issue that directly affects our lives and economy. Despite of many implemented and progressing solutions, this issue seems to be remaining in a significant level in many countries and regions. HybridSeg finds the optimal segmentation of a tweet by maximizing the sum of the stickiness scores of its candidate segments. The stickiness score considers the probability of a segment being a phrase in English and the probability of a segment being a phrase within the batch of tweets. For the latter, we propose and evaluate two models to derive local context by considering the linguistic features and term-dependency in a batch of tweets, respectively. Experiments on two tweet data sets show that tweet segmentation quality is significantly improved by learning both global and local contexts compared with using global context alone. We are trying to cope with this limitation by introducing a real time natural language processing solution to generate machine readable road traffic alerts. We observe many potentials of transforming this raw data into a machine readable format.

Key words: Tweet Stream · Tweet Segmentation · Tokenization · Framework for HybridSeg · Machine Learning · NLP

INTRODUCTION

TWITTER is a popular microblogging and social network-ing service that presents many opportunities for researching natural language processing (NLP) and machine learning. It has attracted great interests from both industry and academia. Nevertheless, due to the extremely large volume of tweets published every day Due to its invaluable business value of timely information from these tweets, it is imperative to understand tweets language for a large body of downstream applications.

Twitter, as a recent type of social media having remendous growth in recent year. Many public and private sector have been described to monitor Twitter stream to collect and understand users' opinion about organizations. However, because of very large volume of tweets published every day, it is practically infeasible and unnecessary to monitor and listen the whole Twitter stream. Therefore, targeted Twitter streams are regularly monitored instead every stream contains tweets that possibly satisfy some information needs of the monitoring organization[2] tweeter is most popular media for sharing and exchanging information on local and global level[4] Targeted

Twitter stream is generally form by cleaning tweets with user-defined selection criteria depends on need of information.

Corresponding Author: S. Singaravelan, Department of Computer Science and Engineering, P.S.R Engineering College, Sivakasi-626140, Tamilnadu, India.

In this study we focused on one such solution emerged with the use of social media. It uses an online social networking service called Twitter.

Targeted Twitter stream is usually constructed by filtering tweets with predefined selection criteria (e.g., tweets published by users from a geographical region, tweets that match one or more predefined keywords). Due to its invaluable business value of timely information from these tweets, it is imperative to understand tweets' language for a large body of downstream applications.

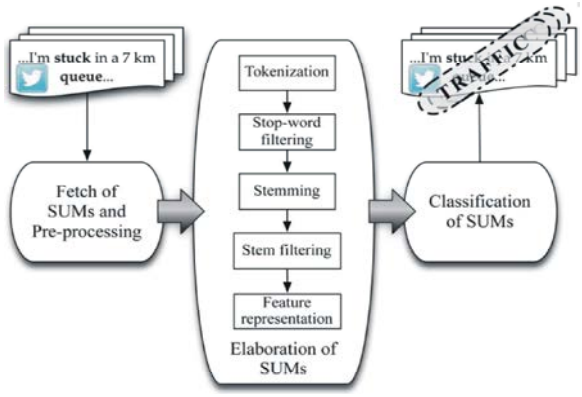

Fig. 1: Architecture of the Twit Stream processing.

In this paper, we focus on the task of tweet segmentation. The goal of this task is to split a tweet into a sequence of consecutive n-grams (n 1), each of which is called a segment. A segment can be a named entity (e.g., a movie title "finding memo"), a semantically meaningful information unit (e.g., "officially released"), or any other types of phrases which appear "more than by chance" [1]. Semantically meaningful segments "spare on effort", "traffic throughput" and "circle line" are preserved. Because these segments preserve semantic meaning of the tweet more precisely than each of its constituent words does, the topic of this tweet can be better captured in the subsequent processing of this tweet.

Background: Road.lk [1] is a website that provides localized traffic alerts from a Twitter feed with a specific mention. This website has its own Twitter account with the ID road lk [2]. If a particular person is experiencing road traffic or if he/she has information regarding road traffic, that person can tweet about that with a mention to road lk Twitter account. The same tweet will be manually retweeted by road lk account. Thus all the Twitter users who are following road lk account will get that particular traffic alert nearly in real-time. We can observe that this model works effectively because of many Twitter users who post traffic updates with road lk mention. This feed was identified as a potential source to extract information on road traffic in real-time. Furthermore the reliability of this source is maintained by the model itself due to higher number of publishers. If there is high traffic in a particular area, we get more traffic alerts from different users. By following these crowdsourced alerts a passenger or a driver can avoid high traffic areas and also can get notified on noteworthy incidents.

But still this model has several limitations such as con-nectivity requirement and unavailability of proper alert mech-anism except Twitter feed or road.lk website. Another notable limitation is imposed by the way users post their traffic updates. Twitter users use natural language to post traffic updates. An alert format has not been imposed by road lk to post tweets on road traffic. Processing tweets can be made more straightforward by imposing such a format, but it can significantly reduce the flexibility of sharing updates on road traffic. We identified the use of natural language processing as the best way to tackle this problem.

Related Work: Traffic monitoring and analysing became an active research area with the development of intelligent transportation systems. Efforts have been made by researchers to provide solutions to certain issues we face in traffic monitoring systems. High quality traffic information is required for effective traffic monitoring and analysing. Several data collection methods are available [3] for real-time traffic monitoring such as traditional on-road sensors, floating car data (FCD) [4], vision systems and crowd sourced social media. FCD and crowd sourced data sources are comparatively more cost-effective methods for traffic data collection. But crowdsourced data sources contain unstructured data and they require certain data cleaning and preprocessing beforehand the actual use. Ritter *et al*. [5] applied unsupervised clustered modelling for Twitter corpus of 1.3 million conversations. Wang *et al*. [6] proposed a real-time traffic alert and warning system based on a Latent Dirichlet Allocation based approach to classify traffic related tweets by using Twitter as a crowdsourced data source. Apart from these methods, most researchers have used vision systems as their data source. For an example we can consider the traffic scene analysis system which was developed by combining a low-level machine vision-based surveillance system with a high-level symbolic reasoner based on dynamic belief networks [7].

Traffic information is considered as an important mea-sure about transportation in most developed countries. Hence countries such as Spain [8] and Finland [9] maintain online traffic information systems to collect traffic data and share them with public. Furthermore, we can observe some existing mobile applications for traffic monitoring like Twitraffic [10]. Twitraffic is a commercial solution which can be used within UK and it also uses Twitter feed as their data source. Users can subscribe to locations to receive traffic related tweets. But these tweets are unprocessed and they can contain misleading information.

Solution: After analysing above mentioned use cases, we had to develop architecture for a solution which addresses multiple requirements. Mainly two architectures were implemented and tested in prototype level to select the optimal approach. In both architectures, multiple tools were utilized to retrieve, process and present information in most efficient way. Overview of the implemented architecture and high level system pipelines are respectively.

Tweet Segmentation: Given a tweet t from batch T , the problem of tweet segmentation is to split the ` words in t = w1w2 : : :w` into m consecutive segments, t = s1s2:::sm, where each segment si contains one or more words. We formulate the tweet segmentation problem as an optimization problem to maximize the sum of stickiness scores of the m segments, A high stickiness score of segment s indicates that it is a phrase which appears “more than by chance” and further splitting it could break the correct word collocation or the semantic meaning of the phrase. Formally, let C(s) denote the stickiness function of segment s. The optimal segmentation can be derived by using dynamic programming with a time complexity of O (L).

Application of Natural Language Processing: Despite the fact that road lk Twitter feed is a reliable data source to generate real-time road traffic alerts, its extent is largely constrained by natural language representation. If we can transform this data into a machine readable representation, we can use the full potential of this source for a better solution. In this study we propose a natural language processing (NLP) model to address this problem. NLP is an area derived from fields such as machine learning and human computer interaction which is concerned in removing or reducing the language gap between humans and computers by introducing tools and techniques that enable natural language understanding to computers. This objective has been considerably achieved by cutting edge NLP tools and we can see their applications in devices such as mobile phones and tablet computers. In this study, we were interested in extracting two entities from a tweet namely, location and traffic level. Timestamp which is essential in providing useful traffic alerts was already available in the dataset. To extract these two entities, our approach was to utilize NLP tools. Ahead of extracting these two entities, a tweet was needed to be classified to identify whether it is a traffic alert or not. Users tweet on topics other than road traffic with mentions to road lk. The NLP tasks required to classify and extract interested fields from a tweet can be listed as below.

- Tweet categorization
- Location extraction
- Traffic level extraction

In NLP terminology, first task is a document categorization task and the latter two are name entity recognition (NER) tasks. To implement these NLP tasks we used Apache OpenNLP toolkit which is a machine learning based toolkit for the processing of natural language text. It supports common NLP tasks such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing and reference resolution. Out of these we only used previously mentioned two functions for our three NLP tasks. Expectation-maximization, a lexical algorithm, plays a major role at the core of these functions. Three separate models were trained for each task by using the dataset generated by retrieved Twitter feed. Initial dataset which contained around 3000 tweets was split and manually tagged to train these models. A custom tokenizer was implemented to tokenize tweet text to words. The main reason to use a custom tokenizer instead of the tokenizer available in OpenNLP was due to complex name entity tagging. For an example some street names and city names became meaning-less when they were tokenized as separate words for location NER task. For traffic level NER task, a predefined set of words was selected to tag in tweet texts. In the training stage, traffic level NER model learns related words which are mostly used by users to express traffic level. Additionally we had to consider factors such as spelling mistakes, informal language and abbreviations when training traffic level NER model. These models were deployed in Siddhi language extensions, explained in next section.

Application of Complex Event Processing: We had to consider another important property in this particular data source when processing information; it was required to process this Twitter feed in real-time. but to generate useful alerts on road traffic, real-time processing was essential. A single data input is considered as an event and a continuous data input is identified as an event stream in CEP context. We used WSO2 Complex Event Processor [14] as the CEP tool to analyses and process Twitter feed input stream. At the core of WSO2 CEP the actual event processing is done by Siddhi Query Language (SiddhiQL) [15], [16]. It is designed to process event streams and identify complex event occurrences. Siddhi queries define how to process and combine existing event streams to create new event streams. When deployed in the Siddhi runtime, SiddhiQL queries process incoming event streams as specified by the queries and generate output event streams according to the query definition. SiddhiQL was extended to address our NLP requirements using language extensions.

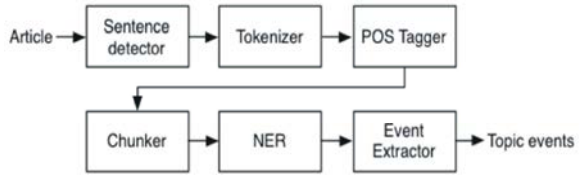

Fig. 2: Pipeline for traffic feed

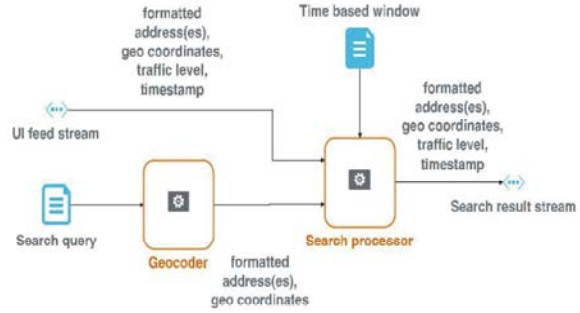

Fig. 3: Pipeline for traffic search

Three extensions were implemented for following real-time processing tasks.

- Tokenizer
- Name entity recognition
- Geocoder

Tokenization: Tokenization is a critical activity in any information retrieval model, which simply segregates all the words, numbers and their characters etc. from given document and these identified words, numbers and other characters are called tokens. Along with token generation this process also evaluates the frequency value of all these tokens present in the input documents. Pre-processing involves the set of all documents are gathered and passed to the word extraction phases in which all words are extracted. In this phase remove those english words which are useless in information retrieval these english words are known as stop words

Geocoder: Geocoding extension converts the locations extracted from tweet content into geo coordinates which are required in setting marker on geo map and for calculating nearby geo area. WSO2 CEP specific structures called event flows and execution plans were deployed to process incoming Twitter feed in real-time. The functionality of the use cases such as traffic feed and traffic search were implemented within CEP instead of having a dedicated back-end server for web UI and for geo location map. A few deployed Siddhi execution plans are given below.

```
from classifiedStream#transform.nlp:getEntities( convertedText,4,true,"/_system/governance/en-location.bin")

select * insert into templocationStream;
from classifiedStream#transform.nlp:getEntities(convertedText,1,false,"/_system/governance/en-trafficlevel.bin")

select * insert into temptrafficlevelStream;
from S1=classifiedStream, S2=temptrafficlevelStream, S3=templocationStream
select S1.createdAt as time, S2.nameElement1 as trafficLevel,
S3.nameElement1 as location1, S3. nameElement2 as location2,
S3.nameElement3 as location3, S3.nameElement4 as location4
insert into locationsStream;
```

```
from uiFeedStream#window.time(120 min) as trafficFeed join
SearchEventStream as request
on (trafficFeed.latitude < request.latitude + 0.018 and trafficFeed.latitude >
 request.latitude - 0.018 and trafficFeed.longitude < request. longitude +
0.027 and trafficFeed.longitude > request.longitude - 0.027)

select trafficFeed.formattedAddress, trafficFeed. latitude,
trafficFeed.longitude, trafficFeed. level, trafficFeed.time

insert into searchResult;
```

HybridSeg Framework: The proposed HybridSeg framework segments tweets in batch mode. Tweets from a targeted Twitter stream are grouped into batches by their publication time using a fixed time interval (e.g., a day). Each batch of tweets are then segmented by HybridSeg collectively.

Information Retrieval: Classical retrieval modeling considers documents as bags of words. This stands for the view of the model as an entity without structure where only the numbers of occurrences of terms are important for determining relevance.Whenever a query is posed to a retrieval system every document is scored with respect to the query. The scores are sorted and then final ranked

list is presented to the user. A retrieval model is in charge of producing these scores. In general models for retrieval.IR System do not care about efficiency: they solely focus on understanding a user's information need and the ranking process.

- The user's internal cognitive state or information need is turned into an external expression or query based on a query model.
- Each document is assigned a representation that indicates what the document is about and what topics it covers based on a document model.
- A similarity function can be used to estimate the relevance of a document to the information need based on the document model and on the query model.

Therefore the three classic models in information retrieval are called Boolean, Vector and Probabilistic. In the Boolean model documents and queries are represented as set of index terms. Thus we say that the model is set theoretic. In the vector model documents and queries are represented as vectors in the dimensional space. Thus we say that the model is algebraic. In the probabilistic model the framework for modeling document and query representations is based on probability theory [3] [4]. Thus as the name indicates we say that the model is probabilistic.

RESULTS

By implementing the approach specified in previous section, we obtained a system which can successfully retrieve and transform a Twitter feed into machine readable format. Using this outcome we implemented a web based interface to demonstrate the functionalities of our implementation. Users can interact with this interface and make use of the use cases we presented in section IIA screencast of the system demonstration is available at WSO2 library [18]. Apart from this web interface, we present accuracy measures of our NLP models which were deployed in CEP tool in tables I. OpenNLP provides APIs to get common accuracy measures for trained models. Document categorizer model accuracy was calculated manually due to the unavailability of an API published tweets to the system from a tweet archive. The system was responsive in total running time and the traffic feed was updated on multiple web interfaces in real-time. Alerts were generated as email notifications to respective subscribers.

Table 1: Performance measures for traffic level NER model

	Cross validation (10%)	Test results
Precision	1.0	0.7083333333
Recall	0.8888888888	0.8095238095
F-measure	0.9411764706	0.7555555556

DISCUSSION AND CONCLUSION

In this document we presented a solution to extract useful information from a crowdsourced social networking service by utilizing a NLP/CEP combined approach. Results of this study demonstrate the potential of such model to cope with an application of real-time natural language processing task. Still the scenarios we have demonstrated in this study do not cover all possible use cases of this approach.

REFERENCES

1. Athuraliya, C.D., M.K.H. Gunasekara, Srinath Perera and Sriskandarajah Suhothayan, 20196. Real-time Natural Language Processing for Crowdsourced Road Traffic Alerts, IEEE 2016.
2. Multiverse, 2015. Traf?c reports for Sri Lanka, Jul. 2015. [Online]. Available: https://road.lk/traf?c/
3. Road, L.K., 2015. road.lk (@road lk) | Twitter, Jul. 2015. [Online]. Available: https://twitter.com/road\ lk
4. Sch¨afer, R., K. Thiessenhusen and P. Wagner, 2002. A traffic information system by means of real-time floating-car data, in ITS World Congress, pp: 2.
5. Wang, D., A. Al-Rubaie, J. Davies and S. Clarke, 2014. Real time road traf?c monitoring alert based on incremental learning from tweets, in Evolving and Autonomous Learning Systems (EALS), 2014 IEEE Symposium on, pp: 50-57.
6. Directorate General of Traffic, 2015. Informacin de trfico (Traffic information), 2015. [Online]. Available: http://infocar.dgt.es/etraffic/ [9] Finnish Transport Agency, "Traffic Situation Service, 2015. [Online]. Available: http://www.finnra.fi/alk/english/.
7. Targett, R., 2013. Twitraffic - UK Traffic Information, 2013. [Online].Available: http://twitraffic.co.uk/
8. WSO2, WSO2 Enterprise Service Bus, 2015. [Online]. Available:http://wso2.com/products/enterprise-service-bus/
9. The Apache Software Foundation, Apache OpenNLP, 2010. [Online]. Available: https://opennlp.apache.org/.