Real Time Traffic Detection using Semantic Analysis

Semil Jain
BrowserStack Inc.
Mumbai, India

Riya Singh
Quantiphi Analytics Solutions Pvt. Ltd.
Mumbai, India

ABSTRACT

Social networks have recently come across as a great source of information for detecting congestion, accidents, as well as crowding due to numerous festivals. Twitter is one of the most popular sites because it expresses knowledgeable information in minimum words. Since the tweets have limited words, processing it becomes easy. Hence, This project uses Twitter as a source of information. This project focuses on classifying whether a tweet is a traffic related tweet or not using Semantic technologies. In this project, the system fetches the tweets using the Twitter Api and then pre-process it, converting into tokens and cleaning all noise. These tokens are then semantically processed and classification is performed using Naïve Bayes multinomial classifier. Different types of tweets data are used for prediction, including tweets from selected road-traffic Twitter accounts, tweets that contain road-traffic-related keywords and geo-tagged tweets. These classified results are then plotted as a colored path on the android application. Such information can help the traveler to make a better travel plan.

Keywords

Semantic Analysis, geo-tagged tweets, Naïve Bayes multinomial classifier, Tokenization, Stop-word filtering, Stemming, Stem filtering, Feature representation.

1. INTRODUCTION

One of the biggest problems that we face in our day to day lives is traffic congestion. Everyone's in the same boat, facing this issue all around the world. There is a miserable waste of time happening in these congestions. One of the solutions to tackle this issue is to use hardware sensors like inductive loops and cameras to monitor traffic status. These tools although effective, have some limitations. High maintenance costs being one of them. One other limitation is that these tools are only effective within certain parameters and are created to collect specific type of information only, like vehicle count. People share statuses, information and opinions in short messages via web applications called Micro-bloggers. These applications provide concise and effective way of communication. Twitter is an example of this service. It gives people a platform to express their views in less than 280 characters. Twitter offers people from various domains like marketers, researchers and decision makers an access to such useful information and data as users share their stories.

Analysing the content of Twitter messages might provide a better understanding of the traffic congestions in terms of why, when, and where does it happen. The limitations of the hardware sensors are overcome by extracting this traffic information. It will also serve as free and easy access to such information for the common public to help them in avoiding traffic spots. Twitter has been proved as a major social media platform that focuses on event detection and has gained a lot of popularity recently; it counts more than 199 million active users, exchanging more than 500 million status update messages per day. In this report, A system is proposed, based on semantic analysis on extracted text and machine learning algorithms, for the detection of traffic/congestion events from Twitter stream analysis. After a feasibility study, this system has been proposed and developed(prototype) from the ground as a Service Oriented Architecture (SOA). The system uses available technologies based on the latest techniques for data analysis and pattern classification. These techniques and technologies have been analyzed, brainstormed, and integrated in order to build this system.

2. AIMS & OBJECTIVES

To avoid traffic congestion spots, real-time traffic information is important. Twitter messages can be proved as a helpful source of information. The existing methodologies that use text mining methods have a limitation in that they depend only on geo-tagged tweets. Here, the aim is to combine a backend server with the android application to display results with a semantic analysis method for filtering.

This project aims at designing a method to extract information about traffic from tweets and plot it on an open-source map. The method is favorable to the existing method, as it is based on the integration between a text mining method and a geo-locating method.

Objectives:

1) Building a system for filtering the tweets and getting the tweets related to traffic.

2) Make use of the official news Twitter accounts that report about traffic.

3) Use text mining method to analyze and extract information from tweets.

4) Based on their content, geo-locate the tweets.

5) Classify the tweets into traffic and non-traffic categories.

6) Create a workflow to show traffic indication based on the tweets.

3. REVIEW OF LITERATURE

This part of the paper covers the review of literature i.e. the review of all the existing solutions as well as existing papers related to the topic and gives a brief idea about the technologies used currently.

3.1 Domain Explanation

A domain is a field of study that defines a set of common requirements, terminology, and functionality for any software program or system constructed. This part covers the different requirements for our system to be constructed.

3.1.1 Semantic Analysis. The semantics of a language provide meaning to its words, like the structure of syntax and token. They give us an interpretation of symbols, their types, and how they relate to each other. Whether the syntax structure built in the source program derives any meaning or not is a part of semantic analysis. Semantic analysis is a key process in processing the tweets received and then classifying them by performing sentiment analysis. The aim is to construct a model that will calculate the tone (positive, neutral, negative) of the text. To perform this, the training of the model will need to be on the existing traffic data. The class (neutral, positive, negative) of new messages (test data that were not used to build the model) will be determined from the resulting model with maximum accuracy.
Detecting and geo-locating events from information extracted from tweets is a relatively new emerging research area. Recently tweets from Twitter are also used for detecting different events like traffic, earthquakes, and disasters. These studies used Text mining concepts and Natural Language Processing (NLP) methodologies to extract information from tweets. Some studies use Named Entity Recognition system (NER) to extract information from tweets. The job of NER is to identify words in the sentence into predefined categories like organizations, locations, names of persons, and expressions of time. To train the system NER uses machine learning classifiers. These machine learning classifiers are like Maximum Entropy taggers, Support Vector Machine (SVM), and Conditional Random Field (CRF). NER systems use CoNLL2003 which is a benchmark dataset for training and validating systems.[2] Some other studies use Part Of Speech (POS) tagging for information extraction. POS aims at generating a label for each word in the sentence indicating its syntactic role with a tag. One of the used methods for text mining is tokenization. The tokenization method breaks the text into tokens (words). Using the custom dictionary, these tokens are categorized into custom tags.[2] Geo-locating Tweets To detect events using Twitter messages, the location where these tweets are being issued is important to know. This location indicates where the event has actually happened. In August 2009 Twitter issued a geo-tagging feature that associates the user's current location in the form of latitude and longitude values with each tweet. This feature will work only if the user enables it.[3] Some studies about event detection depend on Twitter geotagging features for estimating the location of the event(MacEachren et al., 2011). Other methods used the location information associated with the user account.[3] Semantic analysis separates phrases into tokens and gives meaning to the tokens by classifying them. In this project, the words related to road traffic are extracted and then grouped, giving them meaning, hence indicating traffic or no traffic. For this reason, a large lexical database of a desired language (here: English) is used which groups the word and gives meaning. Some examples of such databases are WordNet and Wiktionary.

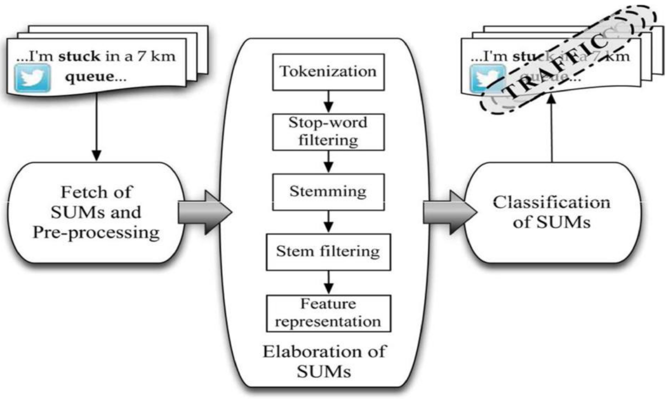


Fig. 1. Semantic Analysis [1]

3.1.2 WordNet. WordNet is a huge lexical database of English. Verbs, adjectives, nouns, and adverbs are grouped into sets of smartly related synonyms (synsets), each showcasing a distinct concept. To look up words in WordNet, Synset is a special kind of interface that is present in NLTK. Synset instances are the forming of similar words that express the same concept. The final network of meaningful connected concepts and words can be traversed with the browser. WordNet is also open-source, publicly available for download for free. The structure of WordNet makes it a useful tool for NLP and computational linguistics.
WordNet groups words together based on their meanings, superficially making it a thesaurus. However, we also need to consider some important things. First, WordNet connects not just word forms, characters of letters but specific meanings of words. As a result, words that are related closely to one another in the library are semantically identical. Secondly, the semantic relations among words are labeled by WordNet, whereas in a thesaurus, the groupings of words do not follow any obvious pattern other than the similarity in meaning.[5]

3.2 Existing Solutions

3.2.1 Google Maps Traffic Indication. Over the years, Google has collected a history of what congestion is usually like at specific times on particular roads. This indicates traffic patterns can be predicted by Google. The density of traffic is indicated by different colored routes on the map. It uses satellites as a tool to determine traffic in an area. [4]

3.2.2 Hardware sensors like cameras, inductive loops and radars to monitor traffic status. These tools although effective, have some limitations. High maintenance costs being one of them. One other limitation is that these tools are only effective within certain parameters and are created to collect specific type of information only, like vehicle count.[2]

3.2.3 Only Geo-tagged tweets. The existing methodologies that use text mining methods have a limitation in that they depend only on geo-tagged tweets. No proper filtering and classification is provided which we aim to encounter here in this project.

4. PROPOSED SYSTEM

The methodology followed in this project will start with reviewing the literature for text mining methods and geo-locating methods. The methods that seem to fulfil the research objectives will be selected. Then, the two methods will be combined and a prototype of the proposed system will be implemented.

The results of the prototype proposed will be tested to see how reliable the proposed system is. The testing will be based on comparing the results with the traffic status feature of Google maps. For the incomparable results some updates will be done on the algorithm. Then, the prototype will be retested after the changes. The last step will be repeated until good results are achieved.

The system is divided into 2 subsystem- android application and web service.

Android application - It takes user location (source) and destination as input. The input is passed to web service for processing. If traffic is there on route notify the user and suggest alternate route.

Web service - It takes Twitter tweets as input and classifies them as traffic tweets and non-traffic tweets. It makes use of open source NLP Libraries for classification. The main aim is detection of traffic related events from Twitter. The system detects traffic tweets in real-time.

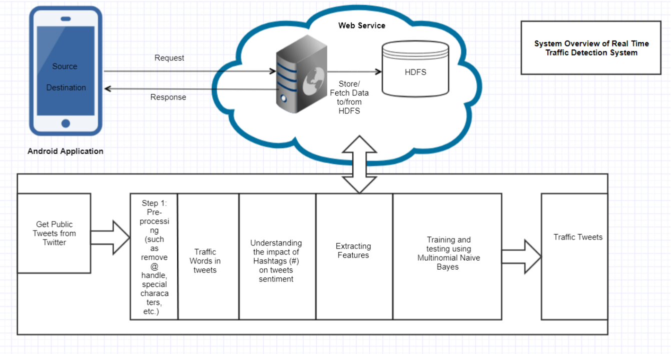

Source
Destination
Android Application
Request
Response
Web Service
Store/ Fetch Data to/from HDFS
HDFS
System Overview of Real Time Traffic Detection System
Get Public Tweets from Twitter
Step 1: Pre-processing (such as remove @ handle, special characaters, etc.)
Traffic Words in tweets
Understanding the impact of Hashtags (#) on tweets sentiment
Extracting Features
Training and testing using Multinomial Naive Bayes
Traffic Tweets

Fig. 2. Proposed System

5. IMPLEMENTATION DETAILS

5.1 Modules

The system implementation is divided into 5 modules as

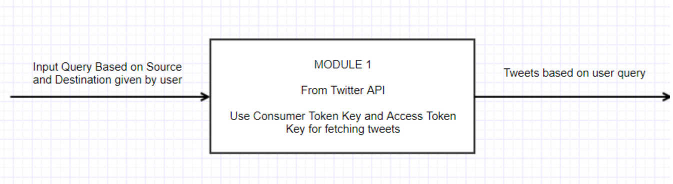

Input Query Based on Source and Destination given by user
MODULE 1
From Twitter API
Use Consumer Token Key and Access Token Key for fetching tweets
Tweets based on user query

Fig. 3. Module 1

Module 1: Steps for taking real-time public tweets require user registration on the Twitter website. After registration -¿ Twitter developer options -¿ create an application -¿ enter the project name. Then, generate four keys: consumer key and consumer secret key, access token key, and access token secret key. Each key is distinct as it has its own permission for access. The purpose of the consumer key is to allow access login of the user via key directly to Twitter. The consumer token key's role is to read and write tweets into the application. The other two keys are for accessing the Twitter account to read and write posts. One can get real-time tweets as input using all these four keys.

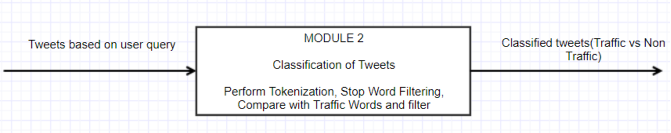

Tweets based on user query
MODULE 2
Classification of Tweets
Perform Tokenization, Stop Word Filtering, Compare with Traffic Words and filter
Classified tweets(Traffic vs Non Traffic)

Fig. 4. Module 2

Module 2: Classify traffic-related tweets. This module classifies tweets and identifies sentiments of each tweet in positive, negative, and neutral. Further processing is done by web service. First of all, take Tweets, split tweets into tokens as called Tokenize, then remove stop words from tweets and after that compare words with traffic-related words called Filter and finally get fully traffic-related tweets to define the class of each tweet.

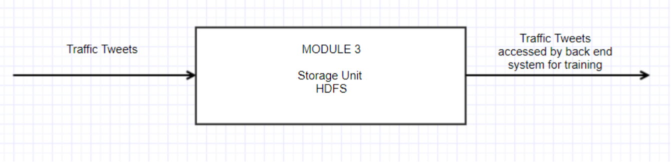


Fig. 5. Module 3

Module 3: Store data in HDFS. All traffic related tweets are stored in Hadoop distributed file system (HDFS) or some similar base. HDFS is used in the system because fast storing, retrieving and processing data.

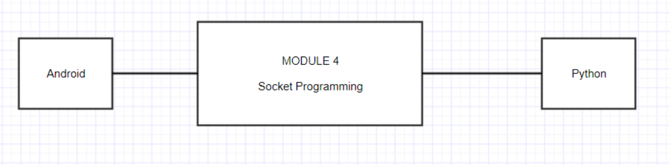


Fig. 6. Module 4

Module 4: Connectivity between Web service and Android application. The 2 subsystem of project are connected using socket programming.

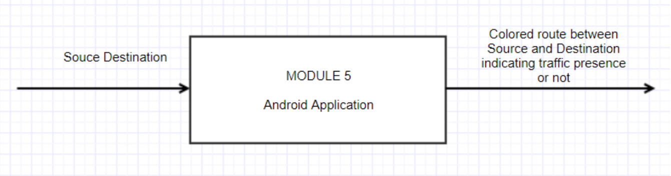

Souce Destination
MODULE 5
Android Application
Colored route between Source and Destination indicating traffic presence or not

Fig. 7. Module 5

Module 5: The android application is used to display the route.

5.2 Naïve Bayes

The Naive Bayes is an easy way to create classifiers: models that offer class labels in problematic situations, are represented as velvety values, where class labels are drawn on a limited set. There are more than one algorithms for training such classifiers based on the same principle: all Bayes classifiers are classified as the value of a particular element more independent than any other element, given the class flexibility. Naive Bayes is a model of conditional possibilities: given an example of a problem to be subdivided, represented by a vector representing some of the features n (independent variables), giving this situation opportunities for each K result or class. The problem with the above design is that if the number of elements n is large or if the element can take up a large number of values, supporting such a model in the tables is probably not possible. So we redesigned the model to make it more flexible. Using the Bayes' theorem, conditional opportunities can be squandered.

Naïve Bayes with Scikit-learn

The following steps are used in the project:

1) Defining dataset: We downloaded a traffic dataset that classifies data into traffic and non-traffic from Mendeley.

2) Encoding Features: Convert string into label numbers i.e. Traffic indicated by 1 and No traffic by 0.

3) Generating Models: Combine both steps and perform classification.

6. RESULTS AND EVALUATION

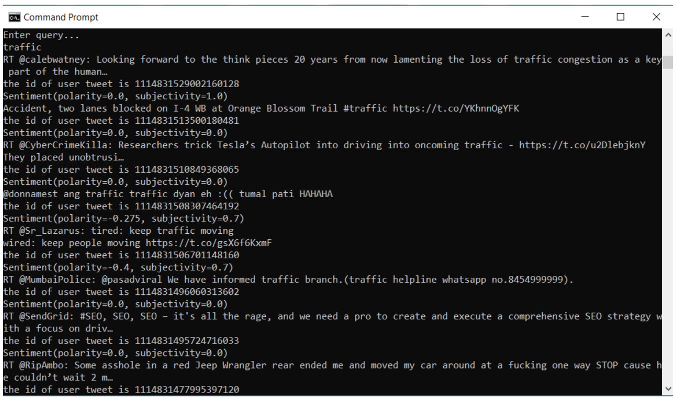
6.1 Result

Input:

1. The user enters the source and destination location in the android application. This location is sent to the backend python algorithm.

2. The classification algorithm then works on these tweets and produces the output.

Output:



```
Enter query...
traffic
RT @calebwatney: Looking forward to the think pieces 20 years from now lamenting the loss of traffic congestion as a key
 part of the human…
the id of user tweet is 1114831529002160128
Sentiment(polarity=0.0, subjectivity=1.0)
Accident, two lanes blocked on I-4 WB at Orange Blossom Trail #traffic https://t.co/YKhnnOgYFK
the id of user tweet is 1114831513500180481
Sentiment(polarity=0.0, subjectivity=0.0)
RT @CyberCrimeKilla: Researchers trick Tesla’s Autopilot into driving into oncoming traffic - https://t.co/u2DlebjknY
They placed unobtrusi…
the id of user tweet is 1114831510849368065
Sentiment(polarity=0.0, subjectivity=0.0)
@donnamest ang traffic traffic dyan eh :(( tumal pati HAHAHA
the id of user tweet is 1114831508307464192
Sentiment(polarity=-0.275, subjectivity=0.7)
RT @Sr_Lazarus: tired: keep traffic moving
wired: keep people moving https://t.co/gsX6f6KxmF
the id of user tweet is 1114831506701148160
Sentiment(polarity=-0.4, subjectivity=0.7)
RT @MumbaiPolice: @pasadviral We have informed traffic branch.(traffic helpline whatsapp no.8454999999).
the id of user tweet is 1114831496060313602
Sentiment(polarity=0.0, subjectivity=0.0)
RT @SendGrid: #SEO, SEO, SEO – it's all the rage, and we need a pro to create and execute a comprehensive SEO strategy w
ith a focus on driv…
the id of user tweet is 1114831495724716033
Sentiment(polarity=0.0, subjectivity=0.0)
RT @RipAmbo: Some asshole in a red Jeep Wrangler rear ended me and moved my car around at a fucking one way STOP cause h
e couldn’t wait 2 m…
the id of user tweet is 1114831477995397120
```

Fig. 8. Fetching Queries

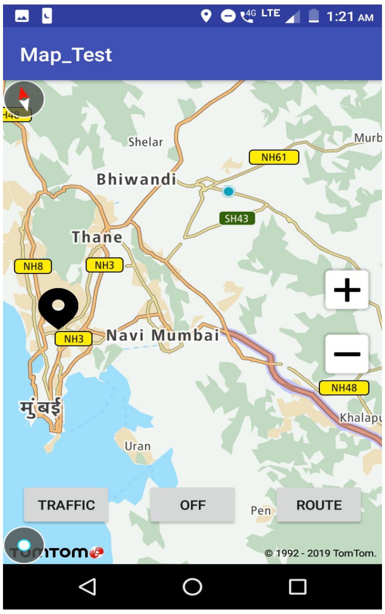
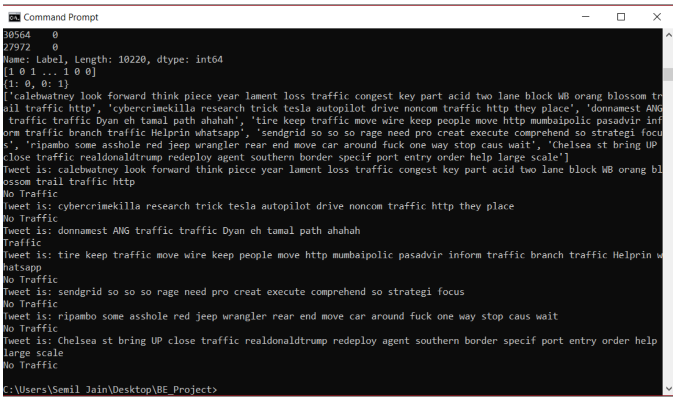


Fig. 9. Android Display



```
30564    0
27972    0
Name: Label, Length: 10220, dtype: int64
[1 0 1 ... 1 0 0]
{1: 0, 0: 1}
['calebwatney look forward think piece year lament loss traffic congest key part acid two lane block WB orang blossom tr
ail traffic http', 'cybercrimekilla research trick tesla autopilot drive noncom traffic http they place', 'donnamest ANG
 traffic traffic Dyan eh tamal path ahahah', 'tire keep traffic move wire keep people move http mumbaipolic pasadvir inf
orm traffic branch traffic Helprin whatsapp', 'sendgrid so so so rage need pro creat execute comprehend so strategi focu
s', 'ripambo some asshole red jeep wrangler rear end move car around fuck one way stop caus wait', 'Chelsea st bring UP
close traffic realdonaldtrump redeploy agent southern border specif port entry order help large scale']
Tweet is: calebwatney look forward think piece year lament loss traffic congest key part acid two lane block WB orang bl
ossom trail traffic http
No Traffic
Tweet is: cybercrimekilla research trick tesla autopilot drive noncom traffic http they place
No Traffic
Tweet is: donnamest ANG traffic traffic Dyan eh tamal path ahahah
Traffic
Tweet is: tire keep traffic move wire keep people move http mumbaipolic pasadvir inform traffic branch traffic Helprin w
hatsapp
No Traffic
Tweet is: sendgrid so so so rage need pro creat execute comprehend so strategi focus
No Traffic
Tweet is: ripambo some asshole red jeep wrangler rear end move car around fuck one way stop caus wait
No Traffic
Tweet is: Chelsea st bring UP close traffic realdonaldtrump redeploy agent southern border specif port entry order help
large scale
No Traffic

C:\Users\Semil Jain\Desktop\BE_Project>
```

Fig. 10. Classification of tweets

6.2 Evaluation

For evaluating precision and recall, the algorithm was run for a number of queries and then calculate the correct number of classification computed. In order to calculate this, we first need to determine the parameters or types of errors. True Positive (TP) = Predicted traffic class and Actual traffic analyzed are both Yes. False Positive (FP) = Predicted traffic class is Yes and Actual traffic analyzed is No. False Negative (FN) = Predicted traffic class is No and Actual traffic analyzed is Yes. True Negative (TN) = Predicted traffic class and Actual traffic analyzed are both No. Precision = (TP / (TP + FP)) and Recall = (TP / (TP+FN)) We run the system ten times. In our case, TP=7, FP=3, FN=5, TN=5. Based on this Precision and Recall percentage is calculated. We found out the precision to be 70%. Recall is calculated to be 58%. The Accuracy of the training algorithm was found to be 94% approximately. A colored route is plotted from the source and destination mentioned by the user, Green indicates no traffic whereas Grey indicates traffic.

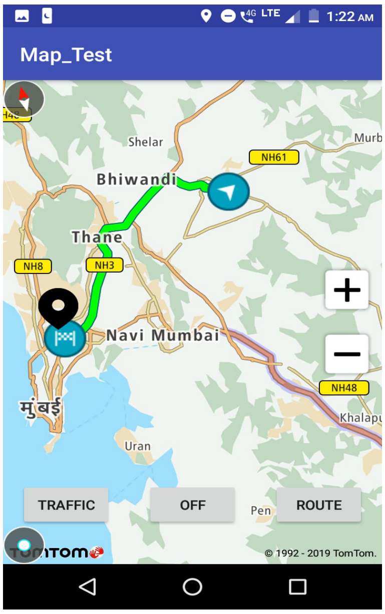


Fig. 11. Traffic Indication

7. CONCLUSION

In this project, A system for real-time detection of traffic-related events from Twitter stream analysis was developed. The developed system, built using the proposed methodologies, fetches and classifies streams of tweets and tells the users whether traffic is present or not. The input queries are pre-processed and converted into tokens after removing noise. These tokens are further classified using Naïve Bayes multinomial classifier,1 for traffic and 0 for no traffic. This is then used to plot colored routes on an open-source map between source and destination. We have examined and used available software libraries and the latest methodologies for text analysis and pattern classification. In order to build the overall system, All these technologies and techniques mentioned have been thoroughly investigated, adapted, and integrated.

8. REFERENCES

[1] Eleonora D'Andrea, Pietro Ducange, Beatrice Lazzerini, Member, IEEE, and Francesco Marcelloni, Member, IEEE, "Real-Time Detection of Traffic From Twitter Stream Analysis", https://ieeexplore.ieee.org/document/7057672

[2] Fatma Amin Elsafoury Enschede, The Netherlands, "Monitoring Urban Traffic Status Using Twitter Messages " , https://ieeexplore.ieee.org/elsafoury_28376

[3] Freddy Lécué, Robert Tucker, Veli Bicer, Pierpaolo Tommasi, Simone TalleviDiotallevi, Marco Sbodio, "Predicting Severity of Road Traffic Congestion using Semantic Web Technologies", https://ieeexplore.ieee.org/document/paper_205

[4] Farman Ali, Daehan Kwak, Pervez Khan, S. M. Riazul Islam, Kye Hyun Kim, K.S. Kwak, "Fuzzy Ontology-based Sentiment Analysis of Transportation and City Feature Reviews for Safe Traveling", https://ieeexplore.ieee.org/document/1701.05334

[5] Fellbaum, Christiane, 'WordNet — A Lexical Database for English', 2005. [Online]. Available: https://wordnet.princeton.edu/. [Accessed: 21- Oct- 2018].