

DETECTING TRAFFIC INFORMATION FROM SOCIAL MEDIA TEXTS USING DEEP LEARNING

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

**JUPALLI MANIBHASKAR(37110427)
KONDURI SRIKANTH(37110360)**



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC JEPPIAAR NAGAR,
RAJIV GANDHI SALAI, CHENNAI – 600 119**

MARCH 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with —All grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafidework of **JUPALLI MANIBHASKAR (37110427)** who carried out the project entitled **“DETECTING TRAFFIC INFORMATION FROM SOCIAL MEDIA TEXTS USING DEEP LEARNING”** under my supervision from **December 2020** to **March 2021**.

Internal Guide

Dr. A.ANKAYARKANNI , M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **JUPALLI MANIBHASKAR (37110427)** hereby declare that the Project Report entitled **-DETECTING TRAFFIC INFORMATION FROM SOCIAL MEDIA TEXT USING DEEP LEARNING”** done by me under the guidance of **DR.A.ANKAYARKANNI, M.E. , Ph.D.** at Sathyabama Institute of Science and technology is submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering degree in Computer Science and Engineering

DATE:

PLACE:CHENNAI

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean of the Department, Department of Computer Science and Engineering** for providing us the necessary support and details at the right time during the progressive reviews.

We convey our thanks to **Dr. S. VIGNESHWARI M.E., Ph.D., and Dr. L.Lakshmanan M.E., Ph.D. Head of the Departments, Department of Computer Science and Engineering** for providing us the necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.A.ANKAYARKANNI, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTRACT

Social media, such as Twitter, Facebook, YouTube, Instagram, Tumblr etc is tremendously the go-to platform for emerging internet and mobile users to share user generated comments or experiences about a variety of traffic information and other forms of public expressions on traffic. In this domain, Deep Learning Techniques will be used for public sentiment analysis on traffic information through social media.

In the existing system, the traffic information can be determined using Google Map or in FM Radio. Google Map will not give you the accurate Traffic Information. In Addition, it will not provide any information about the traffic events. FM Radio will not give you the information 24/7, In addition, it will provide the information only within the city or their coverage.

TABLE OF CONTENTS

ABSTRACT
LIST OF FIGURES

v
viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	DOMAIN OVERVIEW	1
	LIBRARIES	2
	DEEP LEARNING	7
2.	LITERATURE SURVEY	12
	LITERATURE SURVEY 1	12
	LITERATURE SURVEY 2	14
	LITERATURE SURVEY 3	16
	LITERATURE SURVEY 4	18
3.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	21
	AIM OF PROJECT	21
	SCOPE OF PROJECT	22
	ADVANTAGES OF PROPOSED SYSTEM	22
4.	EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS	
	METHODOLOGY	23
	LIST OF MODULES	30
	DATASET PREPARATION	30
	FEATURE ENGINEERING	31
	TRANSFORM TO TRAFFIC INFORMATION	31
	ARCHITECTURE	32
	REQUIREMENTS	33
	HARDWARE REQUIREMENTS	33
	SOFTWARE REQUIREMENTS	33
	PACKAGE LIST	33
1.	RESULTS AND DISCUSSION	34

2.	SUMMARY AND CONCLUSION	38
	REFERENCES	39
	APPENDIX	42
	A.PLAGARISMREPORT	42
	B.PAPER	43
	C.SOURCECODE	46

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.3.1	DEEP LEARNING	8
1.3.2	NEURAL NETWORKS	9
4.1.1	METHODOLOGY	23
4.1.2	LAYERS	25
4.1.3	HUFFMAN TREE	26
4.1.4	CNN ARCHITECTURE	28
4.1.5	LSTM ARCHITECTURE	29
4.1.6	LSTM	29
4.2.1	MODULES	30
4.3.1	ARCHITECTURE	32

CHAPTER 1

INTRODUCTION

Social media has increased noteworthy prevalence and significance during the last decade and has turned into a fundamental piece of numerous individuals' regular day to day existences. As social media clients expound on an expansive scope of subjects, famous social networking sites can fill in as an ideal base for different information mining and data extraction applications. One plausibility among these could be the continuous recognition of startling traffic events or abnormalities, which could be utilized to help traffic administrators to find and alleviate hazardous spots in a convenient way or to help travellers with making educated choices about their travel route.

Traffic mishaps upset the traffic tasks, separate the traffic stream, and cause serious urban issues around the world. Major traffic mishaps can now and again lead to hopeless harms, wounds, and even fatalities. With the enormous increment in the utilization of social system sites like Twitter and Facebook, online network is trading data as feelings, suppositions, feelings, and aims, which mirror their affiliations and inclination towards a substance, event and strategy. Existing methodologies for the most part apply classic machine learning techniques dependent on verifiable mishap records. In this way they fizzled to include the cross-domain information, which contains spatial and temporal dependency. In the existing system, users collect traffic data from various sources, save it, analyse it by applying various algorithms, text analysis, visualize the traffic data, and publish a summary report and visualization.

The existing system can be utilized to necessarily produce reports from a variety of data sources, using analysis to cluster and identify key groups, people and content in the traffic network. Authorities can interact with visualizations, searching, filtering or sorting to reveal insights in the data. The underlying data is directly accessible via the spreadsheet.

DOMAIN OVERVIEW

Deep learning has evolved hand-in-hand with the digital era, which has brought about an

explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

LIBRARIES

PANDAS:

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandas' DataFrames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and DataFrames.

1. Series: a pandas Series is a one dimensional data structure (–a one dimensional ndarray) that can store values — and for every value it holds a unique index, too.
2. DataFrame: a pandas DataFrame is a two (or more) dimensional data structure — basically a table with rows and columns. The columns have names and the rows have indexes.

Those who are familiar with R know the data frame as a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frame's rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc.

Now, DataFrames in Python are very similar: they come with the Pandas library, and they are defined as two-dimensional labeled data structures with columns of potentially different types. In general, you could say that the Pandas DataFrame consists of three main components: the data, the index, and the columns.

Firstly, the DataFrame can contain data that is:

- a Pandas DataFrame
- a Pandas Series: a one-dimensional labeled array capable of holding any data type with axis labels or index. An example of a Series object is one column from a DataFrame.
- a NumPy ndarray, which can be a record or structured
- a two-dimensional ndarray
- dictionaries of one-dimensional ndarray's, lists, dictionaries or Series.

Some of the key features of Python Pandas are as follows:

It provides DataFrame objects with default and customized indexing which is very fast and efficient.

- There are tools available for loading data of different file formats into in-memory data objects.
- It is easy to perform data alignment and integrated handling of missing data in Python Pandas.
- It is very simple to perform pivoting and reshaping of data sets in Pandas.
- It also provides indexing, label-based slicing, and sub-setting of large data sets.
- We can easily insert and delete columns from a data structure.
- Data aggregation and transformations can be done using group by.
- High-performance merging and joining of data can be done using Pandas.
- It also provides time series functionality.
- Inserting and deleting columns in data structures.
- Merging and joining data sets.
- Reshaping and pivoting data sets.
- Aligning data and dealing with missing data.
- Manipulating data using integrated indexing for DataFrame objects.
- Performing split-apply-combine on data sets using the group by engine.
- Manipulating high-dimensional data in a data structure with a lower dimension using hierarchical axis indexing.
- Subsetting, fancy indexing, and label-based slicing data sets that are large in size.
- Generating data range, converting frequency, date shifting, lagging, and other time-series functionality.
- Reading from files with CSV, XLSX, TXT, among other formats.

- Arranging data in an order ascending or descending.
- Filtering data around a condition.
- Analyzing time series.
- Iterating over a data set.

NUMPY

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just can't miss when you're learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices. To work with these arrays, there's a vast amount of high-level mathematical functions operate on these matrices and arrays.

an array is basically nothing but pointers. It's a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and

- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next row.

Or, in other words, an array contains information about the raw data, how to locate an element and how to interpret an element.

With NumPy, we work with multidimensional arrays. We'll dive into all of the possible types of multidimensional arrays later on, but for now, we'll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with. In fact, it's just a different way of thinking about a list of lists. A matrix has rows and columns. By specifying a row number and a column number, we're able to extract an element from a matrix.

We can create a NumPy array using the `numpy.array` function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns.

Because we want all of the elements in the array to be float elements for easy computation, we'll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats. NumPy has several advantages over using core Python mathematical functions, a few of which are outlined here:

1. NumPy is extremely fast when compared to core Python thanks to its heavy use of C extensions.
2. Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library. Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.

NumPy comes with a variety of built-in functionalities, which in core Python would take a fair bit of custom code.

MATPLOTLIB

Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of

data visualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams.

Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well.

Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB, if it is used in combination with Numpy and Scipy. Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB. This can make it extremely easy for MATLAB users to migrate to matplotlib.

Matplotlib can be used to create publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Another characteristic of matplotlib is its steep learning curve, which means that users usually make rapid progress after having started. The official website has to say the following about this: "matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code."

DEEP LEARNING

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural

networks.

Deep learning algorithms are constructed with connected layers.

- The first layer is called the Input Layer
- The last layer is called the Output Layer
- All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.

A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.

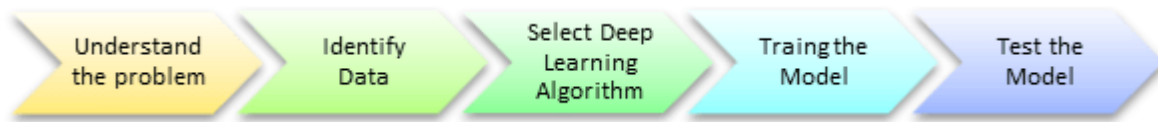


Fig:1.3.1 Deep learning

To grasp the idea of deep learning, imagine a family, with an infant and parents. The toddler points objects with his little finger and always says the word 'cat.' As its parents are concerned about his education, they keep telling him 'Yes, that is a cat' or 'No, that is not a cat.' The infant persists in pointing objects but becomes more accurate with 'cats.' The little kid, deep down, does not know why he can say it is a cat or not. He has just learned how to hierarchies complex features coming up with a cat by looking at the pet overall and continue to focus on details such as the tails or the nose before to make up his mind.

A neural network works quite the same. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers.

The learning occurs in two phases.

1. The first phase consists of applying a nonlinear transformation of the input and create a statistical model as output.
2. The second phase aims at improving the model with a mathematical method known as derivative.

The neural network repeats these two phases hundreds to thousands of time until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration

Classification of Neural Networks

1. Shallow neural network: The Shallow neural network has only one hidden layer between the input and output.
2. Deep neural network: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers.

The computational models in Deep Learning are loosely inspired by the human brain. The multiple layers of training are called Artificial Neural Networks (ANN).

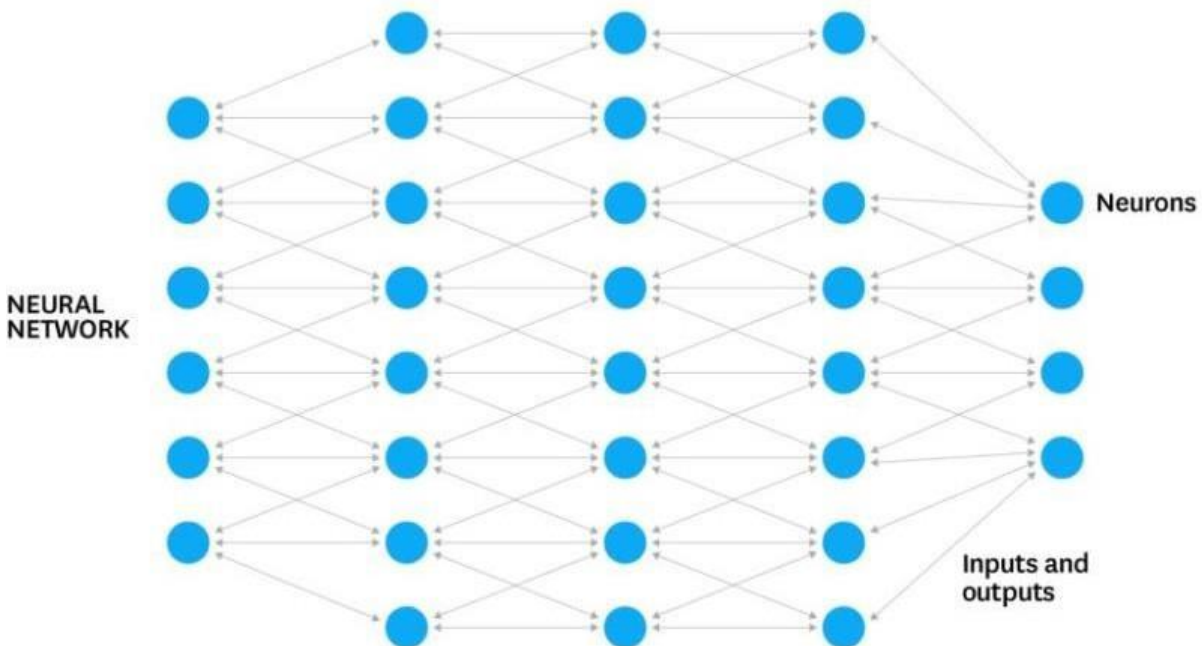


Fig:1.3.2 Neural Network

Neuron

Artificial Neural Networks contain layers of neurons. A neuron is a computational unit that calculates a piece of information based on weighted input parameters. Inputs accepted by the neuron are separately weighted.

Inputs are summed and passed through a non-linear function to produce output. Each layer of neurons detects some additional information, such as edges of things in a picture or tumors in a human body. Multiple layers of neurons can be used to detect additional information about input parameters.

Nodes

Artificial Neural Network is an interconnected group of nodes akin to the vast network of layers of neurons in a brain. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

Inputs

Inputs are passed into the first layer. Individual neurons receive the inputs, with each of them receiving a specific value. After this, an output is produced based on these values.

Outputs

The outputs from the first layer are then passed into the second layer to be processed. This continues until the final output is produced. The assumption is that the correct output is predefined.

Each time data is passed through the network, the end result is compared with the correct one, and tweaks are made to their values until the network creates the correct final output each time.

Some of the commonly used neural networks are as follows:

1. Artificial Neural Network (ANN)
2. Convolutional Neural Network (CNN)
3. Recurrent Neural Network (RNN)
4. Deep Neural Network (DNN)
5. Deep Belief Network (DBN)

Artificial neural networks are one of the main tools used in machine learning. As the —neurall part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

Generally, the working of a human brain by making the right connections is the idea behind ANNs. That was limited to use of silicon and wires as living neurons and dendrites.

Here, neurons, part of human brain. That was composed of 86 billion nerve cells. Also,

connected to other thousands of cells by Axons. Although, there are various inputs from sensory organs. That was accepted by dendrites. As a result, it creates electric impulses. That is used to travel through the Artificial neural network. Thus, to handle the different issues, neuron send a message to another neuron.

As a result, we can say that ANNs are composed of multiple nodes. That imitate biological neurons of the human brain. Although, we connect these neurons by links. Also, they interact with each other. Although, nodes are used to take input data. Further, perform simple operations on the data. As a result, these operations are passed to other neurons. Also, output at each node is called its activation or node value.

CHAPTER 2

LITERATURE SURVEY

A

Literature Survey 1	
Title	A Novel Acoustic Characteristic Extraction Algorithm for Traffic Volume Estimation
Authors	ZHENG ZOU, QINGLU MA, FENGJIE LIU,, AND SALEEM ULLAH
Published Year	2020
Efficiency	✓ Problems remain unresolved under certain complex conditions. ✓ Does not decreases the accuracy of endpointdetection. ✓ Ability to maintain satisfactory traffic volumeestimation
Drawbacks	↓ Does not support human-centriccomputing. ↓ Does not identify highly influential users in microblogging networks ↓ Does not enhance the effectiveness of hot-event dissemination
Description	Inthispaper,acousticdatasetsacquiredfromacousticsensorsare utilized to estimate the road traffic volume. The short-term energy (STE) algorithm and the energy to zero crossing rate (EZCR) algorithm are usually applied to acoustic analysis, and they both perform well under some simple circumstances, however, some urgent problems remain unresolved under certain complex conditions. One of such issues occurs when adjacent vehicle-pass signals(VPSs)intersectpartially,seriouslydecreasingtheaccur

acy of endpoint detection of VPSs, hampering the algorithm ability to maintain satisfactory traffic volume estimation. Another difficulty arises while multiple lanes are considered: some special VPSs cannot be detected. To solve these problems, a novel acoustic characteristic is defined, and an acoustic-based characteristic extraction algorithm for traffic volume estimation, entitled triangular wave analysis (TWA), is proposed. Comparing the TWA algorithm with the STE and EZCR algorithms, experimental results demonstrate the viability of the proposed algorithm in the case of intersectant VPSs.

The biggest limitations of conventional algorithms are that when some valley values are larger than some peak values, we cannot detect out the intersectant VPSs and ordinary VPS at the same time.

But if all valley values are equal to zero and the peak values tend to be larger, then the parameters e_1 , e_2 , c_1 , and c_2 can be simply set to zero, meanwhile the limitations will be suppressed well enough. So, according to this idea, in this section, we present a detailed description of the proposed algorithm, entitled Triangular Wave

Analysis (TWA), to improve the accuracies of traffic volume and endpoint detections. The Triangular Wave Analysis (TWA) algorithm can be divided into three steps: extremum extraction, triangular wave formation and triangular wave combinations. Finally, one triangular wave represents one VPS, and the endpoint of triangular wave namely is the endpoint of VPS.

After the formation of triangular wave, some triangular waves need to be combined into one triangular wave according to the Triangular

	ar wave combinations rules. Triangular wave combinations can be divided into four steps: up combination, down combination, frame length extension, and trafficstatistics.
--	---

B

Literature Survey 2	
Title	Traffic-Optimized Data Placement for Social Media
Authors	Jing Tang, Member, IEEE, Xueyan Tang, Member, IEEE, and Junsong Yuan
Published Year	2017
Efficiency	✓ Continuously adapt data placement according to various dynamics ✓ TOPR not only reduces the inter-server traffic significantly butalsosavesmuchstoragecostofreplicationcomparedto state-of-the-artmethods. ✓ TOPR also tunes the replicas under new partitioning to further reduce inter servercommunication
Drawbacks	↓ It is hard to discover users' potential interests to new contents. ↓ Does not improve the efficiency of the hot-event detection andpropagation. ↓ Does not identify high quality subsets of users, topics, and posts.
Description	This paper investigates how to minimize the inter-server traffic among a cluster of social media servers through joint data partitioning and replication optimization. We formally define the

	problem and study its hardness. We then propose a Traffic-Optimized Partitioning and Replication (TOPR) method to continuously adapt data placement according to various dynamics. Evaluations with real Twitter and LiveJournal social graphs show that TOPR not only reduces the inter-server traffic significantly but also saves much storage cost of replication compared to state-of-the-art methods. We also benchmark TOPR against the offline optimum by a binary linear program. We do not explicitly restrict the number of slave replicas in our problem definition. It is intuitive that creating too many slave replicas (e.g., full replication) would produce a huge amount of write-incurred traffic and result in high inter-server traffic. Thus, even if there is no capacity limit enforced on slave replicas, it would not lead to the creation of an excessive and unrealistic number of slave replicas. The read operation does not cause any inter-server communication, but it still induces request processing at both servers. On the other hand, a write operation has to produce inter-server communication for synchronizing replicas and also needs to be processed by multiple servers. Our problem formulation can be adapted to minimize the total number of requests processed by all servers in the redirect model. Note that the initial requests sent by users to their master servers are beyond the control of the distributed data storage system. Therefore, minimizing the total number of requests processed by all servers is equivalent to minimizing the total number of redirected read requests and inter-server
--	---

	synchronization requests. TOPR dynamically adjusts the partitioning and replication to optimize the inter-server traffic. Specifically, TOPR adjusts the allocation of master and slave replicas together based on the estimate of the read and write rates of the users. To reduce computational overheads, the adjustments are made when these rates change beyond some given thresholds
--	--

2.2)C

Literature Survey 3	
Title	Group Event Recommendation in Event-Based Social Networks Considering Unexperienced Events
Authors	GUOQIONGLIAO, XIAOMEI HUANG MINGSONG MAO, CHANGXUAN WAN, XIPING LIU, AND DEXILIU
Published Year	,2019
Efficiency	✓ Appropriate topic areas for information can drastically improve the disseminationeffect. ✓ A high-influence greedy strategy is adopted to form an alternative set of keyusers. ✓ Optimization based on sub modularity isused.
Drawbacks	↓ The complexity of computing betweenness centrality is very high. ↓ Cannot be directly analysed by the application of these centrality methodsalone. ↓ Collaborative filtering will suffer from the cold-startproblem

Description	This paper proposes a two-phase group event recommendation (2PGER) model for EBSNs. Firstly, we leverage information such as online social behaviors, users' event participation records, and topological structures of EBSNs to establish a global trust network among users and establish ego trust networks of all users. Then, we perform random walks on the pre-built ego trust network for each user to acquire the user's predicted preferences on the unexperienced events. Thirdly, we adopt a random walk with restarts (RWR) method to aggregate users' preferences and recommend top N events to groups. In the end, we compare 2PGER with several baseline approaches on real datasets from Meetup. The results show that 2PGER outperforms baselines. Predict the initial preferences of group members. According to the theme description, time, and location of events, the feature vectors of events are extracted, and the members' initial preferences for newly published events are obtained by using a content-based method, which matching the features of members' historical events against those of the new events. Based on above initial preferences, we determine whether the new events are unexperienced events. If they are unexperienced events, the process of asking for advice from friends outside the group will be performed, otherwise the process will be skipped. Consult with friends outside the group. The specific process is as follows: Firstly, according to the interaction information among users, events and social groups, combined with the topology of heterogeneous networks, the value of
-------------	--

	personality and trust are calculated, and the global trust network is constructed. Secondly, walks are performed randomly on each member's egotrust network to simulate the process of consulting with friends. Finally, members' preferences for unexperienced events are obtained based on their friends' advice.
--	--

2.4)D

Literature Survey 4	
Title	Human-Centric Cyber Social Computing Model for Hot-Event Detection and Propagation
Authors	Lei-Lei Shi, Lu Liu , Member, IEEE, Yan Wu, Liang Jiang , Muhammad Kazim , Haider Ali, and John Panneerselvam
Published Year	20 19

Efficiency	✓ Establish a global trust network among users and establish egotrust networks of all users ✓ User's predicted preferences on the unexperienced events ✓ Outperforms baselines
Drawbacks	↓
Description	This paper proposes a human-centric social computing (HCSC) model for hot-event detection and propagation in microblogging networks. In the proposed HCSC model, all posts and users are preprocessed through hypertext induced topic search (HITS) for determining high-quality subsets of the users, topics, and posts. Then, a latent Dirichlet allocation (LDA)-based multiprototype user topic detection method is used for identifying users with high influence in the network. Furthermore, an influence maximization is used for final determination of influential users based on the user subsets. Finally, the users mined by influence maximization process are generated as the influential user sets for specific topics. Experimental results prove the superiority of our HCSC model against similar models of hot-event detection and information propagation. The MP-LDA model is composed of two modules. After identifying the topic scores using the TD-HITS method, the topics are clustered by using the LDA topic model in the post network and then the Gibbs sampling method is used for user topics detection. Then, the initial influential spreaders in influence maximization are discovered based on their hub value in the microblogging network and their local features in the user-user network using TS-LDA.

	In the influence maximization, the selection of the initial propagator on users depends on the information propagation trend and the sphere of influence. High-quality posts can draw more attention from high-influence users spreading and/or broadcasting such posts in the microblogging network. Developing the ability to identify influential spreaders effectively and efficiently in the microblogging network is a major challenge. We propose a novel solution to this problem. We believe that the initial influential spreader should satisfy a global condition and a network topology condition within a certain period, i.e., a high global feature and a high local feature. Here, we expect that high local feature users will trigger an early and rapid accumulation of contagious transmissions among a large number of candidate users. Finally, the global importance of users can be acquired by the hub value of users along with their posts. Subsequently, high global feature users will spread information, ideas, or rumors much faster than ordinary users.
--	--

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

AIM OF PROJECT

The Aim of project is to extract and analyse the real-time traffic information from the social media sensor. The social media text messages can be used to analysing the traffic events with respect to

- ▶ Where the traffic event happened?
- ▶ When the traffic event happened?
- ▶ Why the traffic event happened?

The proposed system provides a unified end-to-end timeline visualization dashboard which empowers the authorities to clearer recognize of traffic incidents from social media sensor.

In this intended application, the risk factors classification and influence assessment are the main goal for the traffic safety management.

It is a gold dig for the individuals who welcome the benefit of understanding public sentiment.

In spite of the fact that there are numerous approaches to classify applications traffic, deep learning based methodologies are ending up progressively effective in these days. This is on the grounds that machine learning strategies are more proper than existing strategies for precise what's more, proficient applications traffic arrangement. Deep learning strategy can be arrangement of this problem. The best favourable position of machine learning can adapt to various circumstance regardless of whether another circumstance happens unexpectedly.

The proposed procedure includes numerous collaborative steps, such as data pre-processing, feature extraction using business experience, feature selection using weight of evidence

and information value, classification, as well as model validation and result interpretation.

SCOPE OF PROJECT

The main contributions of this project therefore are:

- ▶ Data Analysis
- ▶ Dataset Preprocessing
- ▶ Training the Model
- ▶ Testing of Dataset

ADVANTAGES OF PROPOSED SYSTEM

- ✓ Easy Access
- ✓ Universal coverage
- ✓ Free Building Cost
- ✓ Free Maintenance cost
- ✓ Good performance in the area of Pattern Recognition and Regression Estimation

As an enhancement from the proposed system, our application will extract and analyse the below information like

- ▶ Traffic Management Information
- ▶ Time Series Visualization
- ▶ Accident Distribution Characteristics
- ▶ Identify various factors like
 - Human Factor
 - Vehicle Factor
 - Road Factor
 - Distraction Factor
 - Risk-taking behaviours such as speed racing and overloaded
 - Environment Factor
 - And any other factors

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS;ALGORITHMS

4.1 METHODOLOGY

The methodological framework of this approach mainly includes

A. Data acquisition by crawling Sina Weibo.

B. Word segmentation.

C. Word embedding.

D. Classification of microblogs to extract traffic information. The whole procedure is illustrated in Fig. 1.

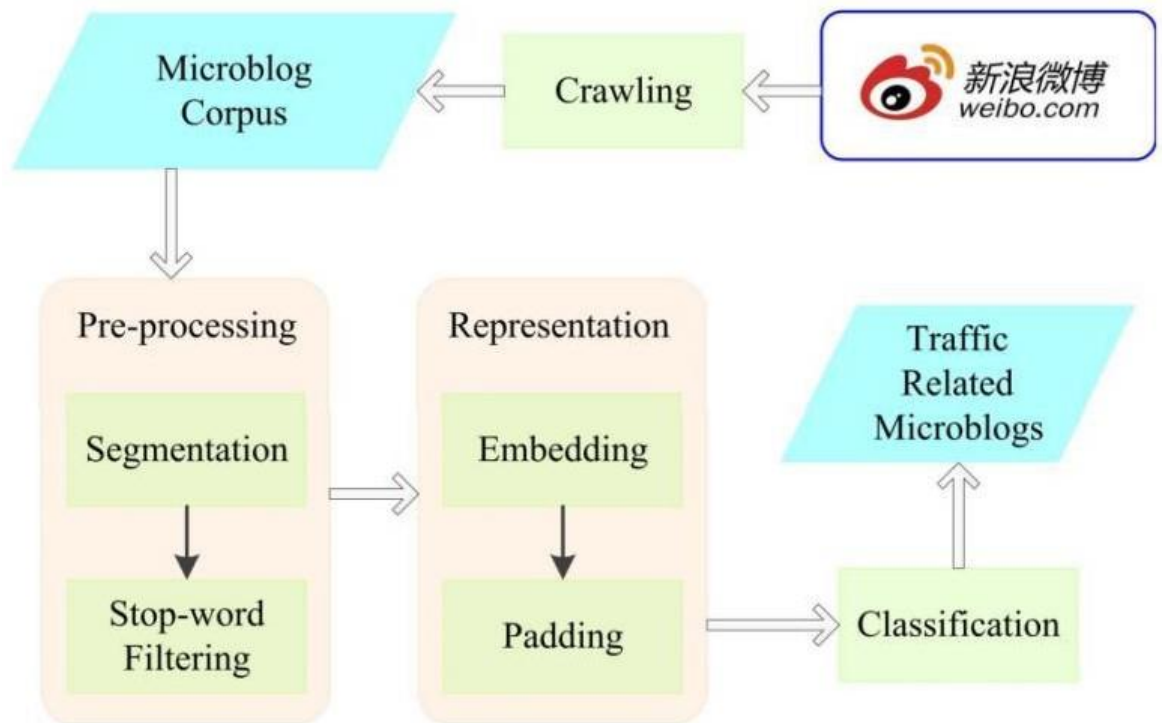


Fig:4.1.1 Methodology

A. Data Acquisition by Crawling SinaWeibo: There are two approaches, namely requesting APIs (Application Programming Interfaces) and crawling websites, to access the microblogs in Sina Weibo. Requesting APIs is officially provided for developers, but it is usually not free and has access rate limits. In this paper, we adopted the approach of crawling the Sina Weibo website. This approach is technically free but more comprehensive compared to requesting official APIs. To fetch raw microblogs, we first set search criteria (e.g., geographic coordinates, time range, keywords). Then we send HTTP requests to the server and obtain the response results which are in the format of HTML source codes. By applying regular expressions matching, we extract microblogs containing the microblog id, the user id, the timestamp, the geographic coordinates and the text from the response results. Then the structured data are stored in a database for future retrieval.

B. Word Segmentation: The space is a natural word delimiter for English and many other languages. However, there is no such separator between words in Chinese texts. Therefore word segmentation is the very first task in Chinese language processing. The accuracy of word segmentation is essential to Chinese language mining and understanding. There have been some open source tools on segmenting Chinese words [35], among which we use ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System), a well-known and widely used Chinese lexical analyzer, in this paper.

C. Word Embedding: Natural language processing systems traditionally represent each word as a one-hot vector which is a vector filled with 0s, except with 1 at the position associated with the word. The one-hot representation is very high-dimensional and sparse. Moreover, such representation cannot capture semantic similarity between words. Representing words in a continuous vector space has been proved effective in natural language processing tasks

by grouping similar words. Recently, using neural networks to get the word representations is very attractive and interesting because the learned vectors explicitly encode many linguistic regularities and patterns. The CBOW model and the Skip-Gram model are two ways of learning word embedding representations [36], [37]. These two models are similar, except that the learning objective between CBOW and Skip-Gram is different. As is illustrated in Fig. 2, the learning objective of the CBOW model is to find word vector representations that are useful for predicting the middle word under a context, while the Skip-Gram model tries to learn word vector representations by maximizing the probability of predicting surrounding words based on the middle word. As the CBOW model and the Skip-Gram model are mirror images of each other, we herein just present the derivation of the CBOW model. Due to high computational complexity of the original CBOW model and the Skip-Gram model, hierarchical softmax or (and) negative sampling are applied in the training processes.

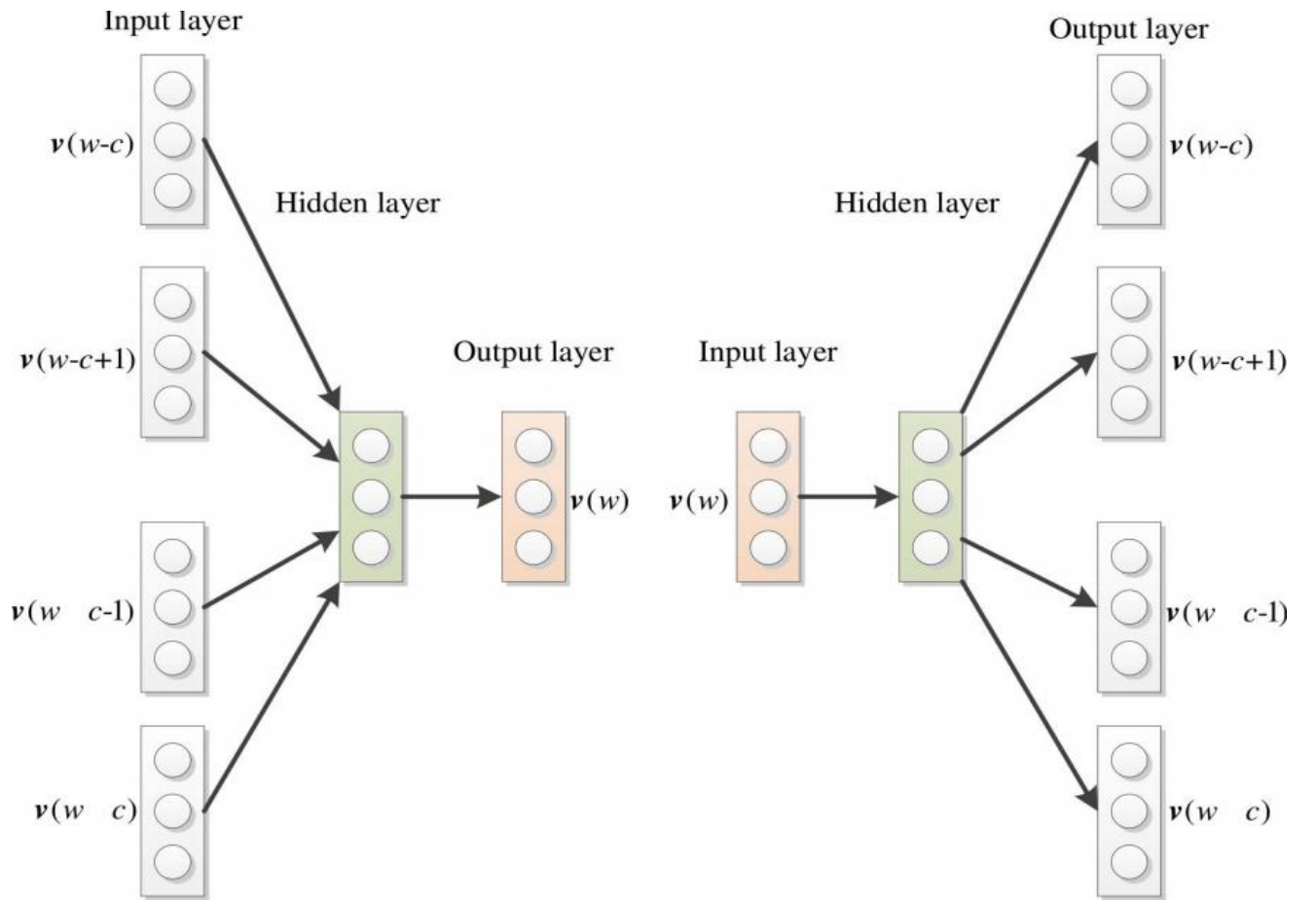


Fig:4.1.2 Layers

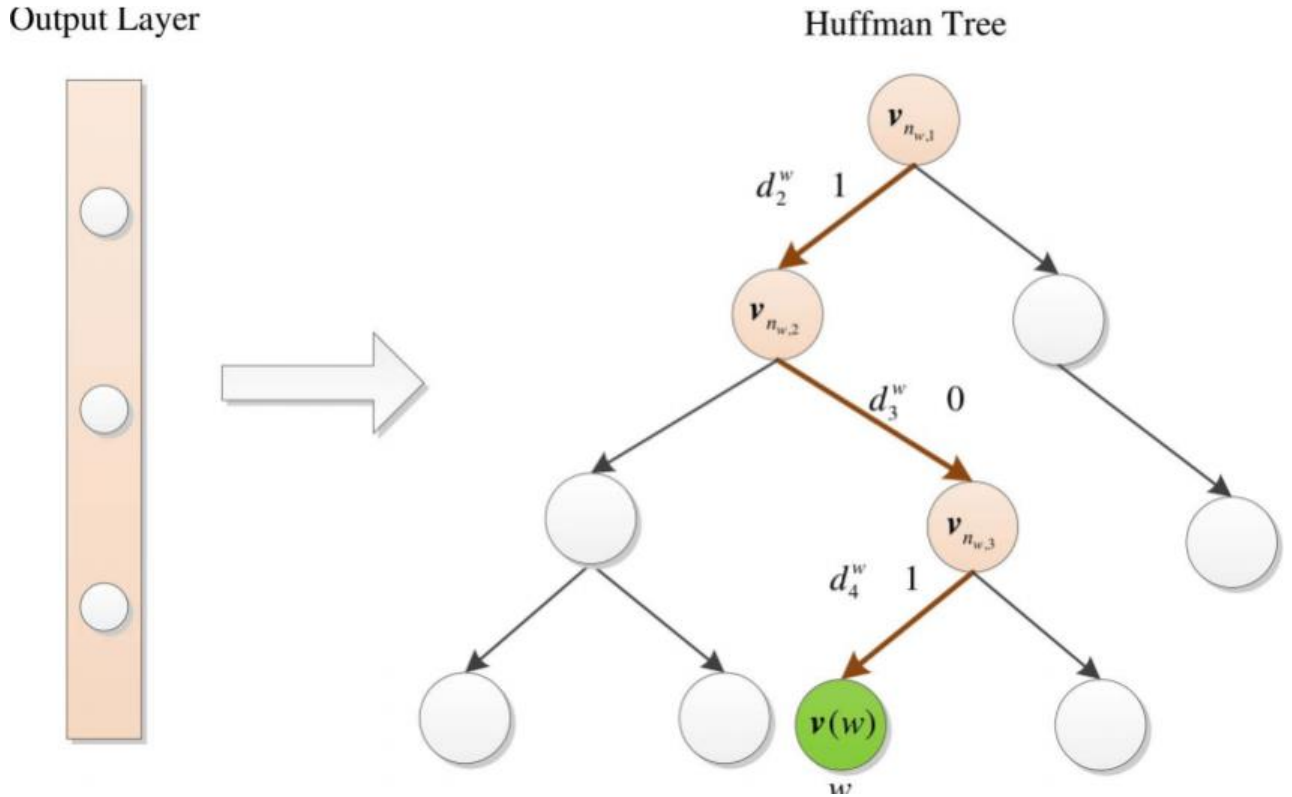


Fig:4.1.3 Huffman Tree

tree, a binary tree, to represent all the words in the vocabulary, which are leaf units. In the CBOW model with hierarchical softmax, the output layer is replaced by a Huffman Tree as shown in Fig. 3. And the hidden layer is designed to average the input word vectors, so that the output of hidden layer is

$$h = \frac{1}{C} \sum_{u \in \text{context}(w)} v(u) \quad (1)$$

where $v(u)$ represents the vector of the word u , $\text{context}(w)$ is the set of contextual words of the word w , and C is the cardinality of the set $\text{context}(w)$. Given the context, the conditional probability of the word w is defined as:

where $n_{w,j}$ is the j th inner point from the root to word w in the Huffman tree, $v_{n_{w,j}}$ is the vector of inner point n , $L(w) - 1$ is the length of the path in Huffman tree for word w , and $\sigma(x)$ is a function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

(3) where $d_{w,j}$ is the j th bit of Huffman code for word w . It is straightforward to train the neural network by maximizing the conditional probability in (2) for a target word w . Take the logarithm of the conditional probability and define the loss function:

$$L = -\log p(w|\text{context}(w))$$

(4) Then we obtain the derivative of L with regard to vector of inner point $n_{w,j}$:

(5) where $j = 1, 2, \dots, L(w) - 1$. And the derivative of L with regard to vector of contextual words u is: $\frac{\partial L}{\partial v(u)} = \sum_{j=1}^{L(w)-1} [1 - h^T v_{n_{w,j}}] v_{n_{w,j}}$

D. Classification Models In this paper, we test three types of deep learning models, i.e., CNN, LSTM, and their combination LSTM-CNN for microblog text classification. Besides, CNN, LSTM, and LSTM-CNN models have been used for traffic prediction [38]–[40].

1) CNN for Text Classification: CNN is a feedforward neural network that consists of convolutional layers interspersed with pooling layers as shown in Fig. 4 [41]–[43]. A convolutional layer aims to learn the region features. The convolution operation can be summarized as moving a filter over the sentence matrix (input map) and computing the dot products as shown in Fig. 5. Convolution with one filter outputs a feature vector. To learn more sophisticated features, there are generally distinct filters to convolve the input map, and all feature vectors are concatenated into a new matrix called feature map that would be passed to a pooling layer. For the next, we would explain the convolution operation in detail. Given a sentence $s = w_1 w_2 \dots w_N$ consisting of N words, among which word w can be

represented by V dimensional word vector, the sentence s can be transformed into a $N \times V$ dimensional matrix S called the input map. The filter is a $M \times V$ dimensional matrix K . Considering one filter p only, the i th element of the feature vector is:

$$a_{pi} = f(\text{conv}(v(w_i) \cdots v(w_{i+M-1}), K_p) + b_p)$$

(7) where K_p is filter matrix, b_p is the bias and f is the activation function. And the operation $\text{conv}(\cdot)$ is defined as: $\text{conv}(v(w_i) \cdots v(w_{i+M-1}), K_p) = \sum_{r=1}^M K_p(r, :) v^T(w_{i+r-1})$

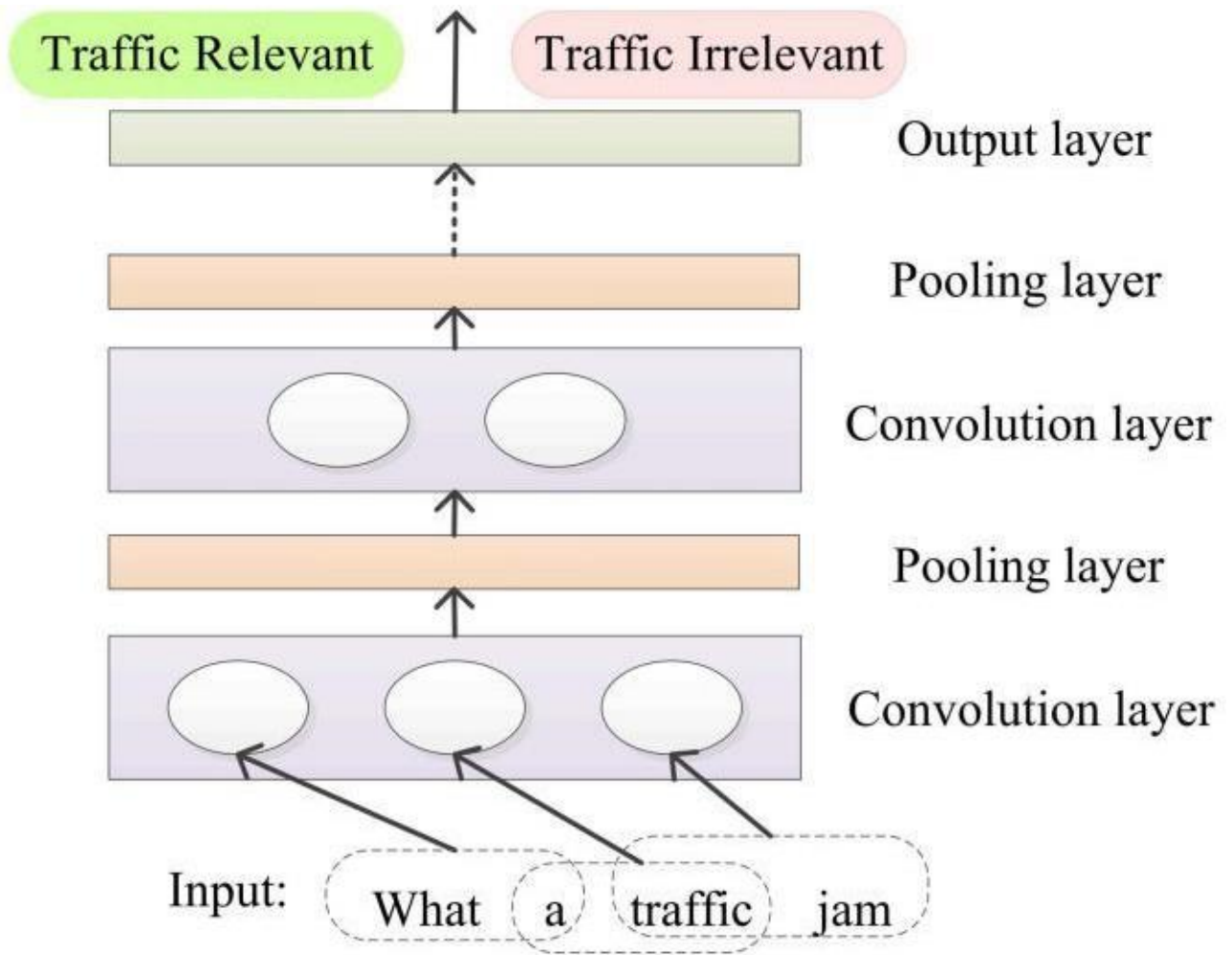


Fig:4.1.4 CNN Architecture

CNN architecture for text classification. The network consists of convolutional layers with interspersed pooling layers and output layer to give class scores.

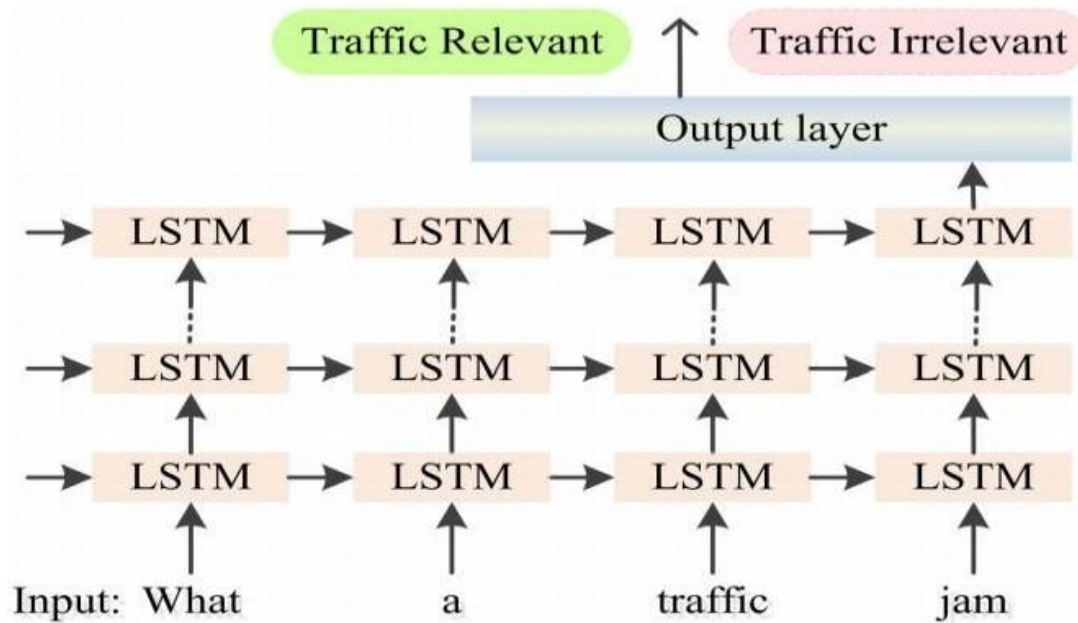


Fig:4.1.5 LSTM Architecture

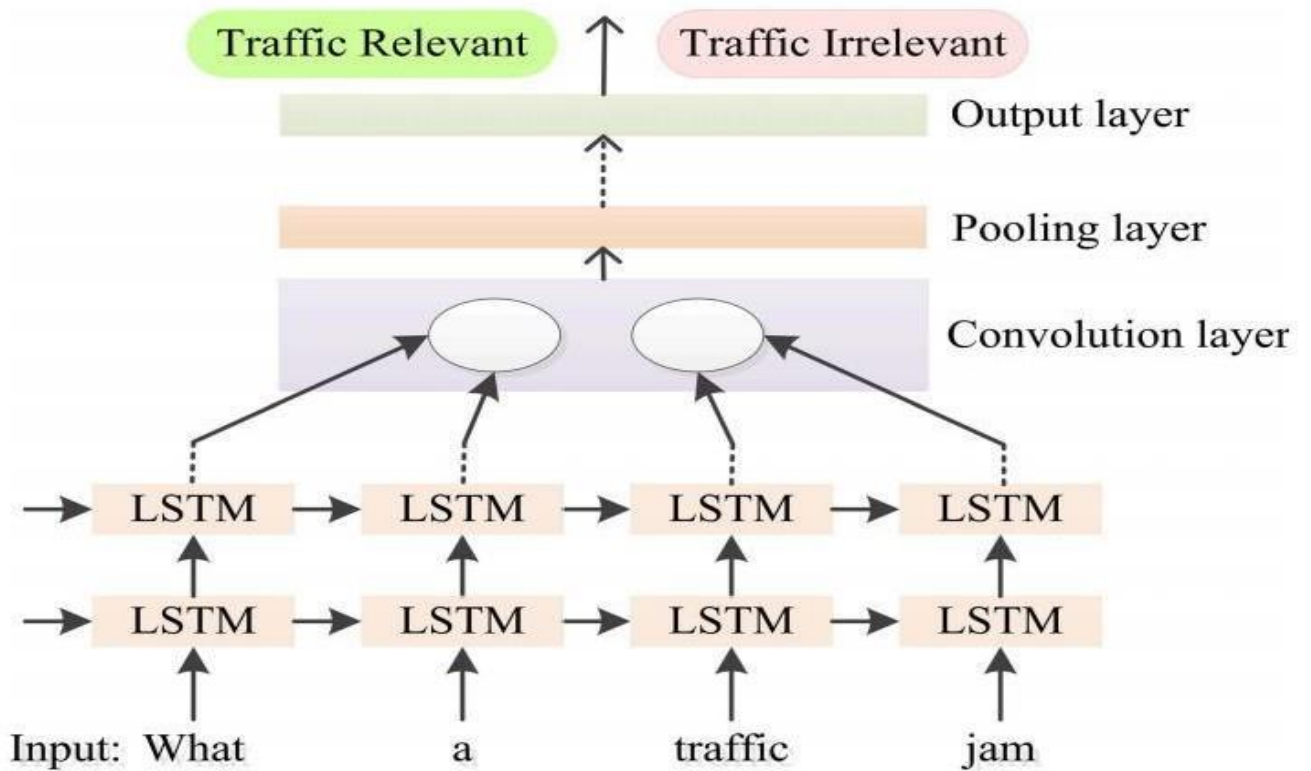


Fig:4.1.6

LSTM

14.2 LIST OF MODULES

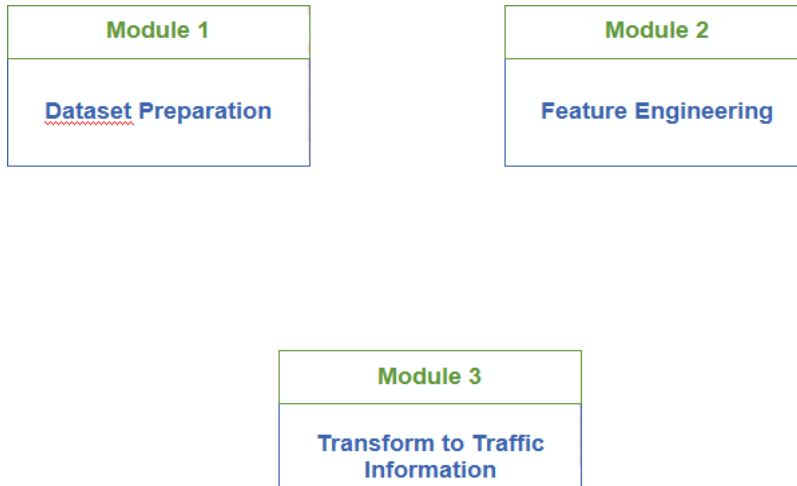


Fig:4.2.1 Modules

Module 1 : Data Preparation

Module 2 : Feature Engineering

Module 3 : Transform to Traffic Information

Dataset Preparation

Data Preparation is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data Preparation is a proven method of resolving such issues.

During the text Preparation phase, the article's content is divided into tokens and non-significant characters are removed. Named entities are recognized by the elementary method: the

first upper-case letter with corpus-based statistic is good enough for this recognition. The tokens are divided by their POS tag and generally only nouns and adjectives can be declared as potential keywords. Then the next idea is to choose only common words instead of unique ones. Keywords are often used for clustering and a keyword is useless when it is assigned to only a few articles. This is the reason why we remove tokens with low frequency and set up rules for general nouns. There is no such rule for named entities. The remaining tokens and named entities are declared as traffic information candidates.

Feature Engineering

In text processing, words of the text represent discrete, categorical features. How do we encode such data in a way which is ready to be used by the algorithms? The mapping from textual data to real valued vectors is called feature extraction.

We make the list of unique words in the text corpus called vocabulary. Then we can represent each sentence or document as a vector with each word represented as 1 for present and 0 for absent from the vocabulary. Another representation can be count the number of times each word appears in a document. The most popular approach is using the Term Frequency- Inverse Document Frequency (TF-IDF) technique.

Transform to Traffic Information

The Traffic Information extraction phase contains only the method for removing useless candidates whose score is lower than threshold value. This boundary can be changed by the number of requested automatic keywords.

The Traffic Information extraction part can be described by these steps:

- ▶ NLP method – interesting n-grams are chosen. The choice is based on their POS tag patterns and the corpus frequency is counted only for these n-grams. These n-grams can be marked as Traffic Information candidates.

- ▶ A score of importance is counted for each Traffic Information candidate. This score contains the n-gram corpus frequency and TF*IDF score for each word. The score is used for document Traffic Information selection.
- ▶ Derivation – Traffic Information candidates are merged with named entities or individual keywords if their co-occurrence is significant for this document.

ARCHITECTURE

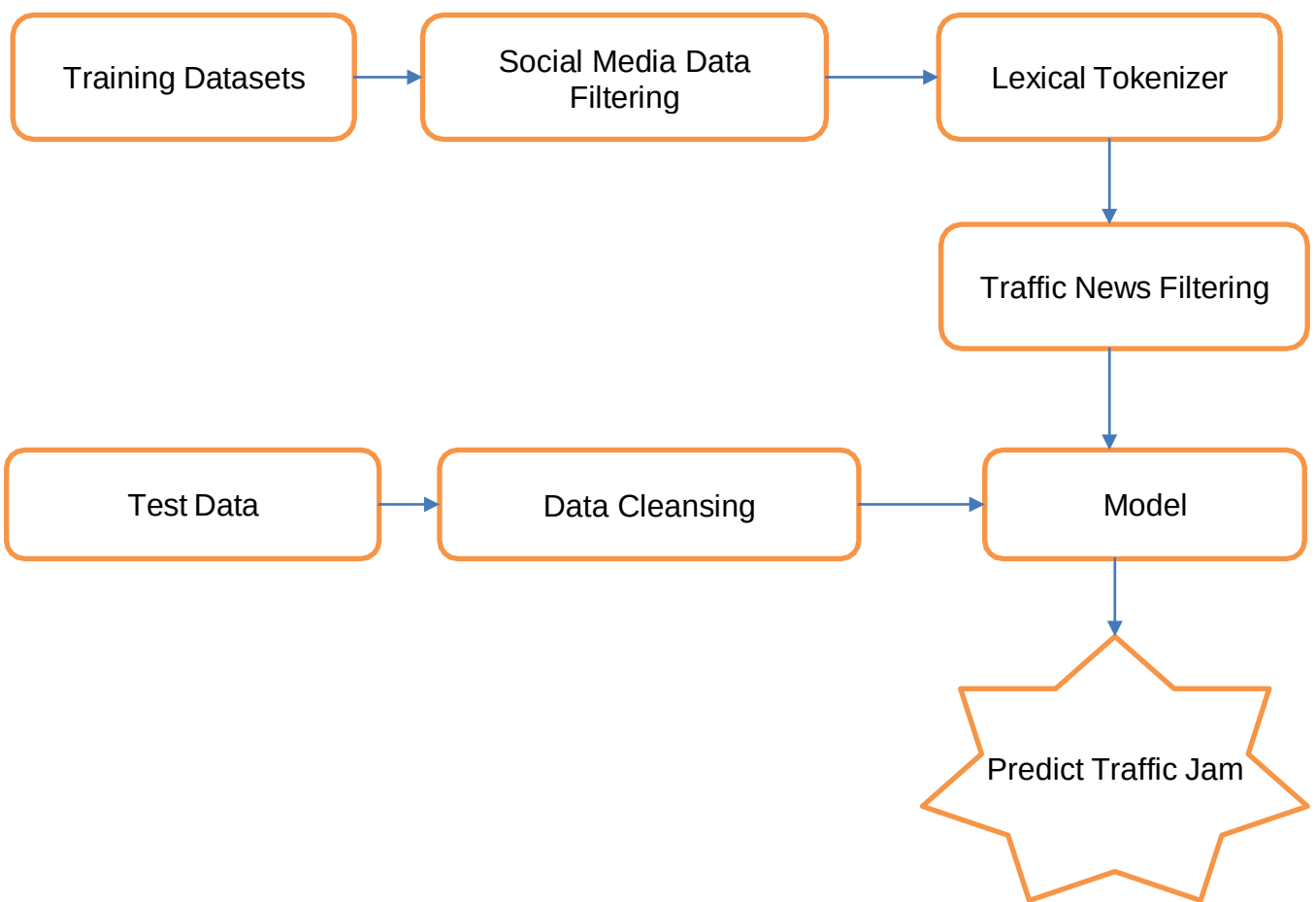


Fig:4.3.1 Architecture

REQUIREMENTS

Hardware Requirements

- Hard Disk : 250 GB and Above
- RAM : 8GB and above
- Processor : i5 and above (64 bit)

Software Requirements

Technologies

- ✓ Python
- ✓ Anaconda python

IDE

- ✓ Jupiter notebook

Packages list

pip install tensorflow==2.2

pip install keras==2.2.4

pip install pandas

pip install flask

pip install numpy

pip install image

pip install opencv-python

CHAPTER-5

RESULTS AND DISCUSSION,PERFORMANCE ANALYSIS

A.SCREEN SHOTS

```
In [21]: import tweepy
import pandas as pd
from geopy.geocoders import Nominatim
from geopy.distance import geodesic
import json

In [22]: geolocator = Nominatim(user_agent = "DSI_US_8_LA")

In [23]: # Function to take a user-entered address and return it's coordinates
def latlong(point):
    coord = geolocator.geocode(point)
    return (coord.latitude, coord.longitude)

In [24]: cord = latlong('360 East 2nd Street, Los Angeles, CA')

In [25]: auth = tweepy.OAuthHandler("IBdUHsdCAKek8rsSxwLpC3sT", "u01SJ6Zx1CyHo3mVFmM26IA7bFpZW3mbRxF2FhzfuaA3rRL17")
auth.set_access_token("1115468387889475584-e427sS2npSxAZykyt8QkmZ56YmD86H", "Fy0hp00DKUNzQT5wxk5PRty1JBXVx9HOXHhQ4P3sSSkKg")

api = tweepy.API(auth)

if (not api):
    print ("Problem connecting to API")

In [26]: def hashtag_process(raw):
    if len(raw)==0:
```

Fig:5.1 Screen shot 1

```
In [18]: df['End Point Of Closure'] = df['Tweet Text'].apply(find_end_improved)

In [19]: df['Location'] = df['Hashtags (Location)'].apply(remove_punc)

In [20]: df['date'] = pd.to_datetime(df['Creation Time (UTC)'])
df['Creation Time'] = df['date'].apply(datetime_from_utc_to_local)

In [21]: df.drop(labels=['Creation Time (UTC)', 'date', 'Hashtags (Location)'], axis=1, inplace=True)

In [22]: df.head()

Out[22]:
```

	Tweet Text	Author	Coordinates	Closed Road	Start Point Of Closure	End Point Of Closure	Location	Creation Time
0	Issigalert!! the road is closed due to a brush...	TotalTrafficLA	[34.25846, -118.1052]	None	Mt Wilson Red Box Rd	Upper Big Tujunga Cyn Rd	Valyermo	2020-09-08 07:15:43
1	Issigalert!! overturned car blocking the left...	TotalTrafficLA	[34.02722, -118.2328]	10 (I-10)	Mateo St/Santa FE Ave	I-5/US-101	LosAngeles	2020-09-08 04:20:43
2	Overturned car blocking the two left lanes in ...	TotalTrafficLA	[34.02722, -118.2328]	10 (I-10)	Mateo St/Santa FE Ave	I-5/US-101	LosAngeles	2020-09-08 04:15:43
3	Wood debris blocking the left lane in #Montere...	TotalTrafficLA	[34.03345, -118.1457]	60 (SR-60)	Atlantic Blvd	I-710	MontereyPark	2020-09-08 04:05:43
4	A crash was moved to the right shoulder in #M...	TotalTrafficLA	[34.03476, -118.3376]	10 (I-10)	Crenshaw Blvd	Arlington Ave	MidCity	2020-09-07 22:15:44

```
In [23]: df.dropna(inplace=True)
```

Fig:5.2 Screen shot 2

```
In [28]: def remove_url(raw_str):
         clean_str = re.sub(r'http\S+', '', raw_str)
         return clean_str

In [29]: sample_submission = pd.read_csv("sample.csv")

In [30]: train_df = pd.read_csv("accident_train.csv")

In [31]: test_df = pd.read_csv("test.csv")

In [32]: train_df.head()

Out[32]:
```

	Unnamed: 0	id	keyword	location	text	target
0	0	20001	accident	Hawthorne	!!sig alert!! a crash investigation has prairi...	1
1	1	20002	accident	Los Angeles	!! sigalert !! a crash is blocking the right l...	1
2	2	20003	accident	Avocado Heights	A crash is blocking the carpool and left lane....	1
3	3	20004	accident	Los Angeles	A crash is blocking the two right lanes. in #S...	1
4	4	20005	accident	Avocado Heights	A crash is blocking the carpool and left lane....	1

```
In [33]: state = 1
```

Fig:5.3 Screen shot 3

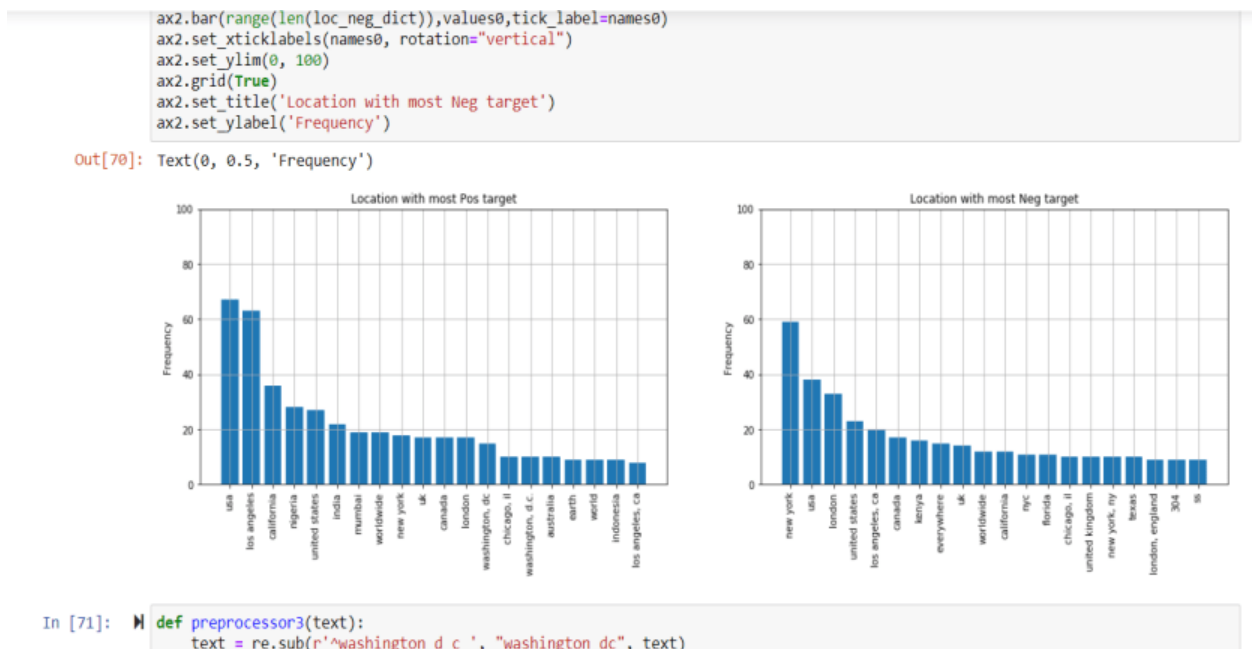
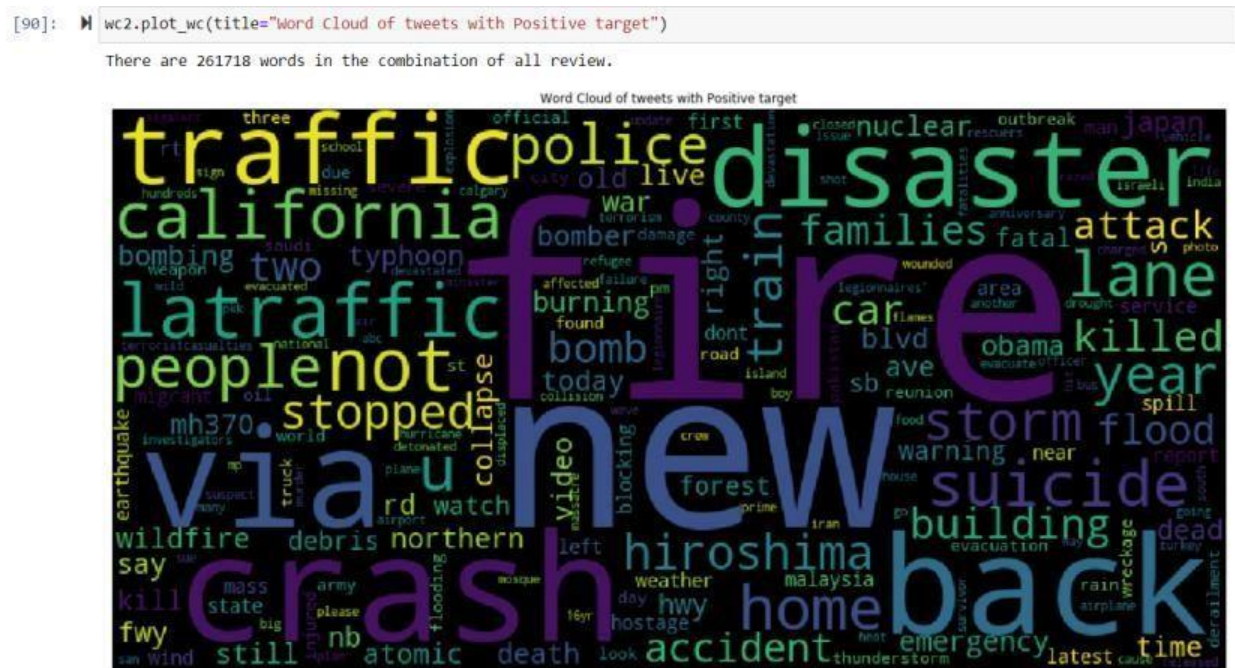


Fig:5.4 Screen shot 4



CHAPTER-6

SUMMARY AND CONCLUSION

CONCLUSION:

Our proposed model outperforms other deep learning methods. The leading proof of concept illustrates the applicability and accessibility of examining social media outcomes for traffic management. The proposed solution identified key areas within the domain of traffic social media. Evidently the enhanced use of social media for traffic management provides many prospects for developing traffic management solutions supporting public who are keen on traffic information. These results will gain from the integration of data from wearable and other types of technologies.

REFERENCES

- [1] K. M. Habibullah, A. Alam, S. Saha, A. Amin and A. K. Das, " A Driver-Centric Carpooling: Optimal Route-Finding Model using Heuristic Multi-Objective Search," 2019 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019.
- [2] Nguyen, H., Cai, C., Chen, F.: 'Automatic classification of traffic incident's severity using machine learning approaches', IET Intell. Transp. Syst., 2017, 11, (10), pp. 615– 623
- [3] Yue, M., Fan, L., Shahabi, C.: 'Traffic accident detection with spatiotemporal impact measurement'. Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Melbourne, Australia, 2018, pp. 471–482
- [4] Yuan, Y., Wang, D., Wang, Q.: 'Anomaly detection in traffic scenes via spatial-aware motion reconstruction', IEEE Trans. Intell. Transp. Syst., 2017, 18, (5), pp. 1198–1209.
- [4] M. A. A. Mamun, J. A. Puspo and A. K. Das, —An intelligent smartphone based approach using IoT for ensuring safe driving,|| 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, 2017, pp. 217-223.
- [5] T. Adhikary, A. K. Das, M. A. Razzaque, A. Almogren, M. Alrubaian, and M. M. Hassan, —Quality of Service Aware Reliable Task Scheduling in Vehicular Cloud Computing,|| Mobile Networks and Applications, Volume 21, Issue 3, pp 482-493, June 2016.
- [6] M. S. Satu, S. Ahamed, F. Hossain, T. Akter and D. M. Farid, "Mining traffic accident data of N5 national highway in Bangladesh employing decision trees," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp.

722- 725.

[7]H. İ. Bülbül, T. Kaya and Y. Tulgar, "Analysis for Status of the Road Accident Occurrence and Determination of the Risk of Accident by Machine Learning in Istanbul," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, 2016, pp. 426-430.

[8]BQ. He, —Research on the Application of Space-time Visual Analysis in Traffic AccidentsII, Xiamen University , June 2017. pp. 37–41 Chen W, Guo F, Wang F. —A survey of traffic data visualizationII. Intelligent Transportation Systems IEEE Transactions, 2015, 16(6):1- 15.

[9]Chen Yi, Liu Ying, Tian Shuai, Fan Chunlin, Sun Yuehong, —A Survey of Visual Analytical Techniques for Big Data in Food Safty Field,II Journal of Computer-Aided Design & Computer Graphics, Vol. 29, No. 1,2017, 9-14.

[10] XueHaibin, —Application of GIS Technology in Highway Bridge ManagementII, (Transportation Information Industry, No. 6, 2016, 131-133.

[11] Wang Peng, —The research and implementation of database in highway based on (GISII, Anhui Polytechnic University, June 2016, pp. 16-19

[12]JiangXiaorui, Tian Ya, Jiang Li, Liang Ronghua, —Visual analytics of urban road transportation data : a surveyII, Journal of Image and Graphics,Vol.20, No.4, 2015, 0454-0467.

[13] TianMiaomiao, Yang Qiuxiang. —Research on data transformation and indexing (method of map-based road condition dectection systemII , Computer Applications and Software. Vol. 31, No. 2, 2014, 66-70.

[14] Diehl A, Pelorosso L, Delrieux C, et al. —Visual analysis of spatiotemporal data

: Applications in weather forecasting. Computer Graphics Forum, 2015, 34(3), 381-390. [15] P. A. Nandurde and N. V. Dharwadkar, "Analyzing road accident data using machine learning paradigms," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud) (I-SMAC), Palladam, 2017, pp. 604-610.

[16] S. Kumar and D. Toshniwal, "A data mining approach to characterize road accident locations," Journal of Modern Transportation(2016), vol. 24, issue no. 1, pp. 62-72.

[17] Mahmood, Z., Haneef, O., Muhammad, N., et al.: 'Towards a fully automated car parking system', IET Intell. Transp. Syst., 2019, 13, pp. 293–302.

[18] Dong, Z.S., Zhu, W.X.: 'Homotopy methods based on ℓ_0 -norm for compressed sensing', IEEE Trans. Neural Netw. Learn. Syst., 2017, 29, pp. 1–

[19] Lu, S., Qiu, X., Shi, J., et al.: 'A pathological brain detection system based on extreme learning machine optimized by bat algorithm', CNS Neurol. Disord. Drug Targets, 2017, 16, (1), pp. 23–29

[20] Wang, H., Klaser, A., Schmid, C., et al.: 'Action recognition by dense trajectories'. Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 3169–3176

APPENDICES

A.PLAGIARISM REPORT

Report Title:	paper report
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/245616787a3e5fcfa0337c38fabd07bd7fc091617613793
Report Generated Date:	05 April, 2021
Total Words:	4512
Total Characters:	30412
Keywords/Total Words Ratio:	81.5%
Excluded URL:	No
Unique:	89%
Matched:	11%

B.BASE PAPER

DETECTING TRAFFIC INFORMATION FROM SOCIAL MEDIA TEXT USING DEEP LEARNING

Jupalli.Mani bhaskar¹, Konduri.Srikanth²,
UG Student, Dept. of CSE, Sathyabama Institute of
Science and Technology, Chennai, India.
manibhaskerjupalli@gmail.com,
²srikanthkonduri9@gmail.com.

B. Ankayarkanni²,
²Assistant Professor, Dept. of CSE, Sathyabama Institute of
Science and Technology, Chennai, India.
²ankayarkanni.s@gmail.com

Abstract-Social media, such as Twitter, Facebook, YouTube, Instagram, Tumblr etc is tremendously the go-to platform for emerging internet and mobile users to share user generated comments or experiences about a variety of traffic information and other forms of public expressions on traffic. In this domain, Deep Learning Techniques will be used for public sentiment analysis on traffic information through social media. In the existing system, the traffic information can be determined using Google Map or in FM Radio. Google Map will not give you the accurate Traffic Information. In Addition, it will not provide any information about the traffic events. FM Radio will not give you the traffic information. 24

1. INTRODUCTION

Social media has improved noteworthy incidence and importance over the last decade and has become a essential piece of several individuals' normal day after day existences. As social media customers expand on an expansive scope of subjects, well-known social networking web sites can fill in as a super base for exceptional statistics mining and statistics extraction packages. One plausibility amongst those might be the non-stop popularity of startling visitors occasions or abnormalities, which might be applied to assist visitors directors to discover and alleviate unsafe spots in a handy manner or to assist guests with making knowledgeable selections approximately their journey route. Traffic mishaps disappointed the visitors duties, separate the visitors stream, and reason critical city problems across the global. Major visitors mishaps wounds, or even fatalities. With the great increment within the usage of social gadget web sites like Twitter and Facebook, on-line community is buying and selling statistics as feelings, suppositions, feelings, and aims, which reflect their affiliations and inclination closer to a substance, occasion and strategy. Existing methodologies for the maximum element practice conventional system mastering strategies depending on verifiable mishap records. In this manner they fizzled to encompass the cross-area statistics, which includes spatial and temporal dependency. In the prevailing gadget, customers accumulate visitors statistics from diverse reassets, keep it, examine it through making use of diverse algorithms, textual content evaluation, visualize the visitors statistics, and post a precis record and visualization. The present gadget may be applied to always produce reviews from a lot of statistics reassets, the use of evaluation to cluster and discover

key groups, humans and content material within the visitors community. Authorities can have interaction with visualizations, searching, filtering or sorting to show insights within the statistics. The underlying statistics is without delay available through the spreadsheet. Social media analytics gear had been applied to examine, gain and examine publicly published visitors statistics.

1.2 Scope of the project

The main contributions of this project therefore are:

- 1) Data Analysis
- 2) Dataset Preprocessing
- 3) Training the Model
- 4) Testing of Dataset

1.3 Domain overview

Deep mastering has advanced hand-in-hand with the virtual era, which has delivered approximately an explosion of statistics in all paperwork and from each location of the global. This statistics, recognised absolutely as massive statistics, is drawn from reassets like social media, net seek engines, e-trade platforms, and on-line cinemas, amongst others. This great quantity of statistics is without problems available and may be shared thru fintech packages like cloud computing. Deep mastering is a system mastering approach that teaches computer systems to do what comes certainly to people: examine through example. Deep mastering is a key generation in the back of driverless cars, permitting them to apprehend a forestall signal, or to differentiate a pedestrian from a lamppost. It is the important thing to voice manipulate in patron gadgets like phones, tablets, TVs, and hands-loose speakers. Deep mastering is getting plenty of interest recently and for top motive. Its accomplishing outcomes that had been now no longer viable earlier than In deep mastering, a pc version learns to carry out category duties without delay from images, textual content, or sound. Deep mastering fashions can gain modern-day accuracy, on occasion exceeding human-stage overall performance. Models are educated through the use of a massive set of categorized statistics and neural community architectures that include many layers. However, the statistics, which

typically is unstructured, is so huge that it can take a long time for people to understand it and extract applicable statistics.

1.3.1 Deep learning

About Pandas

5 Pandas is a famous Python package deal for statistics science, and with top motive: it gives powerful, expressive and bendy statistics systems that make statistics manipulation and evaluation smooth, amongst many different matters. The DataFrame is this sort of systems. Pandas is a excessive-stage statistics manipulation device advanced through Wes McKinney. It is constructed at the Numpy package deal and its key statistics shape is known as the DataFrame. DataFrames permit you to shop and control tabular statistics in rows of observations and columns of variables. Pandas is constructed on pinnacle of the NumPy package deal, which means a variety of the shape of NumPy is used or replicated in Pandas. Data in pandas is frequently used to feed statistical evaluation in SciPy, plotting features from Matplotlib, and system mastering algorithms in Scikit-learn. Jupyter Notebooks provide an excellent surroundings for the use of pandas to do statistics exploration and modeling, however pandas also can be utilized in textual content editors simply as easily. Jupyter Notebooks supply us the opportunity to execute code in a selected modular rather than going for walks the whole file. This saves a variety of time whilst running with massive datasets and complicated transformations. Notebooks additionally offer an smooth manner to visualise pandas DataFrames and plots. As a counted number of reality, this newsletter become created completely in a Jupyter Notebook.

Numpy

Numpy is the middle library for clinical computing in Python. It gives a excessive-overall performance multidimensional array item, and gear for running with those arrays. If you're already acquainted with MATLAB, you would possibly discover this educational beneficial to get commenced with Numpy. A numpy array is a grid of values, all the identical type, and is listed through a tuple of nonnegative integers. The wide variety of dimensions is the rank of the array; the form of an array is a tuple of integers giving the scale of the array alongside every measurement. NumPy is, much like SciPy, Scikit-Learn, Pandas, and many others. one of the programs which you simply cant pass over whilst youre mastering statistics science, specifically due to the fact this library gives you with an array.

Matplotlib

Matplotlib Plotting of statistics may be notably made viable in an interactive manner through Matplotlib, that is a plotting library that may be proven in Python

scripts. Plotting of graphs is part of statistics visualization, and this belongings may be executed through utilising Matplotlib. Matplotlib uses many fashionable-reason GUI toolkits, which include wxPython, Tkinter, QT, and many others., in an effort to offer item-orientated APIs for embedding plots into packages. John D. Hunter become the person that at first wrote Matplotlib, and its lead developer become Michael Droettboom. One of the loose and open supply Python library which is essentially used for technical and clinical computing is Python SciPy. Matplotlib is broadly utilized in SciPy as maximum clinical calculations require plotting of graphs and diagrams. Matplotlib is a plotting library like GNUplot. The foremost benefit closer to GNUplot is the reality that Matplotlib is a Python module. Due to the developing hobby in python the recognition of matplotlib is usually growing as properly. Matplotlib may be used to create e-book excellent figures in a lot of hardcopy codecs and interactive environments throughout platforms. Another feature of matplotlib is its steep mastering curve, this means that customers generally make fast development after having commenced. The official internet site has to mention the subsequent approximately this: "matplotlib attempts to make smooth matters smooth and difficult matters viable. You can generate plots, histograms, energy spectra, bar charts, errorcharts, scatterplots, and many others, with only a few traces of code

Deep Learning:

Deep Learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. All layers in between are called Hidden Layers. The word deep means the network join

- The last layer is called the Output Layer
- The first layer is called the Input Layer
- Deep learning algorithms are constructed with connected layers.

neurons in more than two layers. A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.



Neuron:

Artificial Neural Networks include layers of neurons. A neuron is a computational unit that calculates a chunk of statistics primarily based totally on weighted enter parameters. Inputs well-known through the neuron are one at a time weighted. Inputs are summed and handed thru a non-linear characteristic to supply output. Each layer of neurons detects a few extra statistics, which include edges of factors in a photograph or tumors in a human body.

Multiple layers of neurons may be used to stumble on extra statistics approximately enter parameters.

Nodes:

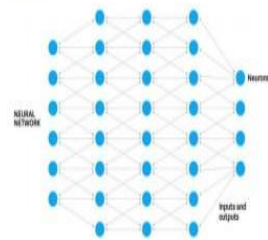
Artificial Neural Network is an interconnected organization of nodes corresponding to the huge community of layers of neurons in a mind. Each round node represents an synthetic neuron and an arrow represents a connection from the output of 1 neuron to the enter of some other.

Inputs:

Inputs are handed into the primary layer. Individual neurons get hold of the inputs, with every of them receiving a selected fee. After this, an output is produced primarily based totally on those values.

Outputs:

The outputs from the first layer are then passed into the second layer to be processed. This continues until the final output is produced. The assumption is that the correct output is



Some of the commonly used neural networks are as follows:

1. Artificial Neural Network (ANN)
2. Convolutional Neural Network (CNN)
3. Recurrent Neural Network (RNN)
4. Deep Neural Network (DNN)
5. Deep Belief Network (DBN)

Artificial neural networks are one of the foremost gear utilized in system mastering. As the neural a part of their call shows, they may be mind-stimulated structures that are meant to copy the manner that we people examine. Neural networks encompass enter and output layers, as properly as (in maximum cases) a hidden layer together with gadgets that remodel the enter into some thing that the output layer can use. They are extremely good gear for locating styles that are a ways too complicated or severa for a human programmer to extract and train the system to apprehend. Generally, the running of a human mind through making the proper connections is the concept in the back of ANNs. That become constrained to apply of silicon and wires as dwelling neurons and dendrites. Here,

neurons, a part of human mind. That become composed of 86 billion nerve cells. Also, linked to different hundreds of cells through Axons. Although, there are numerous inputs from sensory organs. That become well-known through dendrites. As a result, it creates electric powered impulses. That is used to journey thru the Artificial neural community. Thus, to address the exceptional problems, neuron ship a message to some other neuron. As a result, we will say that ANNs are composed of a couple of nodes. That imitate organic neurons of the human mind. Although, we join those neurons through links. Also, they have interaction with every different. Although, nodes are used to take enter statistics. Further, carry out easy operations at the statistics. As a result, those operations are handed to different neurons. Also, output at every node is known as its activation or node fee.

II. LITERATURE SURVEY

This paper investigates a way to limit the inter-server visitors amongst a cluster of social media servers thru joint statisticspartitioning and replication optimization. We officially outline the trouble and take a look at its hardness. We then advise a TrafficOptimized Partitioning and Replication (TOPR) approach to constantly adapt statistics placement in keeping with diverse dynamics. Evaluations with actual Twitter and LiveJournal social graphs display that TOPR now no longer simplest reduces the inter-server visitors appreciably however additionally saves a good deal garage value of replication in comparison to kingdom-of-the-artwork strategies. We additionally benchmark TOPR towards the offline most efficient through a binary linear program. We do now no longer explicitly limitation the wide variety of slave replicas in our problemdefinition. It is intuitive that creating too many slave replicas (e.g., full replication) would produce a huge amount of write incurred visitors and bring about excessive interserver visitors. Thus, even supposing there may be no capability restrict enforced on slave replicas, it'd now no longer cause the introduction of an immoderate and unrealistic wide variety of slave replicas. There a operation does not cause any inter-server communication, however it nonetheless induces request processing at each servers. On the alternative hand, a write operation has to supply inter-server conversation for synchronizing replicas and additionally desires to be processed through a couple of servers. Our trouble formula may be tailored to limit the whole wide variety of requests processed through all servers.

III. PROPOSED METHOD

A. ARCHITECTURE

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components.

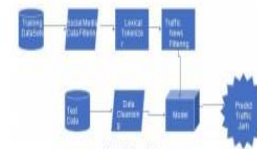


Fig 1 Architecture

B. SYSTEM IMPLEMENTATION



Fig 2 Modules

C. DATA PREPARATION:

Data Preparation is a statistics mining approach that entails reworking uncooked statistics into an comprehensible format. Real-global statistics is frequently incomplete, inconsistent, and/or missing in positive behaviours or trends, and is in all likelihood to include many errors. Data Preparation is a validated approach of resolving such problems. During the textual content Preparation section, the article's content material is split into tokens and nonsignificant characters are removed. Named entities are recognized by the elementary method: the primary upper-case letter with corpus-primarily based totally statistic is ideal sufficient for The tokens are divided through their POS tag and typically simplest nouns and adjectives may be declared as capacity key phrases. The subsequent concept is to select simplest not unusual place phrases as opposed to precise ones. Keywords are frequently used for clustering and a key word is vain whilst it's miles assigned to only some articles. This is the motive why we eliminate tokens with low frequency and installation policies for fashionable nouns. There isn't any such rule for named entities. The final tokens and named entities are declared as visitors statistics applicants.

D. FEATURE ENGINEERING:

In textual content processing, phrases of the textual content constitute discrete, express capabilities. How will we encode such statistics in a manner which is prepared to be utilized by the algorithms? The mapping from textual statistics to actual valued vectors is known as characteristic

extraction. We make the listing of precise phrases withinside the textual content corpus known as vocabulary. Then we will constitute every sentence or file as a vector with every phrase represented as 1 for gift and zero for absent from the vocabulary. Another illustration may be counted number the wide variety of instances every phrase seems in a file. The maximum famous method is the use of the Term Frequency/Inverse Document Frequency (TF-IDF) approach.

E. TRANSFORM TO TRAFFIC INFORMATION:

The Traffic Information extraction section carries simplest the approach for eliminating vain applicants whose rating is decrease than threshold fee. This boundary may be modified through the wide variety of asked automated key phrases. The Traffic Information extraction element may be defined through those steps: NLP approach exciting n-grams are chosen. The desire is primarily based totally on their POS tag styles and the corpus frequency is counted simplest for those n-grams. These n-grams may be marked as Traffic Information applicants. A rating of significance is counted for every Traffic Information candidate. This rating carries the n-gram corpus frequency and TF-IDF rating for every phrase. The rating is used for file Traffic Information selection. Derivation Traffic Information applicants are merged with named entities or character key phrases if their co-incidence is good sized for this file.

IV. EXPERIMENTS

A. Determination of the CNN Model: We repair the activation characteristic of convolutional layers to rectifier ($x = \max(x, 0)$), a typically used activation characteristic in CNN neural networks. And the loss characteristic is described as binary cross-entropy that is minimized through Adam optimizer [50]. To examine the properly-finished category fashions, we've carried out a sequence of experiments to discover the impact of structure additives on version overall performance. In the experiments of exploring the impact of clear out out wide variety, the clear out out wide variety is chosen from one hundred, a hundred and fifty. According to the outcomes, growing the wide variety of filters can yield higher overall performance, even as it takes longer time to teach the version. And the fashions with a hundred and fifty filters gain almost identical performances with the version with one hundred filters, which shows the most important wide variety of filters in grid seek may be a hundred and fifty. In the experiments of exploring the impact of clear out out duration, clear out out duration is chosen from [2, three, four, five, 6, 10]. According to the outcomes, converting the clear out out duration does now no longer enhance performances a good deal for our dataset. To gain the first-rate overall performance version, we run grid seek over the clear out out wide variety and clear out out duration of convolutional layers, and the pooling strategy. The clear

out out wide variety is selected from {20, 50, one hundred, a hundred and fifty} and the clear out out duration is ready from 2 to six with a stride 1. The following pooling operation is selected from most pooling and common pooling. In addition, we've additionally evaluated the overall performance of the version through including a vanilla hidden layer among the pinnacle pooling layer and the After going for walks grid seek, the version with one hundred filters whose duration is four achieves the first-rate F1 rating, that is summarized in Table III. The first-rate structure has a most pooling layer and a vanilla hidden layer with 50 output gadgets. B Determination of the LSTM Model: For the LSTM version, the loss characteristic is likewise described as binary cross-entropy. We use RMSprop optimizer [51] to limit the loss characteristic. To teach the version efficiently, we've carried out a sequence of experiments to discover the impact of the scale of LSTM hidden states and the mastering fee on version overall performance. In the experiments of exploring the impact of a mastering fee on version overall performance, the mastering fee is chosen from zero.01 to zero.05 with a stride zero.01. Experiments display that a decrease mastering fee offers higher performances in particular for Authorized certified use constrained to: Cornell University Library. Downloaded on September 02,2020 at 17:52:02 UTC from IEEE Xplore. Restrictions practice. take into account and F1 rating because the schooling technique going on. In the experiments of investigating the impact of the scale of LSTM hidden states on version overall performance, the scale of LSTM hidden kingdom is selected from {five, 10, 20, 50}. We discovered that the fashions with five and 10 LSTM hidden kingdom gadgets gain almost the identical overall performance after schooling epoch 30, that are higher than that of the fashions with 20 and 50 LSTM hidden kingdom gadgets. To get the LSTM version with first-rate overall performance, a grid seek is implemented. We select the scale of LSTM hidden states from {five, 10, 20} and the mastering fee from zero.01 to zero.03 with a stride zero.01. In addition, we discover whether or not to stack LSTM layers could enhance the performances. Experiments display that the version with stacked LSTM layers that have 10 output gadgets and zero.01 mastering fee obtains the first-rate F1 rating. C.Determination of the LSTM-CNN Model: In LSTM-CNN version, the primary layer is a LSTM layer. And following is a convolutional layer and a most pooling layer. The length of LSTM hidden kingdom is selected from {five, 10, 20, 50}. For convolutional layer, the clear out out wide variety is selected from one hundred and the clear out out duration is ready from 2 to six with a stride 1. And the mastering fee is selected from zero.01 to zero.03 with a stride zero.01. After acting grid seek, we gain the first-rate version.

V.RESULT

The detection outcomes of visitors applicable and inappropriate instructions with exceptional category methods are given in Table V. The F1 measures of the CNN version, the LSTM version and the LSTM-CNN version are near for each visitors applicable elegance and visitors inappropriate elegance, and the LSTM-CNN version gain the best F1 rating. All those 3 fashions proposed on this paper enhance the overall performance than that of the bow-SVM version, the veseq-SVM version and the MLP version. It is really well worth to say that the bow-SVM is easy but has an excellent overall performance. For visitors applicable elegance, the precision of the bow-SVM version is the best in comparison to that of different strategies, however this version has a decrease take into account this means that it'd stumble on fewer visitors applicable microblogs and hence may also pass over a few visitors statistics which include visitors incidents. It additionally desires to factor that the LSTM version has the best take into account fee for visitors applicable elegance, but a exceedingly low precision. A decrease precision fee method that a version has much less exactness and has extra fake advantageous instances. As the performances of bow-SVM, LSTM, CNN and LSTM-CNN methods are near, we similarly talk their advantages and boundaries. The proposed deep neural community method extracts deep capabilities automatically, that is end-to-end mastering with out manually choosing capabilities in comparison to the bow-SVM method. So the deep mastering method desires much less professional understanding on language. Besides, the measurement of characteristic area within the deep mastering method is a good deal decrease than that within the bow-SVM approach. In our experiments, the measurement of characteristic area implemented within the bow-SVM approach is 276 679 this is decided on from 1 128 748 capabilities through dimensionality reduction, even as the measurement of characteristic area for different strategies on this paper is 31 200. The boundaries of the deep mastering method also are obvious. The end-to-end deep mastering, as a black box, lacks version interpretability. Meanwhile, because of excessive version complexity, the deep mastering method desires a good deal extra schooling time than that of the bow-SVM approach

VI.CONCLUSION

In this paper, we mentioned the rising social transportation subject matter on extracting visitors statistics from social media. We proposed deep mastering strategies to stumble on visitors associated microblogs from Sina Weibo. Different from the conventional n gram language fashions taking as enter one-warm vector illustration of phrases, we implemented phrase embedding illustration of phrases and used CNN, LSTM, LSTM-CNN to categorise quick texts of Sina Weibo. We were given the phrase embedding with

non-stop vector illustration of phrases through the CBOW version the use of three billion microblogs gathered among 2009 and 2011 from Sina Weibo. One-warm vector illustration of phrases does now no longer have semantic similarities among phrases even as phrase embeddings have. The phrase embeddings had been fed into CNN, LSTM, and LSTM-CNN. We investigated the impact of version structure and hyperparameters on version overall performance and ran grid seek to decide the first-rate structure of CNN, LSTM, and LSTM-CNN. Experiments confirmed that the proposed deep mastering strategies with phrase embeddings outperform the competing strategies. In destiny studies, the method may be similarly superior and stepped forward in numerous ways. First, we are able to accumulate extra microblogs to categorise visitors applicable microblogs into certain classes like visitors accidents, visitors status, and many others. Second, the certain entities, e.g. place and time of incidence of visitors occasions, can be extracted from uncooked microblogs. Furthermore, we are able to enhance quick and long time visitors prediction through fusing on-line social media and conventional bodily visitors detector statistics.

REFERENCES

- [1] K. M. Habibullah, A. Alam, S. Saha, A. Amin and A. K. Das, "A Driver-Centric Carpooling: Optimal Route-Finding Model using Heuristic Multi-Objective Search," 2019 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019.
- [2] Yuan, Y., Wang, D., Wang, Q.: 'Anomaly detection intrafficscenesiaspatial-awaremotionreconstruction' JIEEETrans.Intell.Transp.Sys t,2017,18,(5),pp.1198-1209
- [3] M. A. A. Mamun, J. A. Puspo and A. K. Das, "An intelligent smartphone based approach using IoT for ensuring safe driving," 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, 2017, pp. 217- 223.
- [4] T.Adhikary,A.K.Das,M.A.Razzaque,A.Almogren,M.A.Irubaian,andM.M.Hassan, "Quality of Service Aware Reliable Task Scheduling in Vehicular Cloud Computing," Mobile Networks and Applications, Volume 21, Issue 3, pp 482-493, June2016.
- [5] M. S. Satu, S. Ahamed, F. Hossain, T. Akter and D. M. Farid, "Mining traffic accident data of N5 national highway in Bangladesh employing decision trees," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp. 722- 725.
- [6] H. İ. Bülbül, T. Kaya and Y. Tuglar, "Analysis for Status of the Road Accident Occurrence and Determination of the Risk of Accident by Machine Learning in Istanbul," 2016 15th IEEE International

Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, 2016, pp. 426-430.

- [7] BQ. He, "Research on the Application of Space-time Visual Analysis in Traffic Accidents", Xiamen University , June 2017, pp. 37-41 Chen W., Guo F, Wang F. "A survey of traffic data visualization". Intelligent Transportation Systems IEEE Transactions, 2015, 16(6):1- 15.
- [8] Chen Yi, Liu Ying, Tian Shuai, Fan Chunlin, Sun Yuehong, "A Survey of Visual Analytical Techniques for Big Data in Food Safty Field," Journal of Computer-Aided Design & Computer Graphics, Vol. 29, No. 1,2017, 9-14.
- [9] Xue Haibin, "Application of GIS Technology in Highway Bridge Management", Transportation Information Industry, No. 6, 2016, 131-133.
- [10] [7] Wang Peng, "The research and implementation of database in highway based on GIS", Anhui Polytechnic University, June 2016, pp. 16-19.
- [11] Jiang Xiaorui, Tian Ya, Jiang Li, Liang Ronghua, "Visual analytics of urban road transportation data : a survey", Journal of Image and Graphics,Vol.20, No.4, 2015, 0454-0467.
- [12] Tian Miaomiao, Yang Qixiang, "Research on data transformation and indexing method of map-based road condition detection system", Computer Applications and Software. Vol. 31, No. 2, 2014, 66-70.
- [13] Diehl A, Pelorosso I, Delrieux C, et al. "Visual analysis of spatiotemporal data : Applicationsinweatherforecasting".ComputerGraphicsForum,2015,34(3),381-390.
- [14] P. A. Nandurde and N. V. Dharwadkar, "Analyzing road accident data using machine learningparadigms,"2017InternationalConferenceonI-SMAC(IoTinSocial,Mobile, Analytics, and Cloud) (I-SMAC), Palladam, 2017, pp.604-610.
- [15] S. Kumar and D. Toshniwal, "A data mining approach to characterize road accident locations," Journal of Modern Transportation(2016), vol. 24, issue no. 1, pp. 62-72.
- [16] Mahmood, Z., Haneef, O., Muhammad, N., et al.: 'Towards a fully automated car parking system', IET Intell. Transp. Syst., 2019, 13, pp. 293-302
- [17] Khalid, S., Muhammad, N., Sharif, M.: 'Automatic measurement of the traffic sign with digitalsegmentationandrecognition'.IETIntell.Transp.Syst.,2018,13,(2),pp.269-279
- [18] Dong, Z.S., Zhu, W.X.: 'Homotopy methods based on l0-norm for compressed sensing',

A.SOURCE CODE

Tweet scraping

```
stance import geodesic
import json import tweepy
import pandas as pd
from geopy.geocoders import Nominatim
from geopy.di

In [2]: geolocator = Nominatim(user_agent="DSI_US_8_LA")

In [3]: # Function to take a user-entered address and return it's coordinates
def latlong(point):
    coord = geolocator.geocode(point)
    return (coord.latitude, coord.longitude)

In [4]: cord = latlong('360 East 2nd Street, Los Angeles, CA')
In [5]: auth = tweepy.OAuthHandler("IBdUHHsdCAKek8rsSxWLpC3sT", "u01SJ6Zx1CyHo3mVFmM26
IA7bFpZW3mbRxRF2FhzfuaA3rRL17")
auth.set_access_token("1115468387889475584-e427sS2npSxAZykyt8QkmZ56YmD86H",
"Fy0hpOODKUNzQT5wxk5PRty1JBXVx9HOXhhQ4P3sSSkKg")

api = tweepy.API(auth)

if (not api):
    print ("Problem connecting to API")
In [6]: def hashtag_process(raw):
    if len(raw) == 0:
        return None
    else:
        output = []
        for x in raw:
            if x.get('text') != 'LATraffic':
                output.append(x.get('text'))
        return output
In [7]: def TweetSearch(location, distance=25, user = 'TotalTrafficLA', limit = 1):
    full_text = []
    author = []
    creation_time = []
    hashtags = []
    geo = []
```

```

for tweet
intweeepy.Cursor(api.user_timeline,id=user,tweet_mode='extended').items(limit):
print(type(tweet))
iftweet.geo!=None:
if geodesic(location,tweet.geo['coordinates']).miles <= distance:
print(tweet.full_text)
full_text.append(tweet.full_text)
author.append(tweet.author.screen_name)
creation_time.append(tweet.created_at)
hashtags.append(hashtag_process(tweet.entities['hashtags']))
geo.append(tweet.geo['coordinates'])

returnpd.DataFrame([full_text,author,geo,hashtags,creation_time],\
                    index=['Tweet Text','Author','Coordinates','Hashtags
(Location)','Creation Time (UTC)']).T
In [8]:df=TweetSearch("34.1581656060606, -118.333947181818")
<class 'tweepy.models.Status'>
!!sig alert!! a multi-car crash. emergency crews on scene. only the right lane is
open. in #Arleta on I-5 NB after Van Nuys Blvd, stopped traffic back to Osborne St
#LATraffic https://t.co/e2e6MWkOvI
In [9]:
df
Out[9]:

```

	Tweet Text	Author	Coordinates	Hashtags (Location)	Creation Time (UTC)
0	!!sig alert!! a multi-car crash. emergency cre...	TotalTrafficLA	[34.25315, - 118.4311]	[Arleta]	2021-03-02 06:20:43

```

In [10]:df.to_csv('tweetsNew.csv',index=False)

```

TWEET PARSING

```
import pandas as pd
import re
from datetime import datetime
import time
```

In [2]:

```
df = pd.read_csv('tweets.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Tweet Text	Author	Coordinates	Hashtags (Location)	Creation Time (UTC)
0	!!sigalert!! the road is closed due to a brush...	TotalTrafficLA	[34.25846, -118.1052]	['Valyermo']	2020-09-08 01:45:43
1	!!sigalert!! overturned car blocking the left...	TotalTrafficLA	[34.02722, -118.2328]	['LosAngeles']	2020-09-07 22:50:43
2	Overturned car blocking the two left lanes in ...	TotalTrafficLA	[34.02722, -118.2328]	['LosAngeles']	2020-09-07 22:45:43
3	Wood debris blocking the left lane in #Montere...	TotalTrafficLA	[34.03345, -118.1457]	['MontereyPark']	2020-09-07 22:35:43
4	A crash was moved to the right shoulder. in #M...	TotalTrafficLA	[34.03476, -118.3376]	['MidCity']	2020-09-07 16:45:44

In [4]:

```
df.shape
```

Out[4]:

```
(49, 5)
```

In [5]:

```
def bad_tweet(tweet_text):
    delete = 0
    for i in ['clear', 'reopen']:
        if i in tweet_text:
            delete = 1
    if delete:
        return True
    else:
        return False
```

In [6]:

```
df['bad_tweet'] = df['Tweet Text'].apply(bad_tweet)
```

```

df=df[df['bad_tweet']==False]

df.drop(labels=['bad_tweet'],axis=1,inplace=True)

df.head()

```

In [7]:

In [8]:

In [9]:

Out[9]:

	Tweet Text	Author	Coordinates	Hashtags (Location)	Creation Time (UTC)
0	!!sigalert!! the road is closed due to a brush...	TotalTrafficLA	[34.25846, -118.1052]	['Valyermo']	2020-09-08 01:45:43
1	!!sigalert!! overturned car blocking the left...	TotalTrafficLA	[34.02722, -118.2328]	['LosAngeles']	2020-09-07 22:50:43
2	Overturned car blocking the two left lanes in ...	TotalTrafficLA	[34.02722, -118.2328]	['LosAngeles']	2020-09-07 22:45:43
3	Wood debris blocking the left lane in #Montere...	TotalTrafficLA	[34.03345, -118.1457]	['MontereyPark']	2020-09-07 22:35:43
4	A crash was moved to the right shoulder. in #M...	TotalTrafficLA	[34.03476, -118.3376]	['MidCity']	2020-09-07 16:45:44

In [10]:

```
df.shape
```

Out[10]:

```
(49, 5)
```

In [11]:

```

def find_road_improved(text):

    on_split=text.split(' on ')
    if len(on_split) ==1:
        return None
    else:

        output =" ".join(on_split[1].split(" ")[0:2])

    if bool(re.match(r'[HIU0-9]',output)):
        return output

def find_start_improved(text):

```

In [12]:

```

on_split=text.split(' on ')
iflen(on_split) ==1:
returnNone
else:

comma_split=on_split[1].split(",")[0]
iflen(comma_split.split(" at ")) >1:
returncomma_split.split(" at ")[1].split(" and")[0].split("#")[0]
eliflen(comma_split.split(" before ")) >1:
returncomma_split.split(" before ")[1].split(" and")[0].split("#")[0]
eliflen(comma_split.split(" after ")) >1:
returncomma_split.split(" after ")[1].split(" and")[0].split("#")[0]
eliflen(comma_split.split(" between ")) >1:
returncomma_split.split(" between ")[1].split(" and")[0].split("#")[0]
eliflen(comma_split.split(" approaching ")) >1:
returncomma_split.split(" approaching ")[1].split(" and")[0].split("#")[0]

```

In [13]:

```

def find_end_improved(text):

```

```

on_split=text.split(' on ')
iflen(on_split) ==1:
returnNone
else:

```

```

after_on=on_split[1]
iflen(after_on.split(" to ")) >1:
returnafter_on.split(" to ")[1].split(",")[0].split("#")[0]
eliflen(after_on.split(" and ")) >1:
returnafter_on.split(" and ")[1].split(",")[0].split("#")[0]

```

In [14]:

```

def datetime_from_utc_to_local(utc_datetime):
now_timestamp=time.time()
offset =datetime.fromtimestamp(now_timestamp) -
datetime.utcnowfromtimestamp(now_timestamp)
returnutc_datetime+ offset

```

In [15]:

```

def remove_punc(text):
returnre.sub("[^A-Za-z]", "", text)

```

In [16]:

```

df['Closed Road'] =df['Tweet Text'].apply(find_road_improved)

```

In [17]:

```

df['Start Point Of Closure'] =df['Tweet Text'].apply(find_start_improved)

```

In [18]:

```

df['End Point Of Closure'] =df['Tweet Text'].apply(find_end_improved)

```

```

df['Location'] =df['Hashtags (Location)'].apply(remove_punc)
df['date'] =pd.to_datetime(df['Creation Time (UTC)'])
df['Creation Time'] =df['date'].apply(datetime_from_utc_to_local)
df.drop(labels=['Creation Time (UTC)', 'date', 'Hashtags (Location)'],axis=1,inplace=True)
df.head()

```

Out[22]:

	Tweet Text	Author	Coordinates	Closed Road	Start Point Of Closure	End Point Of Closure	Location	Creation Time
0	!!sigalert!! the road is closed due to a brush...	TotalTrafficLA	[34.25846, -118.1052]	None	Mt Wilson Red Box Rd	Upper Big Tujunga Cyn Rd	Valyermo	2020-09-08 07:15:43
1	!!sigalert!! overturned car blocking the left...	TotalTrafficLA	[34.02722, -118.2328]	10 (I-10	Mateo St/Santa FE Ave	I-5/US-101	LosAngeles	2020-09-08 04:20:43
2	Overturned car blocking the two left lanes in ...	TotalTrafficLA	[34.02722, -118.2328]	10 (I-10	Mateo St/Santa FE Ave	I-5/US-101	LosAngeles	2020-09-08 04:15:43
3	Wood debris blocking the left lane in #Montere...	TotalTrafficLA	[34.03345, -118.1457]	60 (SR-60	Atlantic Blvd	I-710	MontereyPark	2020-09-08 04:05:43
4	A crash was moved to the right shoulder. in #M...	TotalTrafficLA	[34.03476, -118.3376]	10 (I-10	Crenshaw Blvd	Arlington Ave	MidCity	2020-09-07 22:15:44

```

df.dropna(inplace=True)
df.drop_duplicates(subset='Coordinates',inplace=True)
df.shape
(23, 8)
In [26]:df.head()

```

Out[26]:

	Tweet Text	Author	Coordinates	Closed Road	Start Point Of Closure	End Point Of Closure	Location	Creation Time
1	!!sigalert!! overturned car blocking the left...	TotalTrafficLA	[34.02722, -118.2328]	10 (I-10)	Mateo St/Santa FE Ave	I-5/US-101	LosAngeles	2020-09-08 04:20:43
3	Wood debris blocking the left lane in #Montere...	TotalTrafficLA	[34.03345, -118.1457]	60 (SR-60)	Atlantic Blvd	I-710	MontereyPark	2020-09-08 04:05:43
4	A crash was moved to the right shoulder. in #M...	TotalTrafficLA	[34.03476, -118.3376]	10 (I-10)	Crenshaw Blvd	Arlington Ave	MidCity	2020-09-07 22:15:44
5	A crash is blocking the carpool lane. in #Duar...	TotalTrafficLA	[34.13217, -117.9498]	210 (I-210)	I-605	Buena Vista St	Duarte	2020-09-07 20:20:43
7	A crash has all lanes temporarily stopped. in ...	TotalTrafficLA	[33.93583, -118.0929]	5 (I-5)	I-605	Lakewood Blvd/Rosemead Blvd	Downey	2020-09-07 19:10:44

```
In [27]:cols= ['Location','ClosedRoad','Start Point Of Closure','End Point Of Closure','Tweet Text',
'Creation Time','Coordinates','Author']
In [28]:df=df[cols]
In [29]:df.head()
Out[29]:
```

	Location	Closed Road	Start Point Of Closure	End Point Of Closure	Tweet Text	Creation Time	Coordinates	Author
1	LosAngeles	10 (I-10)	Mateo St/Santa FE Ave	I-5/US-101	!!sigalert!! overturned car blocking the left...	2020-09-08 04:20:43	[34.02722, -118.2328]	TotalTrafficLA
3	MontereyPark	60 (SR-60)	Atlantic Blvd	I-710	Wood debris blocking the left lane in #Montere...	2020-09-08 04:05:43	[34.03345, -118.1457]	TotalTrafficLA

	Location	Closed Road	Start Point Of Closure	End Point Of Closure	Tweet Text	Creation Time	Coordinates	Author
4	MidCity	10 (I-10	Crenshaw Blvd	Arlington Ave	A crash was moved to the right shoulder. in #M...	2020-09-07 22:15:44	[34.03476, -118.3376]	TotalTrafficLA
5	Duarte	210 (I-210	I-605	Buena Vista St	A crash is blocking the carpool lane. in #Duar...	2020-09-07 20:20:43	[34.13217, -117.9498]	TotalTrafficLA
7	Downey	5 (I-5	I-605	Lakewood Blvd/Rosemead Blvd	A crash has all lanes temporarily stopped. in ...	2020-09-07 19:10:44	[33.93583, -118.0929]	TotalTrafficLA

```
In [30]:df.to_csv('Twitter_Output_For_Dataframe.csv')
```

ACCIDENT-EVENT-TWEET PREDICTION

```
import numpy as np
```

In [2]:

```
import pandas as pd
```

In [3]:

```
import matplotlib.pyplot as plt
```

In [4]:

```
import re
```

In [5]:

```
from wordcloud import STOPWORDS
```

In [6]:

```
from wordcloud import WordCloud, ImageColorGenerator
```

In [7]:

```
import matplotlib.pyplot as plt
```

In [8]:

```
from keras.preprocessing.text import Tokenizer  
Using TensorFlow backend.
```

In [9]:

```
from keras.preprocessing import sequence
```

In [10]:

```
from keras.utils import to_categorical
```

In [11]:

```
import keras
```

In [12]:

```
from numpy import array
```

In [13]:

```
from numpy import asarray
```

In [14]:

```
from numpy import zeros
```

In [15]:

```
from keras.models import Sequential
```

In [16]:

```
from keras.models import Model
```

In [17]:

```
from keras.layers import Dense, Dropout, Embedding, LSTM, Input, Conv2D,  
MaxPooling2D, Conv1D, MaxPooling1D, BatchNormalization, TimeDistributed,  
Bidirectional, Flatten, GlobalMaxPool1D
```

In [18]:

```
from keras.layers import Bidirectional, Reshape, Flatten, GRU
```

In [19]:

```
from keras.layers.merge import concatenate
```

In [20]:

```
from keras.optimizers import Adam
```

In [21]:

```
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
```

In [22]:

```
from keras.models import load_model
```

In [23]:

```
from nltk.tokenize import word_tokenize
```

In [24]:

```
from keras.preprocessing.sequence import pad_sequences
```

In [25]:

```
from sklearn.model_selection import train_test_split
```

In [26]:

```
import string
```

In [27]:

```
%matplotlib inline
```

In [28]:

```
def remove_url(raw_str):
    clean_str = re.sub(r'http\S+', '', raw_str)
    return clean_str
```

In [29]:

```
sample_submission = pd.read_csv("sample.csv")
```

In [30]:

```
train_df = pd.read_csv("accident_train.csv")
```

In [31]:

```
test_df = pd.read_csv("test.csv")
```

In [32]:

```
train_df.head(10)
```

Out[32]:

	Unnamed: 0	id	keyword	location	text	target
0	0	20001	accident	Hawthorne	!!sig alert!! a crash investigation has prairi...	1
1	1	20002	accident	Los Angeles	!! sigalert !! a crash is blocking the right l...	1
2	2	20003	accident	Avocado Heights	A crash is blocking the carpool and left lane....	1
3	3	20004	accident	Los Angeles	A crash is blocking the two right lanes. in #S...	1
4	4	20005	accident	Avocado Heights	A crash is blocking the carpool and left lane....	1
5	5	20006	accident	Montebello	A crash is blocking the two left lanes. in #M...	1
6	6	20007	accident	Baldwin Park	Road flares are burning in the two right lanes...	1

	Unnamed: 0	id	keyword	location	text	target
7	7	20008	accident	Los Angeles	Stalled car in the center divider in #LosAngel...	1
8	8	20009	accident	California	Road is closed in both directions due to brush...	1
9	9	20010	accident	Commerce	Earlier crash moved out of lanes in #Commerce ...	1

In [33]:

```
state =1
```

In [34]:

```
train_df = train_df.sample(frac=1,random_state=state)
```

In [35]:

```
test_df = test_df.sample(frac=1,random_state=state)
```

In [36]:

```
train_df.reset_index(inplace=True, drop=True)
```

In [37]:

```
test_df.reset_index(inplace=True, drop=True)
```

In [38]:

```
train_df.head(10)
```

Out[38]:

	Unnamed: 0	id	keyword	location	text	target
0	1040	1506	body%20bags	NaN	I love The body shop%Ü"s bags??n\n#cutekitten...	0
1	1264	1822	burned	San Diego CA	Fire burning on Pendleton has burned 300 acres...	1
2	6464	9248	sunk	glasgow	Near them on the sand half sunk a shattered vi...	1
3	5694	8126	rescued	Earth	Migrants Rescued After Boat Capsizes Off Libya...	1
4	472	682	attack	portland, oregon	illegal alien released by Obama/DHS 4 times Ch...	1
5	477	687	attack	NaN	Heart disease prevention: What about secondhan...	0
6	958	1385	body%20bag	New York	Louis Vuitton Monogram Sophie Limited Edition ...	0
7	6244	8920	snowstorm	Deployed in the Middle East	@CacheAdvance besides your nasty thunderstorm ...	0
8	6739	9656	thunderstorm	NC	I need a thunderstorm please!	0

Unnamed: 0	id	keyword	location	text	target
9	4799	6830	loud%20bang	Kenya	Kijima_Matako: Breaking news! Unconfirmed! I j... 0

```
test_df.head()
```

In [39]:

Out[39]:

	id	keyword	location	text
0	6035	heat%20wave	Brooklyn, NY	I added some dumb ideas to beat the #summer he...
1	2168	catastrophic	NaN	If a £1 rise in wages is going to have such a...
2	317	annihilated	NaN	How do I step outside for 5 seconds and get an...
3	9682	tornado	Canada	Tornado warnings issued for southern Alberta h...
4	5857	hailstorm	Calgary, Alberta, Canada	Get out of the hailstorm and come down to @The...

In []:

In [40]:

```
def clean(tweet):
    tweet = re.sub(r"\x89\u", "", tweet)
    tweet = re.sub(r"\x89\u0", "", tweet)
    tweet = re.sub(r"\x89\u0", "", tweet)
    tweet = re.sub(r"\x89\uWhen", "When", tweet)
    tweet = re.sub(r"\x89\uI", "", tweet)
    tweet = re.sub(r"China\x89\u's", "China's", tweet)
    tweet = re.sub(r"let\x89\u's", "let's", tweet)
    tweet = re.sub(r"\x89\u+", "", tweet)
    tweet = re.sub(r"\x89\ua", "", tweet)
    tweet = re.sub(r"\x89\u\x9d", "", tweet)
    tweet = re.sub(r"\u", "", tweet)
    tweet = re.sub(r"\x89\u", "", tweet)
    tweet = re.sub(r"\x89\u\u", "", tweet)
    tweet = re.sub(r"from\uwounds", "from wounds", tweet)
    tweet = re.sub(r"\u", "", tweet)
    tweet = re.sub(r"\u", "", tweet)
    tweet = re.sub(r"Jap\u_n", "Japan", tweet)
    tweet = re.sub(r"\u", "e", tweet)
    tweet = re.sub(r"\u", "", tweet)
    tweet = re.sub(r"Suru\u", "Suruc", tweet)
```

```

tweet =re.sub(r"he's", "he is", tweet)
tweet =re.sub(r"there's", "there is", tweet)
tweet =re.sub(r"We're", "We are", tweet)
tweet =re.sub(r"That's", "That is", tweet)
tweet =re.sub(r"won't", "will not", tweet)
tweet =re.sub(r"they're", "they are", tweet)
tweet =re.sub(r"Can't", "Cannot", tweet)
tweet =re.sub(r"wasn't", "was not", tweet)
tweet =re.sub(r"don\x89\u00b0t", "do not", tweet)
tweet =re.sub(r"aren't", "are not", tweet)
tweet =re.sub(r"isn't", "is not", tweet)
tweet =re.sub(r"What's", "What is", tweet)
tweet =re.sub(r"haven't", "have not", tweet)
tweet =re.sub(r"hasn't", "has not", tweet)
tweet =re.sub(r"There's", "There is", tweet)
tweet =re.sub(r"He's", "He is", tweet)
tweet =re.sub(r"It's", "It is", tweet)
tweet =re.sub(r"You're", "You are", tweet)
tweet =re.sub(r"I'M", "I am", tweet)
tweet =re.sub(r"shouldn't", "should not", tweet)
tweet =re.sub(r"wouldn't", "would not", tweet)
tweet =re.sub(r"i'm", "I am", tweet)
tweet =re.sub(r"I\x89\u00b0am", "I am", tweet)
tweet =re.sub(r"I'm", "I am", tweet)
tweet =re.sub(r"Isn't", "is not", tweet)
tweet =re.sub(r"Here's", "Here is", tweet)
tweet =re.sub(r"you've", "you have", tweet)
tweet =re.sub(r"you\x89\u00b0ave", "you have", tweet)
tweet =re.sub(r"we're", "we are", tweet)
tweet =re.sub(r"what's", "what is", tweet)
tweet =re.sub(r"couldn't", "could not", tweet)
tweet =re.sub(r"we've", "we have", tweet)
tweet =re.sub(r"it\x89\u00b0s", "it is", tweet)
tweet =re.sub(r"doesn\x89\u00b0t", "does not", tweet)
tweet =re.sub(r"It\x89\u00b0s", "It is", tweet)
tweet =re.sub(r"Here\x89\u00b0s", "Here is", tweet)
tweet =re.sub(r"who's", "who is", tweet)
tweet =re.sub(r"I\x89\u00b0ave", "I have", tweet)
tweet =re.sub(r"y'all", "you all", tweet)
tweet =re.sub(r"can\x89\u00b0t", "cannot", tweet)
tweet =re.sub(r"would've", "would have", tweet)
tweet =re.sub(r"it'll", "it will", tweet)
tweet =re.sub(r"we'll", "we will", tweet)

```

```

tweet =re.sub(r"wouldn\x89Û^t", "would not", tweet)
tweet =re.sub(r"We've", "We have", tweet)
tweet =re.sub(r"he'll", "he will", tweet)
tweet =re.sub(r"Y'all", "You all", tweet)
tweet =re.sub(r"Weren't", "Were not", tweet)
tweet =re.sub(r"Didn't", "Did not", tweet)
tweet =re.sub(r"they'll", "they will", tweet)
tweet =re.sub(r"they'd", "they would", tweet)
tweet =re.sub(r"DON'T", "DO NOT", tweet)
tweet =re.sub(r"That\x89Û^s", "That is", tweet)
tweet =re.sub(r"they've", "they have", tweet)
tweet =re.sub(r"i'd", "I would", tweet)
tweet =re.sub(r"should've", "should have", tweet)
tweet =re.sub(r"You\x89Û^re", "You are", tweet)
tweet =re.sub(r"where's", "where is", tweet)
tweet =re.sub(r"Don\x89Û^t", "Do not", tweet)
tweet =re.sub(r"we'd", "we would", tweet)
tweet =re.sub(r"i'll", "I will", tweet)
tweet =re.sub(r"weren't", "were not", tweet)
tweet =re.sub(r"They're", "They are", tweet)
tweet =re.sub(r"Can\x89Û^t", "Cannot", tweet)
tweet =re.sub(r"you\x89Û^all", "you will", tweet)
tweet =re.sub(r"I\x89Û^d", "I would", tweet)
tweet =re.sub(r"let's", "let us", tweet)


tweet =re.sub(r">", ">", tweet)
tweet =re.sub(r"<", "<", tweet)
tweet =re.sub(r"&", "&", tweet)


tweet =re.sub(r"w/e", "whatever", tweet)
tweet =re.sub(r"w/", "with", tweet)
tweet =re.sub(r"USAgov", "USA government", tweet)
tweet =re.sub(r"recentlu", "recently", tweet)
tweet =re.sub(r"Ph0tos", "Photos", tweet)
tweet =re.sub(r"exp0sed", "exposed", tweet)
tweet =re.sub(r"<3", "love", tweet)
tweet =re.sub(r"amageddon", "armageddon", tweet)
tweet =re.sub(r"Trfc", "Traffic", tweet)
tweet =re.sub(r"8/5/2015", "2015-08-05", tweet)
tweet =re.sub(r"chest/torso", "chest / torso", tweet)
tweet =re.sub(r"WindStorm", "Wind Storm", tweet)
tweet =re.sub(r"8/6/2015", "2015-08-06", tweet)
tweet =re.sub(r"10:38PM", "10:38 PM", tweet)
tweet =re.sub(r"10:30pm", "10:30 PM", tweet)

```

```

tweet =re.sub(r"MH370:", "MH370 :", tweet)
tweet =re.sub(r"PM:", "Prime Minister :", tweet)
tweet =re.sub(r"Legionnaires:", "Legionnaires :", tweet)
tweet =re.sub(r"Latest:", "Latest :", tweet)
tweet =re.sub(r"Crash:", "Crash :", tweet)
tweet =re.sub(r"News:", "News :", tweet)
tweet =re.sub(r"derailment:", "derailment :", tweet)
tweet =re.sub(r"attack:", "attack :", tweet)
tweet =re.sub(r"Saipan:", "Saipan :", tweet)
tweet =re.sub(r"Photo:", "Photo :", tweet)
tweet =re.sub(r"Funtenna:", "Funtenna :", tweet)
tweet =re.sub(r"quiz:", "quiz :", tweet)
tweet =re.sub(r"VIDEO:", "VIDEO :", tweet)
tweet =re.sub(r"MP:", "MP :", tweet)
tweet =re.sub(r"UTC2015-08-05", "UTC 2015-08-05", tweet)
tweet =re.sub(r"California:", "California :", tweet)
tweet =re.sub(r"horror:", "horror :", tweet)
tweet =re.sub(r"Past:", "Past :", tweet)
tweet =re.sub(r"Time2015-08-06", "Time 2015-08-06", tweet)
tweet =re.sub(r"here:", "here :", tweet)
tweet =re.sub(r"fires.", "fires .", tweet)
tweet =re.sub(r"Forest:", "Forest :", tweet)
tweet =re.sub(r"Cramer:", "Cramer :", tweet)
tweet =re.sub(r"Chile:", "Chile :", tweet)
tweet =re.sub(r"link:", "link :", tweet)
tweet =re.sub(r"crash:", "crash :", tweet)
tweet =re.sub(r"Video:", "Video :", tweet)
tweet =re.sub(r"Bestnaijamade:", "bestnaijamade :", tweet)
tweet =re.sub(r"NWS:", "National Weather Service :", tweet)
tweet =re.sub(r".caught", ". caught", tweet)
tweet =re.sub(r"Hobbit:", "Hobbit :", tweet)
tweet =re.sub(r"2015:", "2015 :", tweet)
tweet =re.sub(r"post:", "post :", tweet)
tweet =re.sub(r"BREAKING:", "BREAKING :", tweet)
tweet =re.sub(r"Island:", "Island :", tweet)
tweet =re.sub(r"Med:", "Med :", tweet)
tweet =re.sub(r"97/Georgia", "97 / Georgia", tweet)
tweet =re.sub(r"Here:", "Here :", tweet)
tweet =re.sub(r"horror;", "horror ;", tweet)
tweet =re.sub(r"people;", "people ;", tweet)
tweet =re.sub(r"refugees;", "refugees ;", tweet)
tweet =re.sub(r"Genocide;", "Genocide ;", tweet)
tweet =re.sub(r".POTUS", ". POTUS", tweet)

```

```

tweet =re.sub(r"Collision-No", "Collision - No", tweet)
tweet =re.sub(r"Rear-", "Rear -", tweet)
tweet =re.sub(r"Broadway:", "Broadway :", tweet)
tweet =re.sub(r"Correction:", "Correction :", tweet)
tweet =re.sub(r"UPDATE:", "UPDATE :", tweet)
tweet =re.sub(r"Times:", "Times :", tweet)
tweet =re.sub(r"RT:", "RT :", tweet)
tweet =re.sub(r"Police:", "Police :", tweet)
tweet =re.sub(r"Training:", "Training :", tweet)
tweet =re.sub(r"Hawaii:", "Hawaii :", tweet)
tweet =re.sub(r"Selfies:", "Selfies :", tweet)
tweet =re.sub(r"Content:", "Content :", tweet)
tweet =re.sub(r"101:", "101 :", tweet)
tweet =re.sub(r"story:", "story :", tweet)
tweet =re.sub(r"injured:", "injured :", tweet)
tweet =re.sub(r"poll:", "poll :", tweet)
tweet =re.sub(r"Guide:", "Guide :", tweet)
tweet =re.sub(r"Update:", "Update :", tweet)
tweet =re.sub(r"alarm:", "alarm :", tweet)
tweet =re.sub(r"floods:", "floods :", tweet)
tweet =re.sub(r"Flood:", "Flood :", tweet)
tweet =re.sub(r"MH370;", "MH370 ;", tweet)
tweet =re.sub(r"life:", "life :", tweet)
tweet =re.sub(r"crush:", "crush :", tweet)
tweet =re.sub(r"now:", "now :", tweet)
tweet =re.sub(r"Vote:", "Vote :", tweet)
tweet =re.sub(r"Catastrophe.", "Catastrophe .", tweet)
tweet =re.sub(r"library:", "library :", tweet)
tweet =re.sub(r"Bush:", "Bush :", tweet)
tweet =re.sub(r";ACCIDENT", "; ACCIDENT", tweet)
tweet =re.sub(r"accident:", "accident :", tweet)
tweet =re.sub(r"Taiwan;", "Taiwan ;", tweet)
tweet =re.sub(r"Map:", "Map :", tweet)
tweet =re.sub(r"failure:", "failure :", tweet)
tweet =re.sub(r"150-Foot", "150 - Foot", tweet)
tweet =re.sub(r"failure:", "failure :", tweet)
tweet =re.sub(r"prefer:", "prefer :", tweet)
tweet =re.sub(r"CNN:", "CNN :", tweet)
tweet =re.sub(r"Oops:", "Oops :", tweet)
tweet =re.sub(r"Disco:", "Disco :", tweet)
tweet =re.sub(r"Disease:", "Disease :", tweet)
tweet =re.sub(r"Grows:", "Grows :", tweet)
tweet =re.sub(r"projected:", "projected :", tweet)
tweet =re.sub(r"Pakistan.", "Pakistan .", tweet)

```

```

tweet =re.sub(r"ministers:", "ministers :", tweet)
tweet =re.sub(r"Photos:", "Photos :", tweet)
tweet =re.sub(r"Disease:", "Disease :", tweet)
tweet =re.sub(r"pres:", "press :", tweet)
tweet =re.sub(r"winds.", "winds .", tweet)
tweet =re.sub(r"MPH.", "MPH .", tweet)
tweet =re.sub(r"PHOTOS:", "PHOTOS :", tweet)
tweet =re.sub(r"Time2015-08-05", "Time 2015-08-05", tweet)
tweet =re.sub(r"Denmark:", "Denmark :", tweet)
tweet =re.sub(r"Articles:", "Articles :", tweet)
tweet =re.sub(r"Crash:", "Crash :", tweet)
tweet =re.sub(r"casualties.:", "casualties .:", tweet)
tweet =re.sub(r"Afghanistan:", "Afghanistan :", tweet)
tweet =re.sub(r"Day:", "Day :", tweet)
tweet =re.sub(r"AVERTED:", "AVERTED :", tweet)
tweet =re.sub(r"sitting:", "sitting :", tweet)
tweet =re.sub(r"Multiplayer:", "Multiplayer :", tweet)
tweet =re.sub(r"Kaduna:", "Kaduna :", tweet)
tweet =re.sub(r"favorite:", "favorite :", tweet)
tweet =re.sub(r"home:", "home :", tweet)
tweet =re.sub(r"just:", "just :", tweet)
tweet =re.sub(r"Collision-1141", "Collision - 1141", tweet)
tweet =re.sub(r"County:", "County :", tweet)
tweet =re.sub(r"Duty:", "Duty :", tweet)
tweet =re.sub(r"page:", "page :", tweet)
tweet =re.sub(r"Attack:", "Attack :", tweet)
tweet =re.sub(r"Minecraft:", "Minecraft :", tweet)
tweet =re.sub(r"wounds;", "wounds ;", tweet)
tweet =re.sub(r"Shots:", "Shots :", tweet)
tweet =re.sub(r"shots:", "shots :", tweet)
tweet =re.sub(r"Gunfire:", "Gunfire :", tweet)
tweet =re.sub(r"hike:", "hike :", tweet)
tweet =re.sub(r"Email:", "Email :", tweet)
tweet =re.sub(r"System:", "System :", tweet)
tweet =re.sub(r"Radio:", "Radio :", tweet)
tweet =re.sub(r"King:", "King :", tweet)
tweet =re.sub(r"upheaval:", "upheaval :", tweet)
tweet =re.sub(r"tragedy;", "tragedy ;", tweet)
tweet =re.sub(r"HERE:", "HERE :", tweet)
tweet =re.sub(r"terrorism:", "terrorism :", tweet)
tweet =re.sub(r"police:", "police :", tweet)
tweet =re.sub(r"Mosque:", "Mosque :", tweet)
tweet =re.sub(r"Rightways:", "Rightways :", tweet)
tweet =re.sub(r"Brooklyn:", "Brooklyn :", tweet)

```

```

tweet =re.sub(r"Arrived:", "Arrived :", tweet)
tweet =re.sub(r"Home:", "Home :", tweet)
tweet =re.sub(r"Earth:", "Earth :", tweet)
tweet =re.sub(r"three:", "three :", tweet)

tweet =re.sub(r"IranDeal", "Iran Deal", tweet)
tweet =re.sub(r"ArianaGrande", "Ariana Grande", tweet)
tweet =re.sub(r"camilacabello97", "camila cabello", tweet)
tweet =re.sub(r"RondaRousey", "Ronda Rousey", tweet)
tweet =re.sub(r"MTVHottest", "MTV Hottest", tweet)
tweet =re.sub(r"TrapMusic", "Trap Music", tweet)
tweet =re.sub(r"ProphetMuhammad", "Prophet Muhammad", tweet)
tweet =re.sub(r"PantherAttack", "Panther Attack", tweet)
tweet =re.sub(r"StrategicPatience", "Strategic Patience", tweet)
tweet =re.sub(r"socialnews", "social news", tweet)
tweet =re.sub(r"NASAHurricane", "NASA Hurricane", tweet)
tweet =re.sub(r"onlinecommunities", "online communities", tweet)
tweet =re.sub(r"humanconsumption", "human consumption", tweet)
tweet =re.sub(r"Typhoon-Devastated", "Typhoon Devastated", tweet)
tweet =re.sub(r"Meat-Loving", "Meat Loving", tweet)
tweet =re.sub(r"facialabuse", "facial abuse", tweet)
tweet =re.sub(r"LakeCounty", "Lake County", tweet)
tweet =re.sub(r"BeingAuthor", "Being Author", tweet)
tweet =re.sub(r"withheavenly", "with heavenly", tweet)
tweet =re.sub(r"thankU", "thank you", tweet)
tweet =re.sub(r"iTunesMusic", "iTunes Music", tweet)
tweet =re.sub(r"OffensiveContent", "Offensive Content", tweet)
tweet =re.sub(r"WorstSummerJob", "Worst Summer Job", tweet)
tweet =re.sub(r"HarryBeCareful", "Harry Be Careful", tweet)
tweet =re.sub(r"NASASolarSystem", "NASA Solar System", tweet)
tweet =re.sub(r"animalrescue", "animal rescue", tweet)
tweet =re.sub(r"KurtSchlichter", "Kurt Schlichter", tweet)
tweet =re.sub(r"aRmageddon", "armageddon", tweet)
tweet =re.sub(r"Throwingknives", "Throwing knives", tweet)
tweet =re.sub(r"GodsLove", "God's Love", tweet)
tweet =re.sub(r"bookboost", "book boost", tweet)
tweet =re.sub(r"ibooklove", "I book love", tweet)
tweet =re.sub(r"NestleIndia", "Nestle India", tweet)
tweet =re.sub(r"realDonaldTrump", "Donald Trump", tweet)
tweet =re.sub(r"DavidVonderhaar", "David Vonderhaar", tweet)
tweet =re.sub(r"CecilTheLion", "Cecil The Lion", tweet)
tweet =re.sub(r"weathernetwork", "weather network", tweet)
tweet =re.sub(r"withBioterrorism&use", "with Bioterrorism & use", tweet)
tweet =re.sub(r"Hostage&2", "Hostage & 2", tweet)

```

```

tweet =re.sub(r"GOPDebate", "GOP Debate", tweet)
tweet =re.sub(r"RickPerry", "Rick Perry", tweet)
tweet =re.sub(r"frontpage", "front page", tweet)
tweet =re.sub(r"NewsInTweets", "News In Tweets", tweet)
tweet =re.sub(r"ViralSpell", "Viral Spell", tweet)
tweet =re.sub(r"til_now", "until now", tweet)
tweet =re.sub(r"volcanoinRussia", "volcano in Russia", tweet)
tweet =re.sub(r"ZippedNews", "Zipped News", tweet)
tweet =re.sub(r"MicheleBachman", "Michele Bachman", tweet)
tweet =re.sub(r"53inch", "53 inch", tweet)
tweet =re.sub(r"KerrickTrial", "Kerrick Trial", tweet)
tweet =re.sub(r"abstorm", "Alberta Storm", tweet)
tweet =re.sub(r"Beyhive", "Beyonce hive", tweet)
tweet =re.sub(r>IDFire", "Idaho Fire", tweet)
tweet =re.sub(r"DETECTADO", "Detected", tweet)
tweet =re.sub(r"RockyFire", "Rocky Fire", tweet)
tweet =re.sub(r"Listen/Buy", "Listen / Buy", tweet)
tweet =re.sub(r"NickCannon", "Nick Cannon", tweet)
tweet =re.sub(r"FaroeIslands", "Faroe Islands", tweet)
tweet =re.sub(r"yycstorm", "Calgary Storm", tweet)
tweet =re.sub(r"IDPs:", "Internally Displaced People :", tweet)
tweet =re.sub(r"ArtistsUnited", "Artists United", tweet)
tweet =re.sub(r"ClaytonBryant", "Clayton Bryant", tweet)
tweet =re.sub(r"jimmyfallon", "jimmy fallon", tweet)

```

```

return tweet

```

In [41]:

```

stopwords = ["they've", 'k', 'whom', "he'll", 'could', 'itself', 'hence',
'when's', 'where', 'through',
'was', 'its', 'into', 'however', "she'd", "we'd", 'they', 'below', 'again',
"she'll", "he's",
'did', 'my', 'are', 'our', "where's", 'above', 'ever', 'yourself', "i'd", 'just',
'we', "i'll",
'it', 'from', "we'll", 'that', 'he', 'cannot', "you're", 'her', 'this', "why's",
'once',
'am', 'ourselves', 'out', 'get', 'would', 'up', "it's", 'same', 'these',
"you've", 'such', 'between',
'himself', 'also', "that's", 'r', 'www', 'all', "what's", 'if', 'http', 'herself',
'after', 'had',
'has', 'your', 'while', 'other', 'their', 'shall', 'more', 'off', 'as', 'hers',
'with', 'over',
'by', 'there', "we're", "they'll", 'any', 'to', 'no', 'about', 'both', "he'd",
'only', 'here',

```

```
'than', 'what', 'been', 'does', "we've", 'theirs', 'being', 'ought', "they'd",
'few', 'you', 'under',
'since', 'can', 'them', 'at', 'else', 'each', 'ours', 'therefore', 'most',
'before', 'then', 'his', 'me',
'a', 'further', "how's", 'during', 'of', 'like', 'on', 'themselves', 'why',
'those', 'in', 'too', 'she',
'because', "i've", "let's", 'how', 'very', "you'd", 'own', 'but', "she's", 'i',
'yourselves', 'down', 'should',
'and', 'do', 'or', 'were', 'some', 'an', "who's", 'otherwise', 'be', 'him',
'myself', 'have', 'which', "there's",
'i'm", 'when', 'doing', "you'll", 'com', 'for', 'who', 'yours', 'until', 'the',
'is', 'so', "they're", "here's",
'nor', 'having', 'will', 'may', 'one', 'now']
```

In [42]:

```
defpreprocessor2(text):
    text =text.replace('%20',' ')
    text =text.lower()
    text =text.replace("n't","nt")
    text =re.sub(r"\s\w\s", "", text)
    text =re.sub(r"\s\d\s", "", text)
return text
```

In [43]:

```
tweet_ax =pd.read_csv('accident_train.csv')
test_ax =pd.read_csv('test.csv')
```

In [44]:

```
df_ax =pd.concat([tweet_ax,test_ax])
```

In [45]:

```
df_ax['location'].replace({'United States':'USA',
'New York':'USA',
"London":'UK',
"Los Angeles, CA":'USA',
"Washington, D.C.":'USA',
"California":'USA',
"Chicago, IL":'USA',
"Chicago":'USA',
"New York, NY":'USA',
"California, USA":'USA',
"FLorida":'USA',
"Nigeria":'Africa',
"Kenya":'Africa',
"Everywhere":'Worldwide',
"San Francisco":'USA',
"Florida":'USA',
"United Kingdom":'UK',
```

```

"Los Angeles":'USA',
"Toronto":'Canada',
"San Francisco, CA":'USA',
"NYC":'USA',
"Seattle":'USA',
"Earth":'Worldwide',
"Ireland":'UK',
"London, England":'UK',
"New York City":'USA',
"Texas":'USA',
"London, UK":'UK',
"Atlanta, GA":'USA',
"Mumbai":"India"},inplace=True)

```

In [46]:

```

def clean_text(text):
    text = text.lower()
    text = re.sub('[\.*?]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
df_ax['text'] = df_ax['text'].apply(lambda x: clean_text(x))

```

In [47]:

```

def remove_emoji(text):
    emoji_pattern = re.compile("[
u"\U0001F600-\U0001F64F"# emoticons
u"\U0001F300-\U0001F5FF"# symbols & pictographs
u"\U0001F680-\U0001F6FF"# transport & map symbols
u"\U0001F1E0-\U0001F1FF"# flags (iOS)
u"\U00002702-\U000027B0"
u"\U000024C2-\U0001F251"
"]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', text)
df_ax['text'] = df_ax['text'].apply(lambda x: remove_emoji(x))

```

In [48]:

```

tweet_1 = tweet_ax.text.values
test_1 = test_ax.text.values
sentiments = tweet_ax.target.values

```

In [49]:

```

word_tokenizer = Tokenizer()
word_tokenizer.fit_on_texts(tweet_1)
vocab_length = len(word_tokenizer.word_index) + 1

```

In [50]:

```
def metrics(pred_tag, y_test):
    print("F1-score: ", f1_score(pred_tag, y_test))
    print("Precision: ", precision_score(pred_tag, y_test))
    print("Recall: ", recall_score(pred_tag, y_test))
    print("Accuracy: ", accuracy_score(pred_tag, y_test))
    print("-"*50)
    print(classification_report(pred_tag, y_test))
def embed(corpus):
    return word_tokenizer.texts_to_sequences(corpus)
```

In [51]:

```
train_df['text'] = train_df['text'].apply(lambda x: remove_url(x))
```

In [52]:

```
train_df['keyword_cleaned'] = train_df['keyword'].copy().apply(lambda x :
    clean(str(x))).apply(lambda x : preprocessor2(x)).apply(lambda x: ' '.join([word
    for word in x.split() if word not in (stopwords)]))
```

In [53]:

```
train_df['location_cleaned'] = train_df['location'].copy().apply(lambda x :
    clean(str(x))).apply(lambda x : preprocessor2(x)).apply(lambda x: ' '.join([word
    for word in x.split() if word not in (stopwords)]))
```

In [54]:

```
train_df['text_cleaned'] = train_df['text'].copy().apply(lambda x :
    clean(str(x))).apply(lambda x : preprocessor2(x)).apply(lambda x: ' '.join([word
    for word in x.split() if word not in (stopwords)]))
```

In [55]:

```
test_df['text'] = test_df['text'].apply(lambda x: remove_url(x))
```

In [56]:

```
test_df['keyword_cleaned'] = test_df['keyword'].copy().apply(lambda x :
    clean(str(x))).apply(lambda x : preprocessor2(x)).apply(lambda x: ' '.join([word
    for word in x.split() if word not in (stopwords)]))
```

In [57]:

```
test_df['location_cleaned'] = test_df['location'].copy().apply(lambda x :
    clean(str(x))).apply(lambda x : preprocessor2(x)).apply(lambda x: ' '.join([word
    for word in x.split() if word not in (stopwords)]))
```

In [58]:

```
test_df['text_cleaned'] = test_df['text'].copy().apply(lambda x :
    clean(str(x))).apply(lambda x : preprocessor2(x)).apply(lambda x: ' '.join([word
    for word in x.split() if word not in (stopwords)]))
```

In [59]:

```
train_df.head()
```

Out[59]:

Unnamed: 0	id	keyword	location	text	target	keyword_cleaned	location_cleaned	text_cleaned
------------	----	---------	----------	------	--------	-----------------	------------------	--------------

Unnamed: 0	id	keyword	location	text	target	keyword_cleaned	location_cleaned	text_cleaned	
0	1040	1506	body%20bags	NaN	I love The body shop%0's bags??\n\n#cutekitten...	0	body bags	nan	love body shops bags?? #cutekitten #catsofinst...
1	1264	1822	burned	San Diego CA	Fire burning on Pendleton has burned 300 acres...	1	burned	san diego ca	fire burning pendleton burned 300 acres: smoke...
2	6464	9248	sunk	glasgow	Near them on the sand half sunk a shattered vi...	1	sunk	glasgow	near sand half sunkshattered visage lies...
3	5694	8126	rescued	Earth	Migrants Rescued After Boat Capsizes Off Libya	1	rescued	earth	migrants rescued boat capsizes libya
4	472	682	attack	portland, oregon	illegal alien released by Obama/DHS 4 times Ch...	1	attack	portland, oregon	illegal alien released obama/dhstimes charged ...

In [60]:

```
test_df.head()
```

Out[60]:

	id	keyword	location	text	keyword_cleaned	location_cleaned	text_cleaned
0	6035	heat%20wave	Brooklyn, NY	I added some dumb ideas to beat the #summer he...	heat wave	brooklyn, ny	added dumb ideas beat #summer heat cheaply @br...
1	2168	catastrophic	NaN	If a £1 rise in wages is going to have such a...	catastrophic	nan	if£1 rise wages going suchcatastrophic impact...
2	317	annihilated	NaN	How do I step outside for 5 seconds and get an...	annihilated	nan	dostep outside forseconds annihilated mosquitoes?
3	9682	tornado	Canada	Tornado warnings issued for southern Alberta	tornado	canada	tornado wamings issued southern alberta
4	5857	hailstorm	Calgary, Alberta, Canada	Get out of the hailstorm and come down to @The...	hailstorm	calgary, alberta, canada	hailstorm come @thebrokencity 2night 9pm catch...

In [61]:

```
fig, ax =plt.subplots(figsize = (8,5))
pd.value_counts(train_df['target']).plot(kind="bar")
ax.set_title('Target Count')
ax.set_ylabel('Frequency')
ax.grid(True)
plt.show()
```

In [62]:

```
loc_pos = train_df[(train_df.location_cleaned != 'nan') &
                    (train_df.location_cleaned != '') & (train_df['target']
                    ==1)][['location_cleaned']].value_counts()
```

In [63]:

```
loc_neg = train_df[(train_df.location_cleaned != 'nan') &
                    (train_df.location_cleaned != '') & (train_df['target']
                    ==0)][['location_cleaned']].value_counts()
```

In [64]:

```
loc_pos_dict = loc_pos[:20].to_dict()
```

In [65]:

```
loc_neg_dict = loc_neg[:20].to_dict()
```

In [66]:

```
names0 =list(loc_neg_dict.keys())
```

In [67]:

```
values0 =list(loc_neg_dict.values())
```

In [68]:

```
names1 =list(loc_pos_dict.keys())
```

In [69]:

```
values1 =list(loc_pos_dict.values())
```

In [70]:

```
fig, (ax1, ax2) =plt.subplots(figsize = (20,5), nrows=1, ncols=2)
```

```
ax1.bar(range(len(loc_pos_dict)),values1,tick_label=names1)
ax1.set_xticklabels(names1, rotation="vertical")
ax1.set_ylim(0, 100)
ax1.grid(True)
ax1.set_title('Location with most Pos target')
ax1.set_ylabel('Frequency')
```

```
ax2.bar(range(len(loc_neg_dict)),values0,tick_label=names0)
ax2.set_xticklabels(names0, rotation="vertical")
ax2.set_ylim(0, 100)
ax2.grid(True)
ax2.set_title('Location with most Neg target')
ax2.set_ylabel('Frequency')
```

Out[70]:

```
Text(0, 0.5, 'Frequency')
```

In [71]:

```
defpreprocessor3(text):
    text =re.sub(r'^washington d c ', "washington dc", text)
    text =re.sub(r'^washington +[\w]*', "washington dc", text)
    text =re.sub(r'^new york +[\w]*', "new york", text)
```

```

text = re.sub(r'^nyc$', "new york", text)
text = re.sub(r'^chicago +[\w]*', "chicago", text)
text = re.sub(r'^california +[\w]*', "california", text)
text = re.sub(r'^los angeles +[\w]*', "los angeles", text)
text = re.sub(r'^san francisco +[\w]*', "san francisco", text)
text = re.sub(r'^london +[\w]*', "london", text)
text = re.sub(r'^usa$', "united states", text)
text = re.sub(r'^us$', "united states", text)
text = re.sub(r'^uk$', "united kingdom", text)

```

```

return text

```

In [72]:

```

defpreprocessor4(text):
    abb = ['ak', 'al', 'az', 'ar', 'ca', 'co',
           'ct', 'de', 'dc', 'fl', 'ga', 'hi',
           'id', 'il', 'in', 'ia', 'ks', 'ky',
           'la', 'me', 'mt', 'ne', 'nv', 'nh',
           'nj', 'nm', 'ny', 'nc', 'nd', 'oh',
           'ok', 'or', 'md', 'ma', 'mi', 'mn',
           'ms', 'mo', 'pa', 'ri', 'sc', 'sd',
           'tn', 'tx', 'ut', 'vt', 'va', 'wa',
           'wv', 'wi', 'wy']

    for i in abb:
        text = re.sub(r'^{0}$'.format(i), '', text)

```

```

return text

```

In [73]:

```

train_df['location_cleaned'] = train_df['location_cleaned'].copy().apply(lambda x
: preprocessor3(x)).apply(lambda x : preprocessor4(x))

```

In [74]:

```

train_df['text_cleaned'] = train_df['text_cleaned'].copy().apply(lambda x :
preprocessor3(x)).apply(lambda x : preprocessor4(x))

```

In [75]:

```

test_df['location_cleaned'] = test_df['location_cleaned'].copy().apply(lambda x :
preprocessor3(x)).apply(lambda x : preprocessor4(x))

```

In [76]:

```

test_df['text_cleaned'] = test_df['text_cleaned'].copy().apply(lambda x :
preprocessor3(x)).apply(lambda x : preprocessor4(x))

```

In [77]:

```

loc_pos_dict = loc_pos[:].to_dict()

```

In [78]:

```

loc_neg_dict = loc_neg[:].to_dict()

```

In [79]:

```

loc_list =list(loc_pos_dict.keys()) +list(loc_neg_dict.keys())
In [80]:
unique_loc = []
In [81]:
lower_loc = []
In [82]:
for x in loc_list:
    if x notin unique_loc:
        unique_loc.append(x)
In [83]:
for x in unique_loc:
    lower_loc.append(x.lower().replace(',','').replace('.', ''))
In [84]:

classwc_base2:
def__init__(self, data):
self.temp= data.apply(lambda x: ' '.join([word for word in x.split()]))
self.text =" ".join(word for word in data)
self.wordlist= []

defplot_wc(self, mask=None, max_words=200, figure_size=(20,10), title=None,
stopwords=stopwords):

print ("There are {} words in the combination of all
review.".format(len(self.text)))

        wordcloud =WordCloud(background_color='black',
                                stopwords=stopwords,
                                max_words = max_words,
                                collocations=False,
                                random_state =10,
                                width =800,
                                height =400)

wordcloud.generate(self.text)

self.wordlist=list(wordcloud.words_.keys())

plt.figure(figsize=figure_size)
plt.imshow(wordcloud)
plt.title(title)
plt.axis("off")
In [85]:
print (stopwords)

```

```
[
    "they've", 'k', 'whom', "he'll", 'could', 'itself', 'hence', "when's", 'where',
    'through', 'was', 'its', 'into', 'however', "she'd", "we'd", 'they', 'below',
    'again', "she'll", "he's", 'did', 'my', 'are', 'our', "where's", 'above', 'ever',
    'yourself', "i'd", 'just', 'we', "i'll", 'it', 'from', "we'll", 'that', 'he',
    'cannot', "you're", 'her', 'this', "why's", 'once', 'am', 'ourselves', 'out',
    'get', 'would', 'up', "it's", 'same', 'these', "you've", 'such', 'between',
    'himself', 'also', "that's", 'r', 'www', 'all', "what's", 'if', 'http', 'herself',
    'after', 'had', 'has', 'your', 'while', 'other', 'their', 'shall', 'more', 'off',
    'as', 'hers', 'with', 'over', 'by', 'there', "we're", "they'll", 'any', 'to',
    'no', 'about', 'both', "he'd", 'only', 'here', 'than', 'what', 'been', 'does',
    "we've", 'theirs', 'being', 'ought', "they'd", 'few', 'you', 'under', 'since',
    'can', 'them', 'at', 'else', 'each', 'ours', 'therefore', 'most', 'before',
    'then', 'his', 'me', 'a', 'further', "how's", 'during', 'of', 'like', 'on',
    'themselves', 'why', 'those', 'in', 'too', 'she', 'because', "i've", "let's",
    'how', 'very', "you'd", 'own', 'but', "she's", 'i', 'yourselves', 'down',
    'should', 'and', 'do', 'or', 'were', 'some', 'an', "who's", 'otherwise', 'be',
    'him', 'myself', 'have', 'which', "there's", "i'm", 'when', 'doing', "you'll",
    'com', 'for', 'who', 'yours', 'until', 'the', 'is', 'so', "they're", "here's",
    'nor', 'having', 'will', 'may', 'one', 'now']
```

In [86]:

```
wc = wc_base2(train_df[train_df.target ==0].text_cleaned)
```

In [87]:

```
wc.plot_wc(title="Word Cloud of tweets with Negative target")
There are 286802 words in the combination of all review.
```

In [88]:

```
wc_s =set(wc.wordlist)
```

In [89]:

```
wc2 = wc_base2(train_df[train_df.target ==1].text_cleaned)
```

In [90]:

```
wc2.plot_wc(title="Word Cloud of tweets with Positive target")
There are 261685 words in the combination of all review.
```

In [91]:

```
wc2_s =set(wc2.wordlist)
```

In [92]:

```
train_df['text_cleaned'] = train_df['text_cleaned'].apply(lambda x: ' '.join([word
for word in x.split() if word notin (stopwords)]))
```

In [93]:

```
test_df['text_cleaned'] = test_df['text_cleaned'].apply(lambda x: ' '.join([word
for word in x.split() if word notin (stopwords)]))
```

In [94]:

```
train_df.text_cleaned.head()
```

Out[94]:

```
0    love body shops bags?? #cutekitten #catsofinst...
```

```

1 fire burning pendleton burned 300 acres: smoke...
2 near sand half sunkshattered visage lies...
3 migrants rescued boat capsizes libya
4 illegal alien released obama/dhstimes charged ...
Name: text_cleaned, dtype: object

```

In [95]:

```
train_df['index_no'] = train_df.index
```

In [96]:

```

train_df['sent_w_index'] = train_df['text_cleaned'] + ' ' +
train_df['index_no'].astype('str')

```

In [97]:

```
train_df.head()
```

Out[97]:

	Unnamed: 0	id	keyword	location	text	target	keyword_cleaned	location_cleaned	text_cleaned	index_no	sent_w_index
0	1040	1506	body%20bags	NaN	I love The body shop%U's bags??\n\n#cutekitten...	0	body bags	nan	love body shops bags?? #cutekitten #catsofinst...	0	love body shops bags?? #cutekitten #catsofinst...
1	1264	1822	burned	San Diego CA	Fire burning on Pendleton has burned 300 acres...	1	burned	san diego ca	fire burning pendleton burned 300 acres: smoke...	1	fire burning pendleton burned 300 acres: smoke...
2	6464	9248	sunk	glasgow	Near them on the sand half sunk a shattered vi...	1	sunk	glasgow	near sand half sunkshattered visage lies...	2	near sand half sunkshattered visage lies... 2
3	5694	8126	rescued	Earth	Migrants Rescued After Boat Capsizes Off Libya	1	rescued	earth	migrants rescued boat capsizes libya	3	migrants rescued boat capsizes libya 3
4	472	682	attack	portland, oregon	illegal alien released by Obama/DHS 4 times Ch...	1	attack	portland, oregon	illegal alien released obama/dhstimes charged ...	4	illegal alien released obama/dhstimes charged ...

In [98]:

```
train_df.sent_w_index[1]
```

Out[98]:

```
'fire burning pendleton burned 300 acres: smoke reported drifting temecula. 1'
```

In [99]:

```

def random_swap(text):
    text_list = text.split()
    seed = int(text_list[-1])

```

```

        text_list=text_list[:-1]
        text_length =len(text_list)

np.random.seed(seed)
try:
    a =np.random.randint(0, text_length,size=2)
except:
return
    #print(a)

    temp_a = text_list[a[0]]
    temp_b = text_list[a[1]]

    text_list[a[0]] = temp_b
    text_list[a[1]] = temp_a

    redo = ' '.join([str(i) for i in text_list])

return redo

```

In [100]:

```

def random_del(text):
    text_list =text.split()
    seed =int(text_list[-1])

    text_list=text_list[:-1]
    text_length =len(text_list)

np.random.seed(seed)
try:
    a =np.random.randint(0, text_length,size=1)
except:
return

    text_list.pop((a[0]))

    redo = ' '.join([str(i) for i in text_list])

return redo

```

In [101]:

```

train_df['da_text_cleaned'] = train_df['sent_w_index'].apply(lambda
x:random_swap(x))

```

In [102]:

```
train_df['da_text_cleaned2'] = train_df['sent_w_index'].apply(lambda
x:random_del(x))
```

In [103]:

```
train_df.drop(['index_no','sent_w_index'], axis=1,inplace=True)
```

In [104]:

```
train_df.head()
```

Out[104]:

	Unnamed: 0	id	keyword	location	text	target	keyword_cleaned	location_cleaned	text_cleaned	da_text_cleaned	da_text_cleaned2
0	1040	1506	body%20bags	NaN	I love The body shop%Ū's bags??\n\n#cutekitten...	0	body bags	nan	love body shops bags?? #cutekitten #catsofinst...	#catsofinstagr am body shops bags?? #cutekitten...	love body shops bags?? #cutekitten #summerinsw. ..
1	1264	1822	burned	San Diego CA	Fire burning on Pendleton has burned 300 acres...	1	burned	san diego ca	fire burning pendleton burned 300 acres: smoke...	fire burning pendleton burned 300 drifting smo...	fire burning pendleton burned 300 smoke report...
2	6464	9248	sunk	glasgow	Near them on the sand half sunk a shattered vi...	1	sunk	glasgow	near sand half sunkshattered visage lies...	lies... sand half sunkshattered visage near	sand half sunkshattered visage lies...
3	5694	8126	rescued	Earth	Migrants Rescued After Boat Capsizes Off Libya	1	rescued	earth	migrants rescued boat capsizes libya	boat rescued migrants capsizes libya	migrants rescued capsizes libya
4	472	682	attack	portland, oregon	illegal alien released by Obama/DHS 4 times Ch...	1	attack	portland, oregon	illegal alien released obama/dhstimes charged ...	illegal alien released obama/dhstimes charged ...	illegal alien released obama/dhstimes charged ...

In [105]:

```
test_df =
test_df[['id','keyword_cleaned','location_cleaned','text_cleaned']].copy()
```

In [106]:

```
state =1
```

In [107]:

```
train_df = train_df.sample(frac=1,random_state=state)
```

In [108]:

```
train_df.reset_index(inplace=True, drop=True)
```

In [109]:

```
train_df['text_cleaned'] = train_df['text_cleaned'].apply(lambda x : str(x))
```

In [110]:

```
train_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7728 entries, 0 to 7727
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
-----
0   Unnamed: 0            7728 non-null  int64
1   id                    7728 non-null  int64
2   keyword               7667 non-null  object
3   location              5195 non-null  object
4   text                  7728 non-null  object
5   target               7728 non-null  int64
6   keyword_cleaned       7728 non-null  object
7   location_cleaned      7728 non-null  object
8   text_cleaned          7728 non-null  object
9   da_text_cleaned       7728 non-null  object
10  da_text_cleaned2      7728 non-null  object
dtypes: int64(3), object(8)
memory usage: 664.2+ KB

```

In [111]:

```
top_word = 35000
```

In [112]:

```
text_lengths = [len(x.split()) for x in (train_df.text_cleaned)]
```

In [113]:

```
tok = Tokenizer(num_words=top_word)
```

In [114]:

```
tok.fit_on_texts((train_df['text_cleaned']+train_df['keyword_cleaned']+train_df['location_cleaned']))
```

In [115]:

```
max_words = max(text_lengths) + 1
```

In [116]:

```
max_words_ky = max([len(x.split()) for x in (train_df.keyword_cleaned)]) + 1
```

In [117]:

```
max_words_lc = max([len(x.split()) for x in (train_df.location_cleaned)]) + 1
```

In [118]:

```
print("top_word: ", str(top_word))
```

```
top_word: 35000
```

In [119]:

```
print("max_words: ", str(max_words))
```

```
max_words: 33
```

In [120]:

```
print("max_words_ky: ", str(max_words_ky))
```

```
max_words_ky: 3
```

In [121]:

```
print("max_words_lc: ", str(max_words_lc))
```

```

max_words_lc: 11

plt.hist(text_lengths, bins=25)
plt.title('Histogram of # of Words in Texts')

Text(0.5, 1.0, 'Histogram of # of Words in Texts')

X_train_tx = tok.texts_to_sequences(train_df['text_cleaned'])
X_train_ky = tok.texts_to_sequences(train_df['keyword_cleaned'])
X_train_lc = tok.texts_to_sequences(train_df['location_cleaned'])
X_test_tx = tok.texts_to_sequences(test_df['text_cleaned'])
X_test_ky = tok.texts_to_sequences(test_df['keyword_cleaned'])
X_test_lc = tok.texts_to_sequences(test_df['location_cleaned'])
Y_train = train_df['target']

print('Found %s unique tokens.' % len(tok.word_index))
Found 25928 unique tokens.

Y_train = to_categorical(Y_train)
print("Y_train.shape: ", Y_train.shape)
Y_train.shape: (7728, 2)

X_train_tx = sequence.pad_sequences(X_train_tx, maxlen=max_words)
X_train_ky = sequence.pad_sequences(X_train_ky, maxlen=max_words_ky)
X_train_lc = sequence.pad_sequences(X_train_lc, maxlen=max_words_lc)
X_test_tx = sequence.pad_sequences(X_test_tx, maxlen=max_words)
X_test_ky = sequence.pad_sequences(X_test_ky, maxlen=max_words_ky)
X_test_lc = sequence.pad_sequences(X_test_lc, maxlen=max_words_lc)

print("X_train_tx.shape: ", X_train_tx.shape)
X_train_tx.shape: (7728, 33)

print("X_train_ky.shape: ", X_train_ky.shape)

```

```

X_train_ky.shape: (7728, 3)
In [140]:
print("X_train_lc.shape: ", X_train_lc.shape)
X_train_lc.shape: (7728, 11)
In [141]:
embeddings_dictionary =dict()
In [142]:
glove_file =open('glove.6B.300d.txt', encoding="utf8")
In [ ]:
for line in glove_file:
    records =line.split()
    word =records[0]
    vector_dimensions = asarray(records[1:], dtype='float32')

    embeddings_dictionary [word] = vector_dimensions
In [ ]:
glove_file.close()
In [ ]:
embedding_dim =300
In [ ]:
embedding_matrix =zeros((top_word, embedding_dim))
In [ ]:
for word, index in tok.word_index.items():
    embedding_vector = embeddings_dictionary.get(word)
if embedding_vector isnotNone:
    embedding_matrix[index] = embedding_vector
In [ ]:
input1 = Input(shape=(max_words,))
In [ ]:
embedding_layer1 =Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=max_words, trainable=False)(input1)
In [ ]:
dropout1 = Dropout(0.2)(embedding_layer1)
In [ ]:
lstm1_1 =LSTM(128,return_sequences =True)(dropout1)
In [ ]:
lstm1_2 =LSTM(128,return_sequences =True)(lstm1_1)
In [ ]:
lstm1_2a =LSTM(128,return_sequences =True)(lstm1_2)
In [ ]:
lstm1_3 = LSTM(128)(lstm1_2a)
In [ ]:
input2 = Input(shape=(max_words_ky,))
In [ ]:

```

```

embedding_layer2 =Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=max_words_ky, trainable=False)(input2)

dropout2 = Dropout(0.2)(embedding_layer2)

lstm2_1 =LSTM(64,return_sequences =True)(dropout2)

lstm2_2 =LSTM(64,return_sequences =True)(lstm2_1)

lstm2_3 = LSTM(64)(lstm2_2)

input3 = Input(shape=(max_words_lc,))

embedding_layer3 =Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=max_words_lc, trainable=False)(input3)

dropout3 = Dropout(0.2)(embedding_layer3)

lstm3_1 =LSTM(32,return_sequences =True)(dropout3)

lstm3_2 =LSTM(32,return_sequences =True)(lstm3_1)

lstm3_3 = LSTM(32)(lstm3_2)

merge =concatenate([lstm1_3, lstm2_3,lstm3_3])

dropout = Dropout(0.8)(merge)

dense1 =Dense(256, activation='relu')(dropout)

dense2 =Dense(128, activation='relu')(dense1)

output =Dense(2, activation='softmax')(dense2)

model1 = Model(inputs=[input1,input2,input3], outputs=output)

model1.summary()

```

LSTM & CNN

```

input1 = Input(shape=(max_words,))

embedding_layer1 =Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=max_words, trainable=False)(input1)

```

```

lstm1_1 = Bidirectional(LSTM(100, return_sequences=True, dropout
=0.2))(embedding_layer1)
In [ ]:
lstm1_1a = Bidirectional(LSTM(100, return_sequences=True, dropout =0.2))(lstm1_1)
In [ ]:
lstm1_1b = Bidirectional(LSTM(100, return_sequences=True, dropout =0.2))(lstm1_1a)
In [ ]:
res = Reshape((-1, X_train_tx.shape[1], 100))(lstm1_1b)
In [ ]:
conv1 = Conv2D(100, (3,3), padding='same', activation="relu")(res)
In [ ]:
pool1 = MaxPooling2D(pool_size=(2,2))(conv1)
In [ ]:
flat1 = Flatten()(pool1)
In [ ]:
input2 = Input(shape=(max_words_ky,))
In [ ]:
embedding_layer2 = Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=max_words_ky, trainable=False)(input2)
In [ ]:
lstm2_1 = Bidirectional(LSTM(100, return_sequences=True, dropout
=0.2))(embedding_layer2)
In [ ]:
lstm2_1a = Bidirectional(LSTM(100, return_sequences=True, dropout =0.2))(lstm2_1)
In [ ]:
lstm2_1b = Bidirectional(LSTM(100, return_sequences=True, dropout =0.2))(lstm2_1a)
In [ ]:
res2 = Reshape((-1, X_train_ky.shape[1], 100))(lstm2_1b)
In [ ]:
conv2 = Conv2D(100, (3,3), padding='same', activation="relu")(res2)
In [ ]:
pool2 = MaxPooling2D(pool_size=(2,2))(conv2)
In [ ]:
flat2 = Flatten()(pool2)
In [ ]:
input3 = Input(shape=(max_words_lc,))
In [ ]:
embedding_layer3 = Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=max_words_lc, trainable=False)(input3)
In [ ]:
lstm3_1 = Bidirectional(LSTM(100, return_sequences=True, dropout
=0.2))(embedding_layer3)
In [ ]:
lstm3_1a = Bidirectional(LSTM(100, return_sequences=True, dropout =0.2))(lstm3_1)

```

```

lstm3_1b = Bidirectional(LSTM(100, return_sequences=True, dropout = 0.2))(lstm3_1a)
res3 = Reshape((-1, X_train_lc.shape[1], 100))(lstm3_1b)
conv3 = Conv2D(100, (3,3), padding='same', activation="relu")(res3)
pool3 = MaxPooling2D(pool_size=(2,2))(conv3)
flat3 = Flatten()(pool3)
merge = concatenate([flat1, flat2, flat3])
dropout = Dropout(0.4)(merge)
dense1 = Dense(256, activation='relu')(dropout)
dense2 = Dense(128, activation='relu')(dense1)
output = Dense(2, activation='softmax')(dense2)
model2 = Model(inputs=[input1, input2, input3], outputs=output)
model2.summary()
input1 = Input(shape=(max_words,))
input2 = Input(shape=(max_words_ky,))
input3 = Input(shape=(max_words_lc,))
merge = concatenate([input1, input2, input3])
embedding_layer1 = Embedding(top_word, embedding_dim, weights=[embedding_matrix],
input_length=42, trainable=False)(merge)
lstm1_1 = Bidirectional(LSTM(128, return_sequences=True, dropout
=0.2))(embedding_layer1)
lstm1_1a = Bidirectional(LSTM(128, return_sequences=True, dropout = 0.2))(lstm1_1)
lstm1_1b = Bidirectional(LSTM(128, return_sequences=True, dropout = 0.2))(lstm1_1a)
conv2 = Conv1D(64, 3, padding='same', activation="relu")(lstm1_1b)

```

```

pool2 = MaxPooling1D(pool_size=2) (conv2)
conv3 = Conv1D(64, 3, padding='same', activation="relu") (pool2)
pool3 = MaxPooling1D(pool_size=2) (conv3)
flat2 = Flatten() (pool3)
dense1 =Dense(256, activation='relu') (flat2)
dropout = Dropout(0.8) (dense1)
dense2 =Dense(128, activation='relu') (dropout)
output =Dense(2, activation='softmax') (dense2)
model3 = Model(inputs=[input1,input2,input3], outputs=output)
model3.summary()

optimizer1 =Adam(lr =.0001, beta_1 =.9, beta_2 =.999, epsilon =1e-10, decay =.0,
amsgrad =False)

optimizer2 =Adam(lr =.0001, beta_1 =.9, beta_2 =.999, epsilon =1e-10, decay =.0,
amsgrad =False)

longest_train =max(tweet_1, key=lambda sentence: len(word_tokenize(sentence)))
length_long_sentence =len(word_tokenize(longest_train))
padded_sentences = pad_sequences(embed(tweet_1), length_long_sentence,
padding='post')
test_sentences = pad_sequences(
    embed(test_1),
    length_long_sentence,
    padding='post'
)

embeddings_dictionary =dict()
embedding_dim =100
glove_file =open('glove.6B.100d.txt', encoding="utf8")
for line in glove_file:
    records =line.split()
    word =records[0]
    vector_dimensions =np.asarray(records[1:], dtype='float32')

```

```

        embeddings_dictionary[word] = vector_dimensions
glove_file.close()

```

In []:

```

embedding_matrix = np.zeros((vocab_length, embedding_dim))
for word, index in word_tokenizer.word_index.items():
    embedding_vector = embeddings_dictionary.get(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector

```

In []:

```

X1_train, X1_test, y1_train, y1_test = train_test_split(
    padded_sentences,
    sentiments,
    test_size=0.25
)

```

In []:

```

def CNN_LSTM():
    model = Sequential()
    model.add(Embedding(input_dim=embedding_matrix.shape[0],
                        output_dim=embedding_matrix.shape[1],
                        weights = [embedding_matrix],
                        input_length=length_long_sentence))
    model.add(Bidirectional(LSTM(length_long_sentence, return_sequences = True,
                                recurrent_dropout=0.2)))
    model.add(GlobalMaxPool1D())
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(length_long_sentence, activation = "relu"))
    model.add(Dropout(0.5))
    model.add(Dense(length_long_sentence, activation = "relu"))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation = 'sigmoid'))
    model.compile(optimizer='rmsprop', loss='binary_crossentropy',
                  metrics=['accuracy'])
    return model

```

In []:

```

model_ax = CNN_LSTM()
checkpoint = ModelCheckpoint(
    'model.h5',
    monitor = 'val_loss',
    verbose = 1,
    save_best_only = True
)
reduce_lr = ReduceLROnPlateau(
    monitor = 'val_loss',

```

```

        factor =0.2,
        verbose =1,
        patience =5,
        min_lr =0.001
    )
    history =model_ax.fit(
        X1_train,
        y1_train,
        epochs =1,
        batch_size =256,
        validation_data = [X1_test, y1_test],
        verbose =1,
        callbacks = [reduce_lr, checkpoint]
    )

model1.compile(loss="binary_crossentropy", optimizer=optimizer1,
               metrics=["acc"])

model2.compile(loss="binary_crossentropy", optimizer=optimizer2,
               metrics=["acc"])

es =EarlyStopping(monitor='val_loss', mode='min',verbose=1, patience =4)

learning_rate_reduction =ReduceLRonPlateau(monitor ='val_accuracy', patience =2,
verbose =1,
                                           factor =0.5, min_lr =1e-8, cooldown=1)

history = model2.fit([X_train_tx,X_train_ky,X_train_lc], Y_train,
                    validation_split=0.2, epochs=1, batch_size=16, verbose=2, callbacks=[es,
                    learning_rate_reduction])

history2 = model1.fit([X_train_tx,X_train_ky,X_train_lc], Y_train,
                    validation_split=0.2, epochs=1, batch_size=16, verbose=2, callbacks=[es,
                    learning_rate_reduction])

defresult_eva (loss,val_loss,acc,val_acc):
importmatplotlib.pyplotasplt
%matplotlib inline
    epochs =range(1,len(loss)+1)
    plt.plot(epochs, loss, 'b-o', label ='Training Loss')
    plt.plot(epochs, val_loss, 'r-o', label ='Validation Loss')
    plt.title("Training and Validation Loss")
    plt.xlabel('Epochs')

```

```

plt.ylabel('Loss')
plt.legend()
plt.show()

epochs =range(1, len(acc)+1)
plt.plot(epochs, acc, "b-o", label="Training Acc")
plt.plot(epochs, val_acc, "r-o", label="Validation Acc")
plt.title("Training and Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
In [ ]:

result_eva(history.history['loss'], history.history['val_loss'],
history.history['acc'], history.history['val_acc'])
In [ ]:

result_eva(history2.history['loss'], history2.history['val_loss'],
history2.history['acc'], history2.history['val_acc'])
In [ ]:

model2.save('accident_model.h5')
In [ ]:

model1.save('accident_model_2.h5')
In [ ]:

model = load_model('accident_model.h5')
In [ ]:

Y_pred =model.predict([X_test_tx,X_test_ky,X_test_lc], batch_size=64, verbose=2)
In [ ]:

Y_pred =np.argmax(Y_pred,axis=1)
In [ ]:

pred_df =pd.DataFrame(Y_pred, columns=['target'])
In [ ]:

result =pd.concat([test_df,pred_df], axis=1, join='outer', ignore_index=False,
keys=None, sort =False)
In [ ]:

result = result[['id','target']]
In [ ]:

print(Y_pred)
In [ ]:

result.to_csv('sample1.csv',index=False)
In [ ]:

Y_pred = model1.predict([X_test_tx,X_test_ky,X_test_lc], batch_size=16, verbose=2)
In [ ]:

Y_pred =np.argmax(Y_pred,axis=1)
In [ ]:

```

```

pred_df =pd.DataFrame(Y_pred, columns=['target'])
In [ ]:
result =pd.concat([test_df,pred_df], axis=1, join='outer', ignore_index=False,
keys=None, sort =False)
In [ ]:
result = result[['id','target']]
In [ ]:
print(Y_pred)
In [ ]:
result.to_csv('sample2.csv',index=False)
In [ ]:
fromsklearn.metricsimport (
    precision_score,
    recall_score,
    f1_score,
    classification_report,
    accuracy_score
)
In [ ]:
preds = model_ax.predict_classes(X1_test)
metrics(preds, y1_test)

```