DISASTER TWEET CLASSIFICATION USING LSTM & SIMPLE RNN

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Under the Guidance of

Dr.Priyanka Singh

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[12,2022]

Certificate

Date: 16-Nov-22

This is to certify that the work present in this Project entitled "**DISASTER TWEET CLASSIFICATION USING LSTM & SIMPLE RNN**" has been carried out by

name under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in the **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Acknowledgment

We would like to thank **Dr.Priyanka Singh** for her constant support and guidance during this research period. Her mentoring Inspired us to learn new emerging concepts of Machine learning & Data science and apply them to real data.

Table of Contents

Abstract

Disasters are very regular on earth, due to many reasons. This model that was developed uses the data from twitter and predict the whether the tweet is correct or not, if so can we extract some information and send them much earlier to the disaster rescue teams.

In this we have preprocessed the data taken, so as to fit that into the model we have choose. we have chosen LSTM as our important classifier which have resulted us with 76% accuracy on the testing data. we have also used Simple RNN model on the preprocessed data which have yielded us with 74% accuracy on the testing data.

The future work of this remains, improving accuracy and developing an API that notifies Disaster Rescue teams much earlier from the tweets that people were expressing through their twitter handles regionally.

List of Tables

List of Figures

1. Introduction

Disasters are a relatively regular occurrence in modern life. Due to urbanization, rising population, environmental degradation, and other factors, catastrophe losses have exhibited increasing patterns in human fatalities and property destruction worldwide. In these situations, not only has the immediate harm to society been enormous, but it is also harder to predict how the economy will fare in the future.

It is impossible to eliminate the risks of disaster, but it is feasible to lessen their effects with proper planning and safeguards [1]. The same is achieved via disaster management. It goes through several phases, including prevention, mitigation, readiness, reaction, and recovery. Time is precious in catastrophe situations.

Therefore, several processes, such as evacuating affected individuals, identifying affected areas and dispersing relief supplies, obtaining pertinent information from various sources, and disseminating that information among the appropriate parties, should run concurrently to save time. Taken together, these processes make up a sizable task. Making judgments on where to focus your attention during the phases above always benefits from having relevant information. Finding trustworthy, accurate, and fast geospatial data is complex, particularly when a crisis is developing quickly.

Twitter, a microblogging platform, is the biggest of the several social networking sites. In recent years, Twitter has grown in popularity as a communication tool. A Twitter user posts views, images, and news. Twitter data analysis may be used to help in disaster recovery.

2. Related Works

In this Research Project, we have worked with 2 RNN Models in Deep Learning

1. **LSTM**
2. **Simple RNN**

2.1 LSTM

Deep learning uses long short-term memory networks or LSTMs. In tasks involving sequence prediction, many recurrent neural networks (RNNs) are able to learn long-term dependencies. In addition to single input points like photographs, LSTM[1] has feedback links that let it process the whole data stream. This may be used, among other things, for speech recognition and machine translation. A special type of RNN known as LSTM performs exceptionally well across various problems.

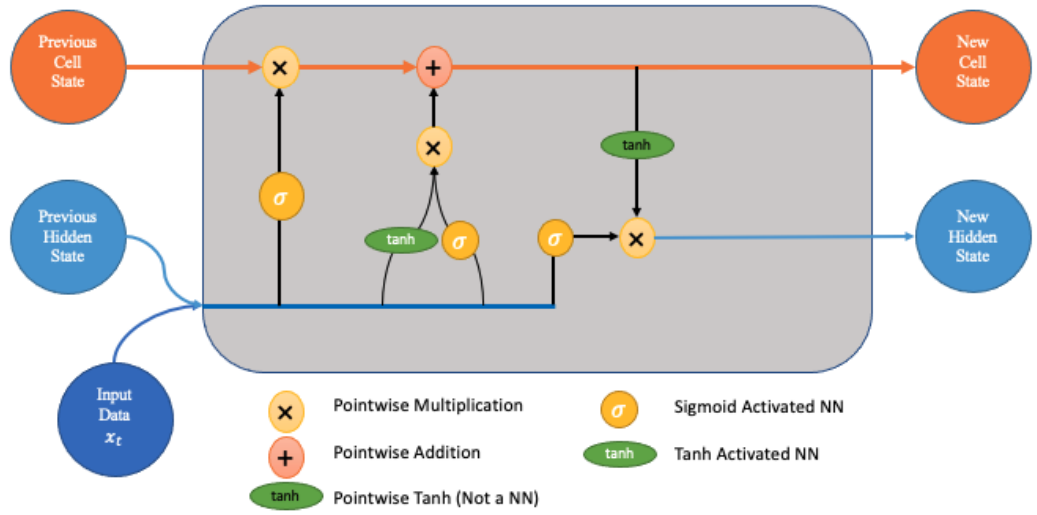

Fig-1: LSTM Cell in Detail explanation

LSTM networks find useful applications in the following areas:

- Language modeling
- Machine translation
- Handwriting recognition

2.2 Simple RNN

In Keras, the whole RNN layer[2] is provided as the SimpleRNN class. The Keras implementation is considerably different from the proposed design in many papers, but it is still straightforward. One data input and one hidden state are transmitted from one one-time step to the next in each RNN cell.

The RNN cell looks like this:

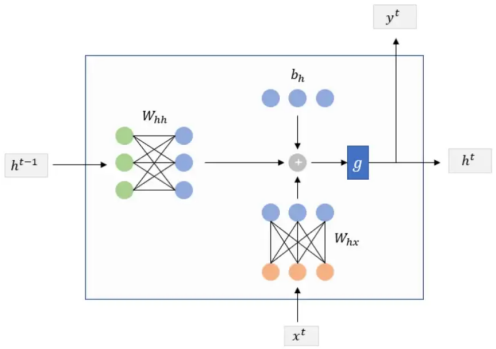

Fig-2: RNN cell in Detail explanation

3. Experiments and Results

3.1. Data Description:

We have taken from Kaggle, especially from the "natural language processing - Disaster management "competition[3].

It actually consists of train.csv for training the model and test.csv for testing the trained model.

Each sample in the train and test set has the following information:

- The text of a tweet that was collected
- A keyword in that tweet (although this may be blank!)
- The location of the tweet, from where the tweet was sent [it can even be null].

few data from train.csv

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

Table-1: Input Training Data

few data from test.csv

	id	keyword	location	text
0	0	NaN	NaN	Just happened a terrible car crash
1	2	NaN	NaN	Heard about #earthquake is different cities, s...
2	3	NaN	NaN	there is a forest fire at spot pond, geese are...
3	9	NaN	NaN	Apocalypse lighting. #Spokane #wildfires
4	11	NaN	NaN	Typhoon Soudelor kills 28 in China and Taiwan

Table-2: Input Testing Data

3.2 Tweet Classification Process:

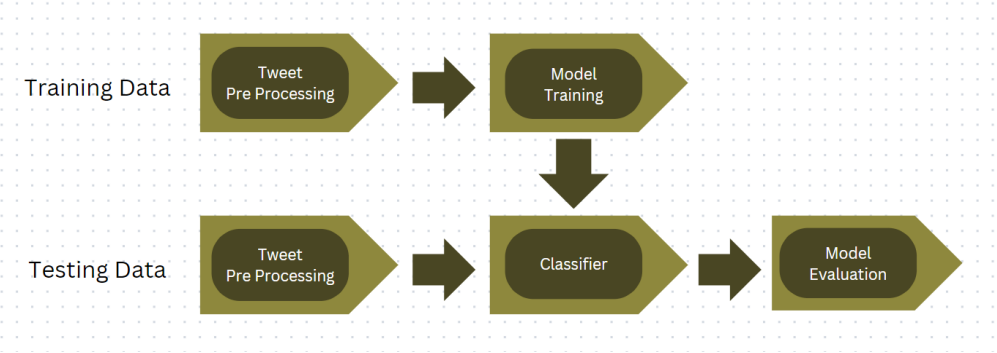

Fig-3: Tweet Classification Process.

(i)Tweets are brief, diverse, and chaotic. A tweet, which is just 140 characters long, presents several difficulties for standard NLP, including the fact that tweets are frequently miswritten and contain several grammatical and punctuation problems. It may occasionally be written in various regional tongues.

(ii)Articles and auxiliary verbs are commonly left out of tweets. They usually include acronyms, brief hand signals, regional slang, Etc. This kind of text data is frequently tricky for conventional NLP parsers. There is seldom any lexical duplication in tweets. The steps for classifying tweets are described.

3.3 Preprocessing Of Tweets:

Removing stop words:

Stop words are often the most frequently used terms in a language with only sentence-level importance. There currently needs to be a comprehensive list of stop words used by NLP technologies.

"A," "the," "is," "are," "which," Etc., are examples of stop words.

Removing URLs:

In this phase URLs that were present in the tweet are removed.

Removing Emojis

There are 2 ways of handling emojis because they express an emotion in the tweet. those are

1. Removing them from the tweet
2. Replacing the emoji with the related word.

In this model Implementation, we have followed the 1st method.

Removing Punctuations

Depending on the use case, we must carefully select the list of punctuation that we will ignore. The string module in Python, for instance, has the following list of punctuation.

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

here in our case we have considered the same function and removed punctuations.

Tokenization

is the process of breaking up text and unstructured data into units that can be treated as separate items. Token occurrences in a document can be used directly as a vector to represent the document. This quickly converts text or an unstructured string into a numerical data format which is appropriate for machine learning.

Relabeling the POS

The goal of part-of-speech tagging (POS tagging) is to determine a word's grammatical grouping depending on the context, such as whether it is a noun, adjective, verb, adverb, etc.

Each word in a sentence is given the proper tag after relationships within the phrase are looking for.

Lemmatization

In Natural Language Processing (NLP), a text normalization technique called lemmatization is used to convert any sort of word to its basic root mode. Lemmatization is responsible for combining several word inflections with the same meaning into their root forms.

Data before and after preprocessing

Training data set:

	id	keyword	location	text	target	lemma_str
7608	10869	NaN	NaN	Two giant cranes holding a bridge collapse int...	1	two giant crane hold bridge collapse nearby home
7609	10870	NaN	NaN	@aria_ahrary @TheTawniest The out of control w...	1	ariaahrary thetawniest control wild fire calif...
7610	10871	NaN	NaN	M1.94 [01:04 UTC]?5km S of Volcano Hawaii. htt...	1	m194 0104 utc5km volcano hawaii
7611	10872	NaN	NaN	Police investigating after an e-bike collided ...	1	police investigate ebike collide car little po...
7612	10873	NaN	NaN	The Latest: More Homes Razed by Northern Calif...	1	late home raze northern california wildfire ab...

Tab-3: Train Data before and after preprocessing

Testing Data set:

	id	keyword	location	text	lemma_str
3258	10861	NaN	NaN	EARTHQUAKE SAFETY LOS ANGELES □ÛÒ SAFETY FASTE...	earthquake safety los angeles safety fastener ...
3259	10865	NaN	NaN	Storm in RI worse than last hurricane. My city...	storm ri bad last hurricane cityamp3others har...
3260	10868	NaN	NaN	Green Line derailment in Chicago http://t.co/U...	green line derailment chicago
3261	10874	NaN	NaN	MEG issues Hazardous Weather Outlook (HWO) htt...	meg issue hazardous weather outlook hwo
3262	10875	NaN	NaN	#CityofCalgary has activated its Municipal Eme...	cityofcalgary activate municipal emergency pla...

Tab-4: Test Data before and after preprocessing

3.4 Data Visualization:

Here we can figure out that 0's are the majority in the Training data set

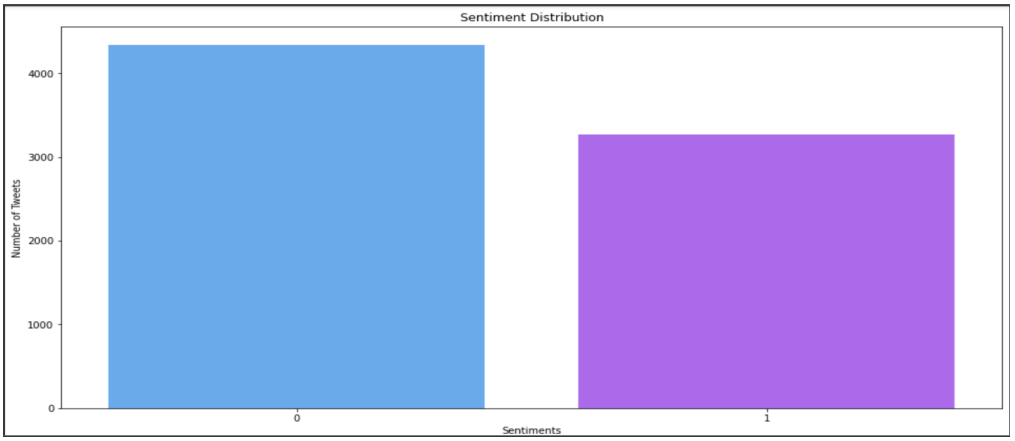

Fig-4: Overview of train data set targets

3.5. Model

Here we have Implemented 2 Recurrent Neural Network Models on the preprocessed data. Those are

1. LSTM [Long short-term memory]
2. Simple RNN

3.5.1 LSTM [Long short-term memory]:

Below mentioned Layers were used while Implementing this model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 20, 32)            96000

 dropout (Dropout)           (None, 20, 32)            0

 lstm (LSTM)                 (None, 32)                8320

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 104,353
Trainable params: 104,353
Non-trainable params: 0
_________________________________________________________________
None
```

Fig-5: Layers of LSTM

3.5.2 Simple RNN:

Below mentioned Layers were used while Implementing this model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 20, 32)            192800

 simple_rnn (SimpleRNN)      (None, 32)                2080

 dense (Dense)               (None, 10)                330

 dense_1 (Dense)             (None, 1)                 11

=================================================================
Total params: 195,221
Trainable params: 195,221
Non-trainable params: 0
_________________________________________________________________
```

Fig-6: Layers of Simple RNN

3.5.3 Model Evaluation

Each model was run for 100 Epochs

3.5.3.1 LSTM RESULTS

Training Data Evaluation Results:

```
Train Accuracy  is  :  0.9204353537249015
Train Recall  is    :  0.8825065274151436
Train Precision  is :  0.9294225481209899
Train F1 Score  is :  0.9053571428571429
```

Fig-7: Results of Training Data for LSTM

Testing Data Evaluation Results:

```
Test Accuracy  is  :  0.760507880910683
Test Recall  is    :  0.7153134635149023
Test Precision  is :  0.7204968944099379
Test F1 Score  is :  0.7178958225889633
```

Fig-8: Results of Testing Data for LSTM

3.5.3.2 Simple RNN RESULTS

Training Data Evaluation Results:

```
Train Accuracy  is  :  0.9810471007693751
Train Recall  is    :  0.9712793733681462
Train Precision  is :  0.984561093956771
Train F1 Score  is :  0.9778751369112815
```

Fig-9: Results of Training Data for Simple RNN

Testing Data Evaluation Results:

```
Test Accuracy  is  :  0.7412434325744308
Test Recall  is    :  0.7358684480986639
Test Precision  is :  0.6819047619047619
Test F1 Score  is :  0.707859614434009
```

Fig-10: Results of Testing Data for Simple RNN

3.5.4 Overall Evaluation of 2 Models

Model	Precision	Recall	F1-Score	Accuracy
LSTM	0.7204	0.7153	0.7178	76
Simple RNN	0.6819	0.7358	0.7078	74.12

Tab-5: Overall results

Methodology

The methodology we have followed during our project:

We have taken Data of tweets during disasters, not specific on purticular disaster. we have downloaded that data and implemented our further preprocessing on that data.

We have refered many articles on disaster twitter classification and came to a conclusion which model has to be implemented and what are all the methods to be applied on data before sending data inside the model.;

We have gone through sub divisions of preprocessing from various kaggle notebooks irrespective of their datasets, and by looking our data we have decided to perform certain preprocessing which are actually needed for our data.

while on the other hand for the model, by refering to many documents as we mentioned we found LSTM was predicting things, with higher accuracy. This is the reason we chosen for LSTM.

We have chosen Simple RNN as another model, to just compare the older RNN method with the newer RNN model [LSTM].

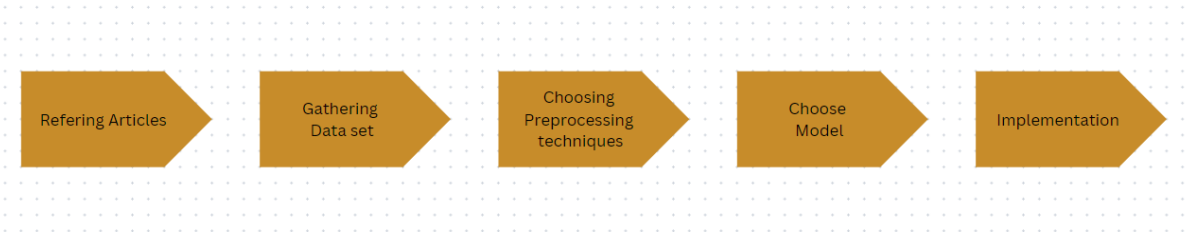

Fig-11: Methodology diagram

Discussion

We have got more test accuracy for LSTM Model than Simple RNN, which we expected.

Model	Accuracy
LSTM	76
Simple RNN	74.12

Fig-12: Final comparison

we have explored the reason why this happening, we found some of the reasons discussed below

- RNN is challenging to train since it requires long-term memory, while LSTM performs better because it includes more additional special units that can keep information for longer. The LSTM has a "memory cell" that can store data in memory for a very long time.
- The vanishing gradients or long-term dependency problem of RNNs is overcome by LSTM networks. Gradient vanishing is the term used to describe how information in a neural network is lost when connections occur more often over time. Simply said, LSTM prevents gradients from disappearing in the network by disregarding irrelevant data and information.

Conclusion

Tweets increase dramatically during emergencies. Because there are so many tweets, it is exceedingly challenging to evaluate them all and extract the pertinent data. Using techniques for machine learning, we can automatically choose a small subset of tweets from enormous data sets, making the information extraction process considerably simpler and quicker. In our study, using a smaller dataset of about 3,000 tweets, we have shown how this concept works. The goal is to gather data from tweets that will enable emergency responders during natural catastrophes to be more prepared with better preparation and less impact. In this study, we demonstrate that machine learning algorithms can extract pertinent data from social platforms that have been disrupted. Tweets are categorized using classification algorithms, and information is extracted using POS tagging, In the event of a natural disaster, we are able to get useful information from the tweets that may even be actionable. Future research has a wide range of possibilities.

Future Work

Our Future extension of this project is to make a tweet classification website.

It takes input a tweet and predicts the accuracy regarding the disasters.

References

1. LSTM:
   https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9

2. Simple RNN:
   https://towardsdatascience.com/a-practical-guide-to-rnn-and-lstm-in-keras-980f176271bc

3. Dataset References:
   https://www.kaggle.com/competitions/nlp-getting-started/data

4. Sadhukhan, S., Banerjee, S., Das, P., & Sangaiah, A. K. (2018). Producing Better Disaster Management Plans in Post-Disaster Situations Using Social Media Mining. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, 171-183. https://doi.org/10.1016/B978-0-12-813314-9.00009-8

5. Weakly Supervised and Online Learning of Word Models for Classification to Detect Disaster Reporting Tweets

   https://link.springer.com/article/10.1007/s10796-018-9830-2

6. The Real-Time Monitoring System of Social Big Data for Disaster Management
   https://link.springer.com/chapter/10.1007/978-3-662-45402-2_115