Rumour Detection

CS5803 - NLP
Project Final Presentation

Kranthi Kumar P
(AI20RESCH14002)

Sandhya A
(AI20RESCH14001)

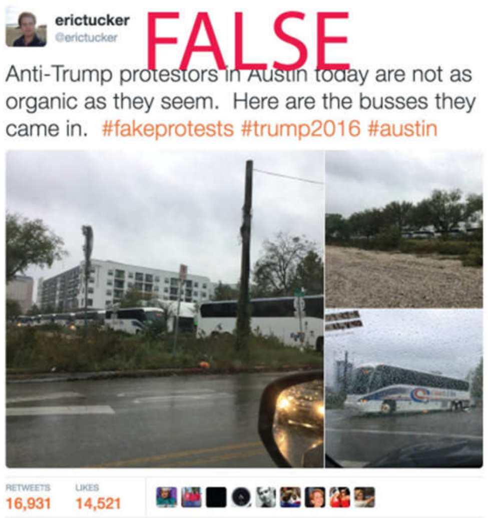
Instructors: Dr. Maunendra Desarkar & Dr. Srijith P K

Agenda

- Introduction
- Problem Statement
- Datasets
- Model Description
- Hyperparameter tuning
- Word Embeddings
- Results and Evaluation

Introduction:

What are rumours?



A story or statement whose truth value is **unverified** or deliberately **false**.

Motivation:

- Difficult to distinguish rumours by humans
- crucial to track and debunk rumors early to minimize their harmful effects
- Online fact-checking services have limited topical coverage and long delay
- During the outbreak of COVID-19 pandemic, social media platform used for information of new revelations. Some of the information is factual, others are rumours. This causes adverse effect on individuals and society. Identify rumours and stop them before they become viral.

Problem statement and Proposed Solution

Problem Statement:

- Given a dataset of tweets, classification of each tweet as rumour or non-rumour.

Solution:

- DNN model based on Bi-Directional Long Short-Term Memory with Convolutional Neural Network
- Considers contextual information in both directions for classification

Reference:

- Asghar, M.Z., Habib, A., Habib, A., Khan, A., Ali, R. and Khattak, A., 2019. Exploring deep neural networks for rumor detection. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-19.

Datasets

- **Pheme**: #Source tweets - 5800, Labels - 2046(R) / 3754 (NR)
- **Twitter15**: #Source tweets - 1490, Labels - 374 (NR), 370 (F), 372(T), 374(UR) collapsed to 374 (NR), 1116 (R)
- **Twitter16**: #Source tweets - 818, Labels - 205 (NR), 205 (F), 205(T), 203(UR) collapsed to 205 (NR), 613 (R)
- **COVID-19**: Obtained using Twitter API. #source tweets - 313036(Unlabeled - 29 Mar 2020 - 15 Apr-2020)

R - rumour, NR - non-rumour, F - false rumour,

T - true rumour, UR - unverified rumour

Data Preprocessing:

Input Tweet

↓

Preprocessing →
- Lower case
- Removal of punctuations
- Removal of stop words
- Removal of rare words

↓

Tokenization

↓

Seq of integers

↓

Vocabulary →
- Glove
- FastText
- Google word2vec

↓

To Embedding Layer

Model Description

Embedding - Googlevec, Fasttext, Glove

Dropout Layer - 0.5

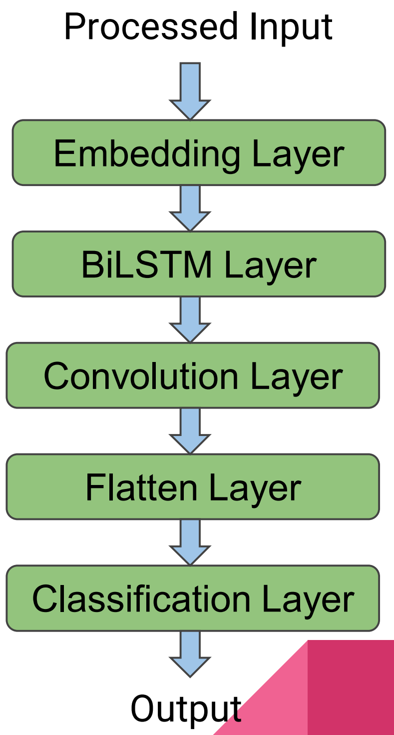
BiLSTM - Forward and backward (units =278)

Convolution - Feature extraction (f=8, k=2, relu)

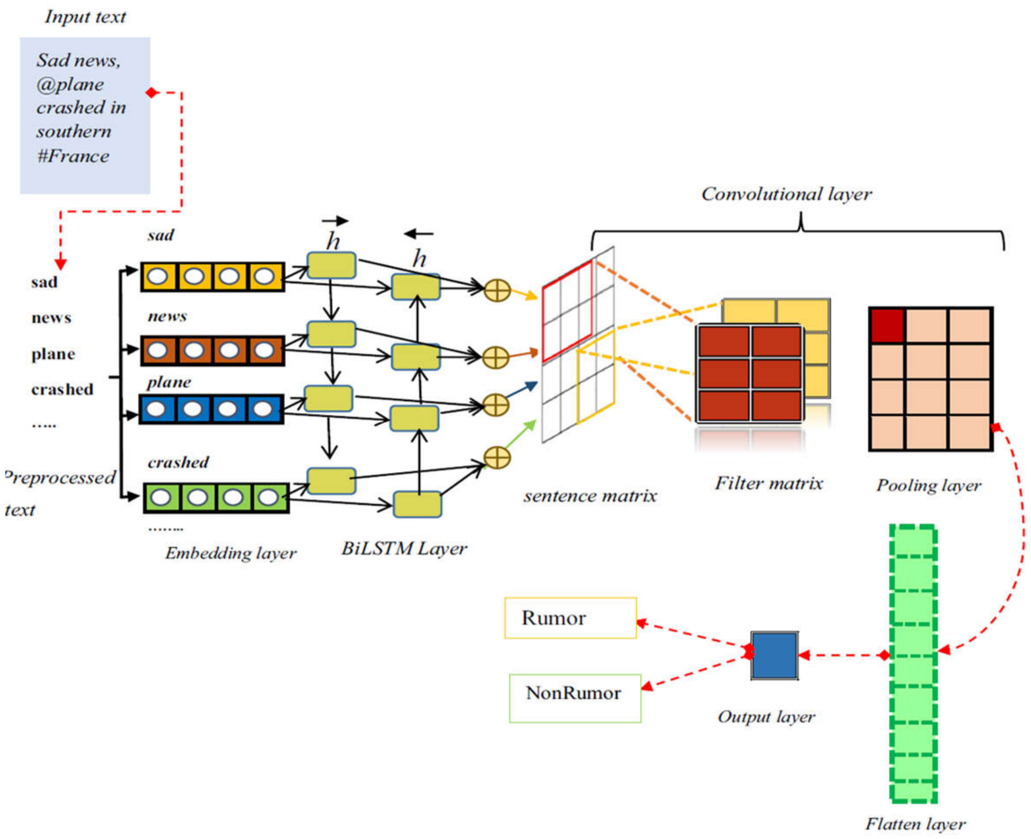
  Maxpool (2)

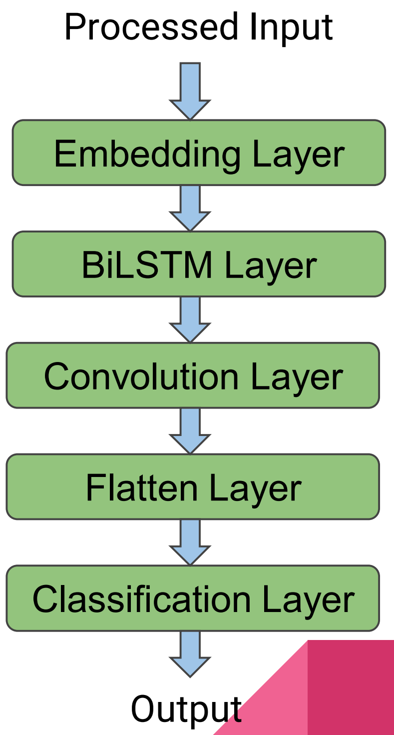
  Flatten

Single Neuron - Classification (Sigmoid)



Model Description





Hyperparameter Tuning

BiLSTM-CNN model Parameter setting

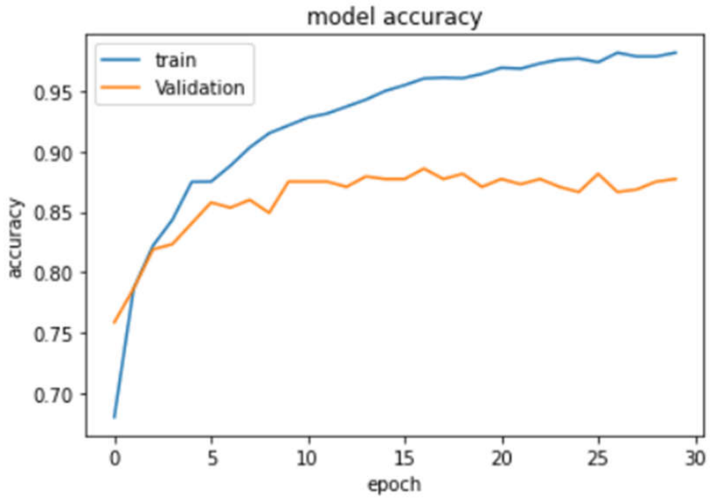
Train and Test: Pheme Dataset (Train:80%, Test:20%)

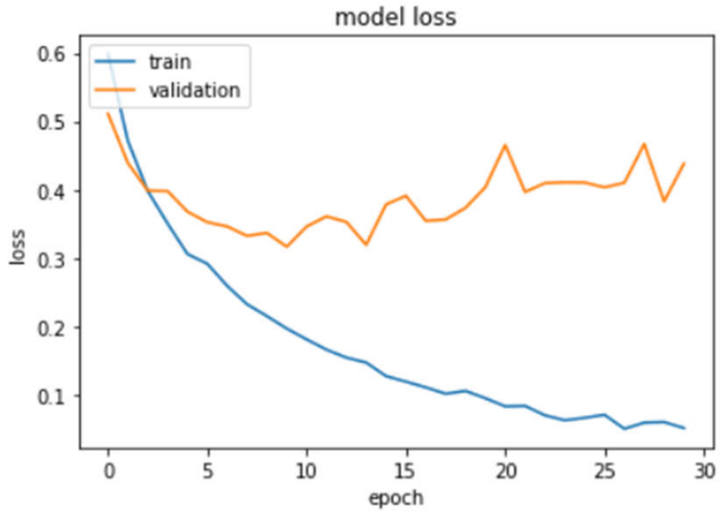
No.of Filters	Filter Size	Train acc	Train loss	Train time(s)	Test acc	Test loss	Precisi on	Recall	F1- Score
64	3	98.6	0.38	330	87	0.63	0.80	0.82	0.81
32	2	98.4	0.04	330	86.5	0.58	0.82	0.79	0.80
10	3	98.3	0.046	330	86	0.59	0.78	0.85	0.81
8	2	98	0.0538	432	87	0.60	0.80	0.83	0.82

Results:

Pheme Dataset: train - 80%(4640), Test- 20%(1160) #epochs = 30

Train: 90%(4176), Val: 10% (464)





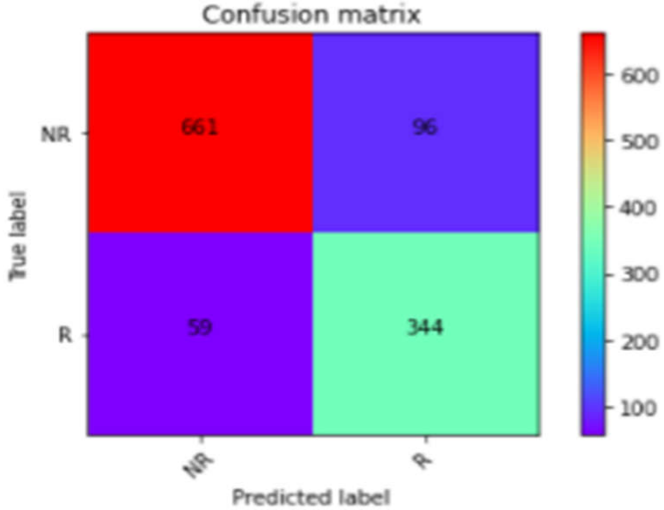
Train Accuracy: 0.9816
Validation Accuracy: 0.8772

Train Loss: 0.0510
Validation Loss: 0.438

Results:

Pheme Dataset:

```
Confusion matrix, without normalization
[[661  96]
 [ 59 344]]
```



	precision	recall	f1-score	support
NR	0.92	0.87	0.90	757
R	0.78	0.85	0.82	403
accuracy			0.87	1160
macro avg	0.85	0.86	0.86	1160
weighted avg	0.87	0.87	0.87	1160

Pre-trained Word Embeddings

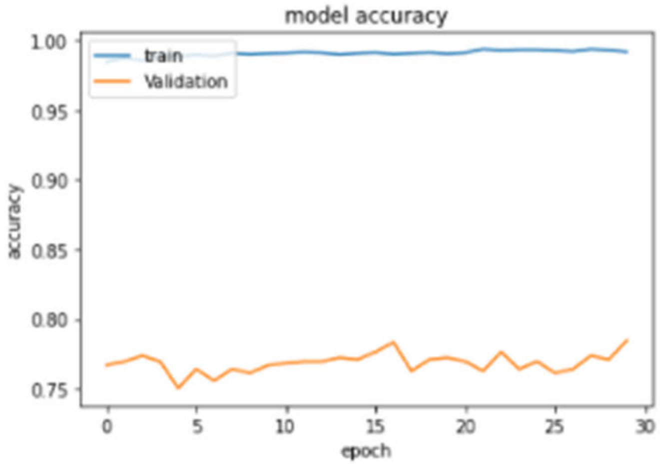
Performance evaluation on BiLSTM CNN model using Pretrained word embeddings

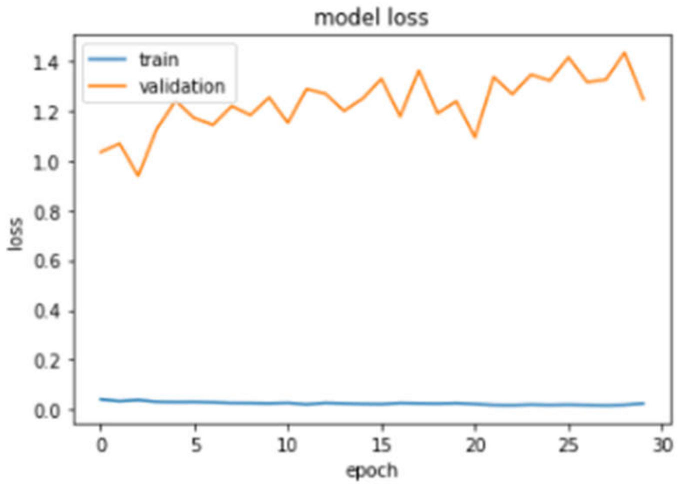
Train: Pheme + Twitter15 Test: Twitter16

Pretrained Word Embedding	Train Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1-score
Base Model	98.1	78.5	70	0.84	0.74	0.79
Fasttext	94.1	75.2	74.82	0.83	0.84	0.83
Glove	89.2	74.6	73.11	0.81	0.84	0.82
Google Word2Vec	92.7	77.5	73.23	0.80	0.86	0.83
With self attention	90.8	75.9	71.39	0.84	0.77	0.80

Results: Cross-Domain Training

Pheme + Twitter15 Dataset: train - 90%(6561), validation - 10%(729) #epochs = 30





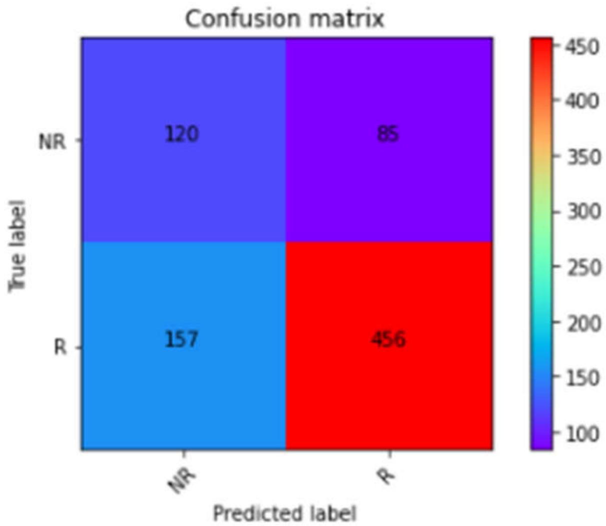
Train Accuracy: 0.981
Validation Accuracy: 0.785

Train Loss: 0.025
Validation Loss: 1.25

Results: Cross-Domain Training

Test Dataset: Twitter16 (818)

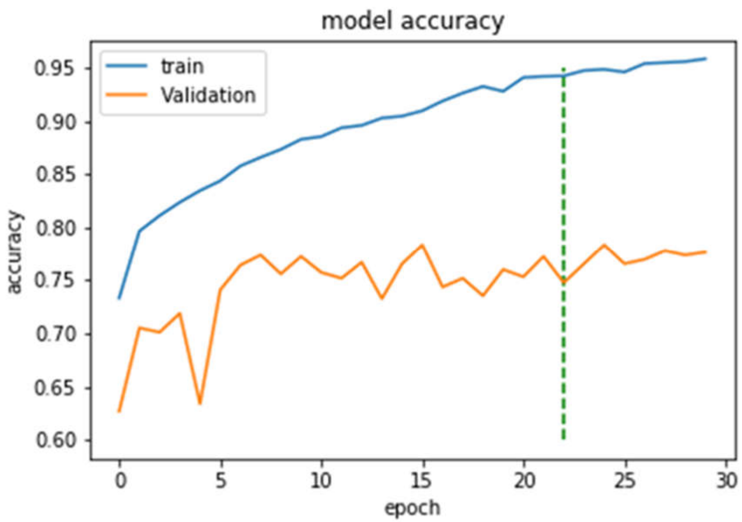
```
Confusion matrix, without normalization
[[120  85]
 [157 456]]
```

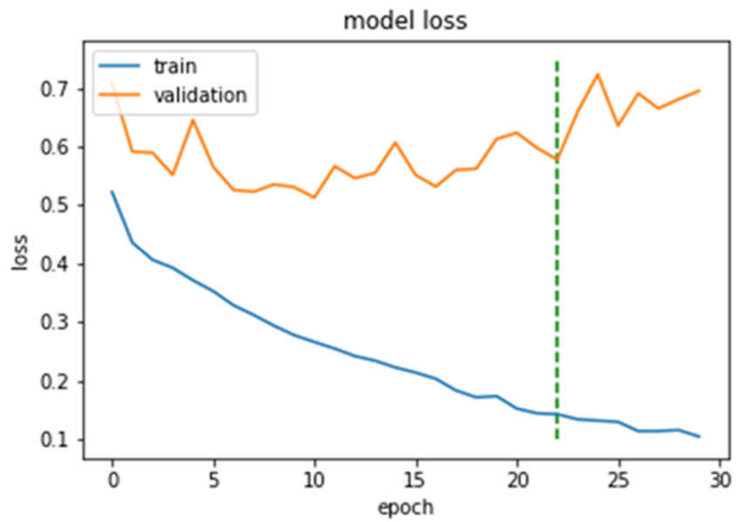


	precision	recall	f1-score	support
NR	0.43	0.59	0.50	205
R	0.84	0.74	0.79	613
accuracy			0.70	818
macro avg	0.64	0.66	0.64	818
weighted avg	0.74	0.70	0.72	818

FastText:

Train: Pheme + Twitter15   Test: Twitter16





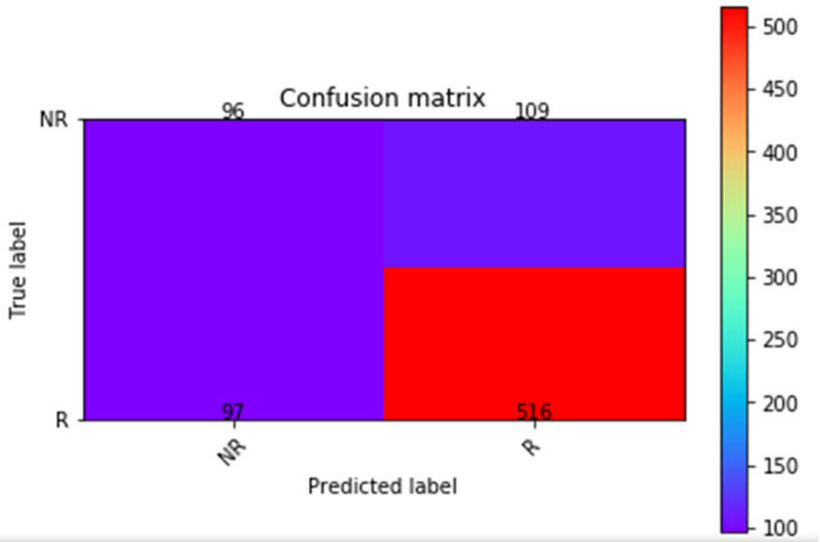
Train Accuracy: 0.941
Validation Accuracy: 0.752

Train Loss: 0.164
Validation Loss: 0.586

FastText:

Train: Pheme + Twitter15   Test: Twitter16

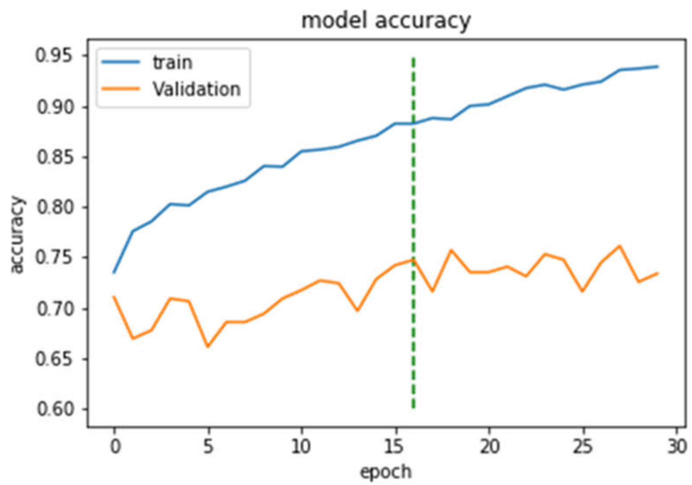
```
[[ 96 109]
 [ 97 516]]
```

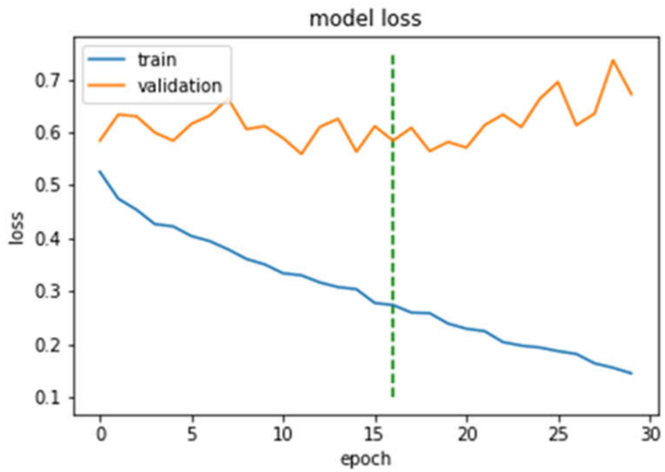


	precision	recall	f1-score	support
NR	0.50	0.47	0.48	205
R	0.83	0.84	0.83	613
accuracy			0.75	818
macro avg	0.66	0.66	0.66	818
weighted avg	0.74	0.75	0.75	818

Glove:

Train: Pheme + Twitter15   Test: Twitter16



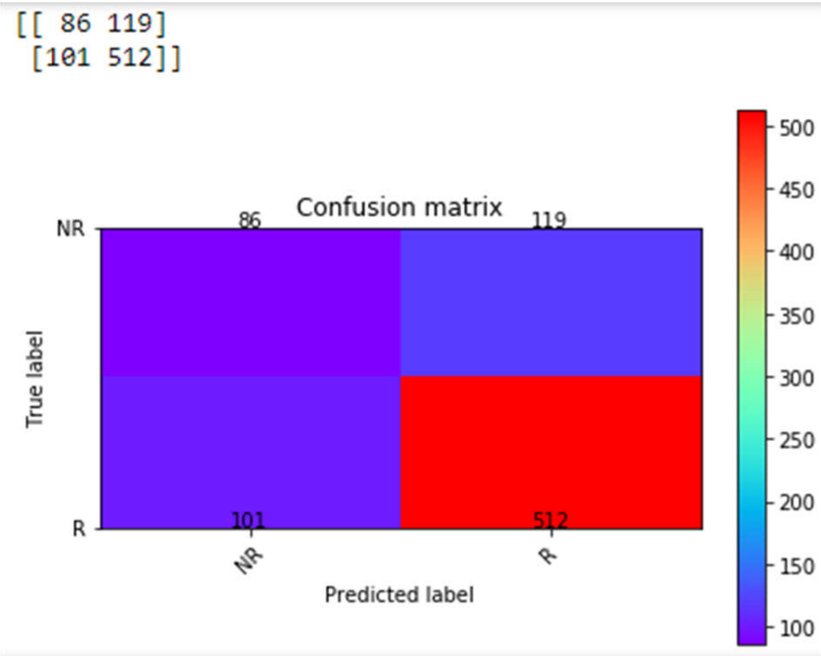


Train Accuracy: 0.892
Validation Accuracy: 0.746

Train Loss: 0.281
Validation Loss: 0.586

Glove:

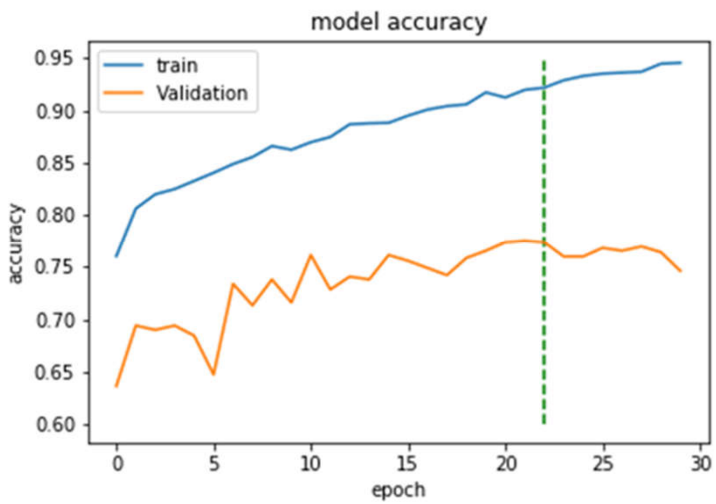
Train: Pheme + Twitter15   Test: Twitter16

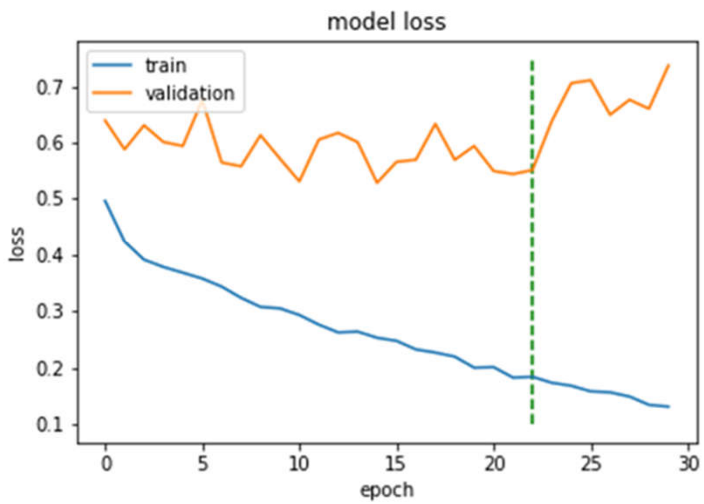


	precision	recall	f1-score	support
NR	0.46	0.42	0.44	205
R	0.81	0.84	0.82	613
accuracy			0.73	818
macro avg	0.64	0.63	0.63	818
weighted avg	0.72	0.73	0.73	818

Googlevec:

Train: Pheme + Twitter15   Test: Twitter16





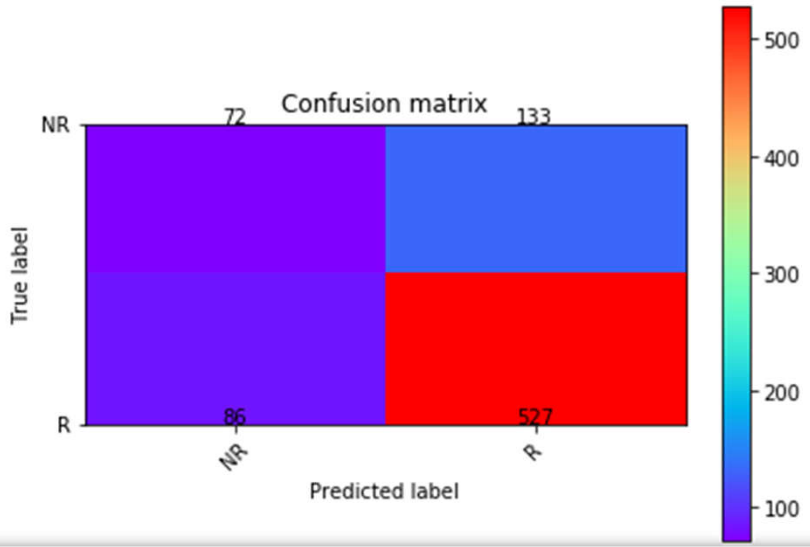
Train Accuracy: 0.927
Validation Accuracy: 0.775

Train Loss: 0.21
Validation Loss: 0.58

Googlevec:

Train: Pheme + Twitter15   Test: Twitter16

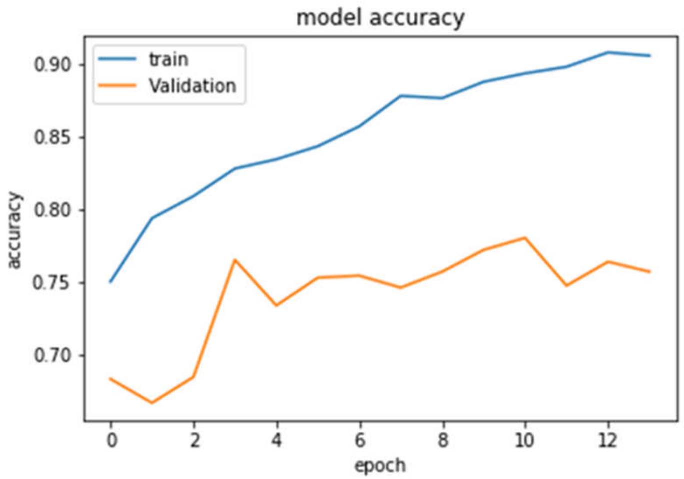
```
[[ 72 133]
 [ 86 527]]
```

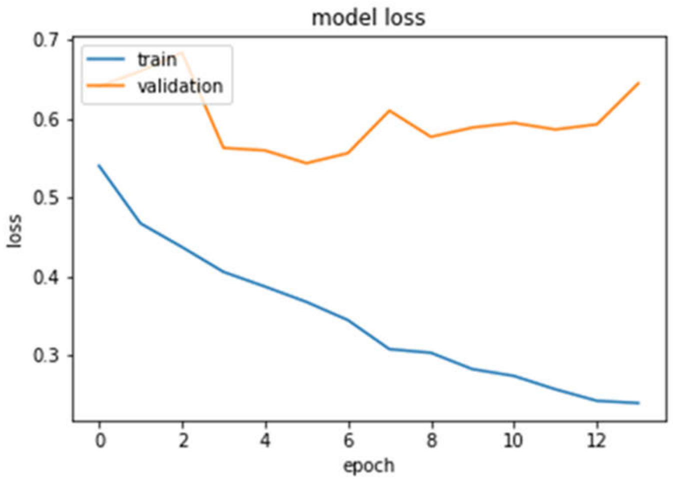


	precision	recall	f1-score	support
NR	0.46	0.35	0.40	205
R	0.80	0.86	0.83	613
accuracy			0.73	818
macro avg	0.63	0.61	0.61	818
weighted avg	0.71	0.73	0.72	818

With Self Attention Layer:

Train: Pheme + Twitter15   Test: Twitter16





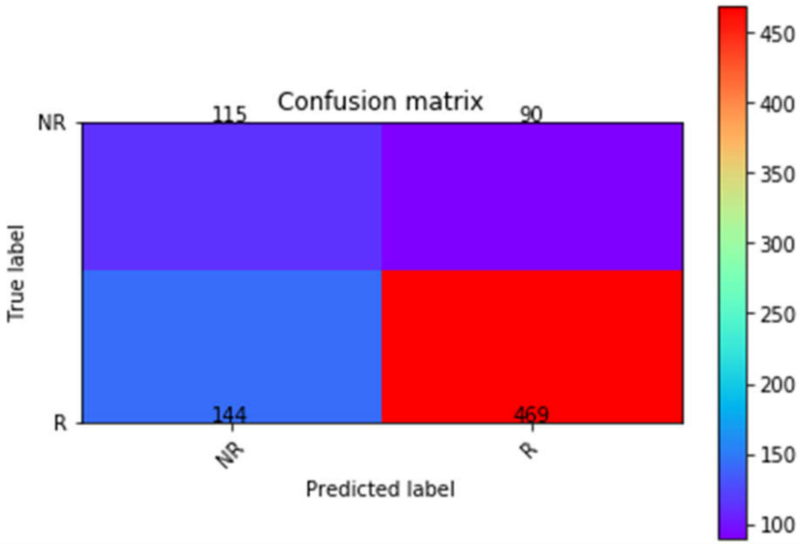
Train Accuracy: 0.908
Validation Accuracy: 0.759

Train Loss: 0.267
Validation Loss: 0.652

With Self Attention Layer:

Train: Pheme + Twitter15   Test: Twitter16

```
[[115  90]
 [144 469]]
```



	precision	recall	f1-score	support
NR	0.44	0.56	0.50	205
R	0.84	0.77	0.80	613
accuracy			0.71	818
macro avg	0.64	0.66	0.65	818
weighted avg	0.74	0.71	0.72	818

COVID-19 Dataset:

Twitter API

- Keywords - coronavirus, COVID-19, #COVID19
- Language - english
- Start date - 29 Mar 2020
- End date - 15 Apr 2020
- Metadata - text (Tweet)
- Tweepy package and Bearer_Token for data access

Preprocessing : Lower case, Removal of punctuations, stop words, rare words

Tokenizing

Prediction using model

COVID-19 Dataset Results:

Classification Results on COVID-19 Dataset (#tweets: 313036) by Bi-LSTM-CNN model with different word embeddings.

Bi-LSTM-CNN model with word embedding	Fasttext	Glove	Googlevec
#Rumours	71971	25570	50574
#Non-Rumours	241065	287466	262462

Conclusion:

- The problem of tweet classification into rumor and non-rumor has been addressed by using deep learning based BiLSTM-CNN model.
- Various experiments have been conducted like hyper parameter tuning, usage of pre-trained word embeddings, self attention layer and reported the results.
- Achieved the best results using FastText pre-trained word embedding model.
- Obtained the COVID-19 dataset from Twitter using Twitter API and used the same for rumor detection.

References:

[1] Ma, J., Gao, W. and Wong, K.F., 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

[2] Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y. and Huang, J., 2020, April. Rumor detection on social media with bi-directional graph convolutional networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 549-556)

[3] Asghar, M.Z., Habib, A., Habib, A., Khan, A., Ali, R. and Khattak, A., 2019. Exploring deep neural networks for rumor detection. Journal of Ambient Intelligence and Humanized Computing, pp.1-19

[4] Chen, E., Lerman, K. and Ferrara, E., 2020. Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set. JMIR Public Health and Surveillance, 6(2), p.e19273

[5] Christopher D. Manning, Hinrich Schuetze - Foundations of Statistical Natural Language Processing-The MIT Press (1999)

Thank You!