Research Assignment 4:

Artificial Intelligence and Deep Learning

Tyrone Brown

Abstract

In this research we explore the use of 3 types of recurrent networks: Simple RNN, LSTM and GRU. Each of these networks gives the capability of taking a sequence of data and attempting to remember what it has observed as it moves along the sequence. Simple RNNs are known to come up short with longer sequences, and that was proven true by this research. Both LSTM and GRU greatly outperformed all RNN configurations. The overall best model was an LSTM network with a 2 dimensional word embedding input. This model trained much quicker than others that performed near the top in terms of F1 score. GRU models came in at the top in terms of accuracy, however the training of these models was more computationally expensive than the top LSTM model. The F1 score of the best LSTM model was .8827 but the recommended model yielded an F1 score of .8808 in 25% of the training time (80 seconds vs 20 seconds).

Introduction

In previous studies we evaluated the performance of a sentiment analysis engine for IMDb film reviews built using a Dense Neural Network as well as a Convolutional Neural Network. This research expands on the topic but introduces the use of Recurrent Neural Networks. An advantage of RNNs over CNNs or DNNs is that an RNN can take a sequence (such as words in a sentence) and "remember" past information from the sequence as it steps through it. The types of RNNs that this research analyzes are Simple RNN networks, Long Short-Term Memory networks and Gated Recurrent Unit networks.

Literature Review

An interesting variant of this research was performed by Pawan Kumar Sarika in the paper "Comparing LSTM and GRU for Multiclass Sentiment Analysis of Movie Reviews" (Sarika, 2018). Instead of simply predicting positive or negative labels for IMDb reviews, Sarika's research covers predicting sentiment on a scale of 1 to 10. The accuracies of LSTM and GRU were compared and ultimately concluded that GRU was superior to LSTM, contrary to what was found in this research. Similarly, LSTM and GRU did have similar accuracies.

Methodology

Methodology Overview

For this research, SimpleRNN, LSTM and GRU models were analyzed and compared with varying architectures for each. The purpose of this group of networks is that they each allow for each word in a sequence to be stepped through, remembering information along the way prior to generating an output (positive or negative review).

Implementation and Programming

Figure 1 shows the libraries that were used in performing all experimentation steps. To load in the IMDb data, 300 was chosen as the maximum document length and the top 26 words were skipped using the Keras load_data() function. Since the training and test data are split in half as provided by Keras, both sets were combined to create one full 38,501 review dataset (full set is 50,000, however Keras removes testing data that exceed maximum length). From here, these were split into a training set of 24,640 reviews, a validation set of 6,160 reviews and a testing set of 6,161 reviews

The final LSTM model's implementation requires the use of TensorFlow/Keras to build and compile the model. The layers used within the model are Embedding(), LSTM() and Dense().

Data Preparation, Exploration and Visualization

The dataset used for training and testing the models used in this research is the IMDb movie review dataset that comes with and was obtained from TensorFlow's library. The data loads into a training set and testing set, consisting of 25,000 film reviews each. Each film consists of a label indicating whether the review's sentiment is negative (0) or positive (1). The subset that was used in this experimentation includes 19,346 negative reviews and 19,155 positive reviews.

Results

Experiment Results

267 models were evaluated as a part of this research. See Figure 2 for the top 25 results. For each network type, the parameters were varied in similar ways. Once a top baseline was achieved for each type, word embedding, and memory units were removed to create as sparse of networks that achieved satisfactory results.

SimpleRNN. This experiment consisted of SimpleRNN models with varying RNN layers, RNN units and word embedding dimensions. Additionally, both uni-directional and bi-directional memory layers were analyzed. The top SimpleRNN model used 300 embedding dimensions and 1 uni-directional RNN layer with 16 RNN memory units. Training for this model was completed in 330 seconds. The accuracy of the model was 85.89% with and F1 score of .8622. See Figure 3 for the progression of F1 score as embedding dimensions, RNN layers and RNN units were modified with all else held constant. The top 5 RNN configurations are shown in Figure 4.

LSTM. This experiment consisted of LSTM models with varying LSTM layers, LSTM units and word embedding dimensions. Additionally, both uni-directional and bi-directional memory layers were analyzed. The top LSTM model used 500 embedding dimensions and 1 uni-directional LSTM layer with 8 LSTM memory units. The accuracy of the model was 88.22% with and F1 score of .8827. See Figure 5 for the progression of F1 score as embedding dimensions, LSTM layers and LSTM units were modified with all else held constant.

While this model was the top overall performer, its training time was 80 seconds. However, reducing the Word Embedding dimensions from 500 to 2 proved to be much more

efficient. The F1 score decreases from .8827 to .8808 (with an increase in recall), however the time to train was reduced by 75% down to 20 seconds. The top 5 LSTM configurations are shown in Figure 6.

GRU. This experiment consisted of GRU models with varying GRU layers, GRU units and word embedding dimensions. Additionally, both uni-directional and bi-directional memory layers were analyzed. The top GRU model used 500 embedding dimensions and 1 bi-directional GRU layer with 8 GRU memory units. Training for this model was completed in 90 seconds. The accuracy of the model was 88.43% with and F1 score of .8815. See Figure 7 for the progression of F1 score as embedding dimensions, GRU layers and GRU units were modified with all else held constant. The top 5 GRU configurations are shown in Figure 8.

Key Observations

Simple RNN / LSTM / GRU. Out of the top 25 models (by F1 score), no SimpleRNN models were included. Not only were SimpleRNN networks the worst performers in classifying results, but the computational resources required far exceeded that of LSTM and GRU. A uni-directional SimpleRNN took 435 seconds on average to train while LSTM and GRU took around 65 seconds each for similar architectures. See Figure 9 for a summary of time and performance by model type.

Model Behaviors. While changing the embedding dimensions, layers and memory units, the behavior of each model was similar relative to one-another. LSTM and GRU show similar trends, while RNN performs a bit worse and seems to degrade while LSTM and GRU performance increases (ex. Additional embedding dimensions). See Figures 10-12 for the trends of each model type as word embedding dimensions were increased.

Conclusions

After evaluating all models and the impacts of various structures/parameters, the recommendation for this task is to use an LSTM network. Although not the top performer overall, and architecture only utilizing 2 word embedding dimensions provided the most efficient performance in terms of time/computational resources and performance. See Figure 13 for the training history and confusion matrix of this model. When controlling for all variables other than embedding dimensions/layers/memory units, less seems to be more in that the F1 scores for each were higher with fewer of each. However, when considering the configurations in relation to one another, fewer word embedding dimensions seems to have had a bigger impact on performance and speed.

As expected, due to its known shortcomings in long-term memory due to vanishing gradient, the RNN was not a viable option at all for performing this task. Results were inferior to both GRU and LSTM and training the models were extremely expensive.

References

Sarika, P. K. (2018). *Comparing LSTM and GRU for Multiclass Sentiment Analysis of Movie Reviews.* (Unpublished master's thesis). Blekinge Institute of Technology.

Appendix

Figure 1

Python Libraries Used for Research

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'  # or any {'0', '1', '2'}

# Helper libraries
import datetime
import time
from numpy.random import seed
from packaging import version
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score, auc, f1_score, precision_score, recall_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.ensemble import RandomForestClassifier

from collections import Counter
import numpy as np
import pandas as pd

# TensorFlow and tf.keras
import tensorflow as tf
tf.debugging.set_log_device_placement(False)

from tensorflow.keras.utils import to_categorical
from tensorflow import keras
from tensorflow.keras import preprocessing
from tensorflow.keras import models, layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, SimpleRNN,RNN, LSTM, Bidirectional, GRU
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import MaxPooling1D, MaxPooling2D, BatchNormalization
from tensorflow.keras.layers import Dropout, Flatten, Input, Dense
from tensorflow.keras.datasets import imdb
```

Figure 2

Top 15 Model Results

Type	Embed_Dims	Layers	Units	Bidirectional	Epochs	Time	Test Accuracy	Test Precision	Test Recall	Test F1
LSTM	500	1	8	0	5	80.314639	0.882223	0.877409	0.888137	0.882741
LSTM	500	1	64	0	5	83.880702	0.884301	0.900678	0.863424	0.881658
LSTM	300	2	8	0	5	66.540332	0.876769	0.846707	0.919615	0.881656
GRU	500	1	8	1	5	89.673430	0.884301	0.901550	0.862383	0.881532
GRU	300	1	64	0	5	52.646481	0.884431	0.906663	0.856660	0.880952
LSTM	2	1	8	0	5	20.338643	0.876899	0.852311	0.911290	0.880815
LSTM	500	1	32	1	5	98.511660	0.883781	0.909015	0.852497	0.879850
GRU	500	2	8	1	5	115.990555	0.884560	0.917255	0.844953	0.879621
LSTM	500	3	16	0	5	105.211872	0.877419	0.864139	0.895161	0.879376
LSTM	500	3	32	1	5	146.715487	0.881963	0.900847	0.857960	0.878881
GRU	300	3	64	0	5	78.039325	0.883262	0.914438	0.845213	0.878464
GRU	500	1	64	0	5	79.279962	0.883262	0.915374	0.844173	0.878333
GRU	500	3	64	1	5	167.875353	0.883132	0.915819	0.843392	0.878115
GRU	300	3	64	1	5	137.686554	0.884041	0.924360	0.836108	0.878022
LSTM	300	1	16	1	5	68.624712	0.880405	0.902063	0.853018	0.876855
GRU	500	1	64	1	5	100.985168	0.880275	0.903814	0.850676	0.876441
LSTM	8	1	8	0	5	20.031910	0.880405	0.908354	0.845734	0.875926
LSTM	500	1	5	0	5	90.830172	0.877938	0.898463	0.851717	0.874466
LSTM	500	2	32	1	5	123.602031	0.879236	0.910423	0.840791	0.874222
LSTM	500	1	64	1	5	109.896741	0.880405	0.920818	0.831946	0.874129
LSTM	100	2	64	0	5	43.774473	0.878847	0.908733	0.841831	0.874004
LSTM	300	2	32	1	5	95.757177	0.876769	0.892809	0.855879	0.873954
LSTM	300	2	32	0	5	68.608816	0.878847	0.909423	0.841051	0.873902
GRU	500	2	32	1	5	117.360364	0.878717	0.911948	0.837929	0.873373
LSTM	300	1	64	1	5	75.086601	0.878457	0.914009	0.835068	0.872757

Figure 3

SimpleRNN Model Performance (F1) against Parameter Changes

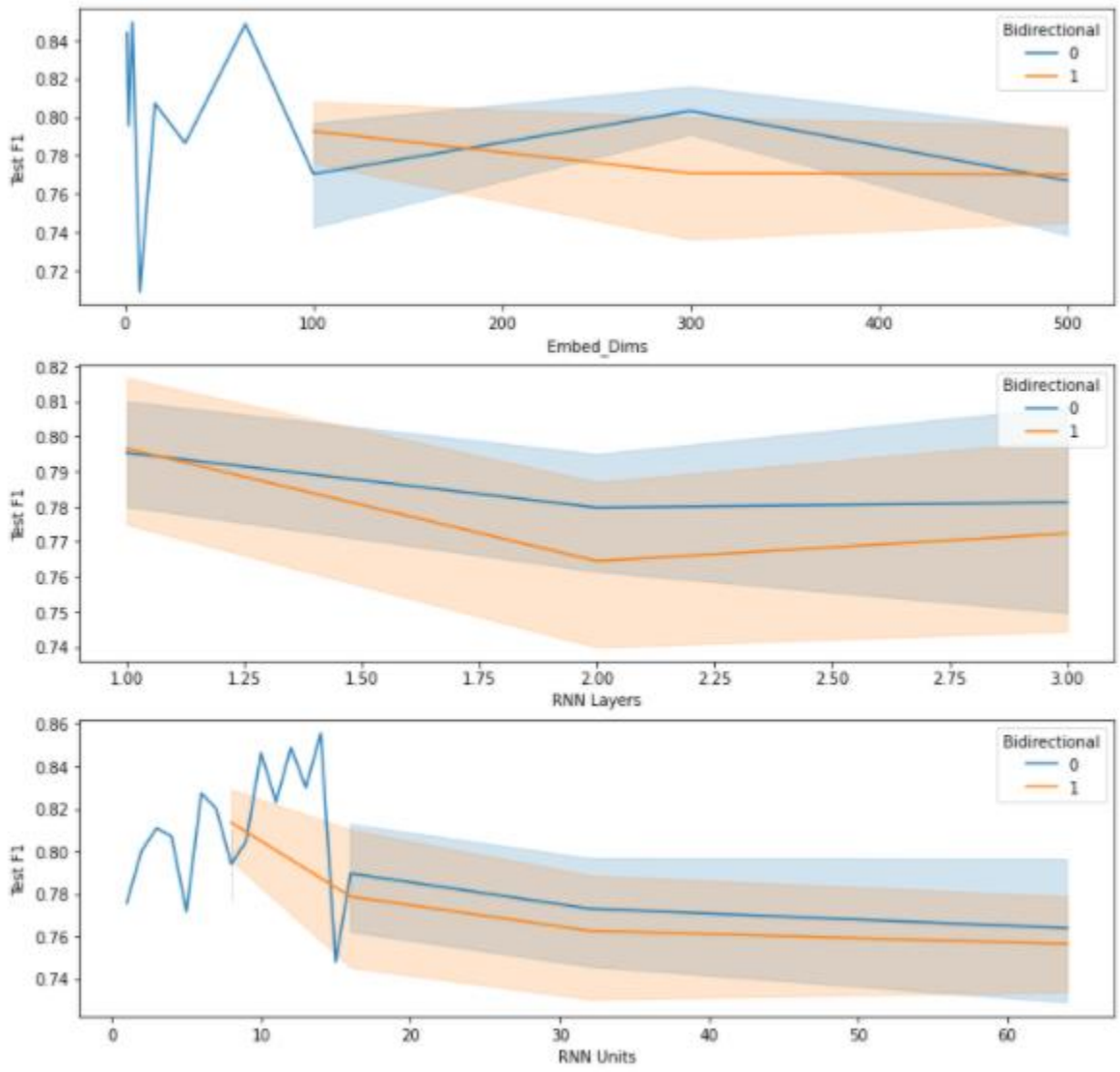

Figure 4

Top 5 SimpleRNN Model Configurations

Embed_Dims	RNN Layers	RNN Units	Epochs	Bidirectional	Time	Test Accuracy	Test Precision	Test Recall	Test F1
300	1	16	5	0	329.543164	0.858850	0.840960	0.884495	0.862178
300	1	14	5	0	238.880294	0.860538	0.886009	0.827003	0.855490
500	1	16	5	1	602.046327	0.856123	0.869131	0.837929	0.853245
4	1	16	5	0	195.575758	0.855603	0.886312	0.815297	0.849323
300	1	12	5	0	244.321100	0.854694	0.883048	0.817118	0.848804

Figure 5

LSTM Model Performance (F1) against Parameter Changes

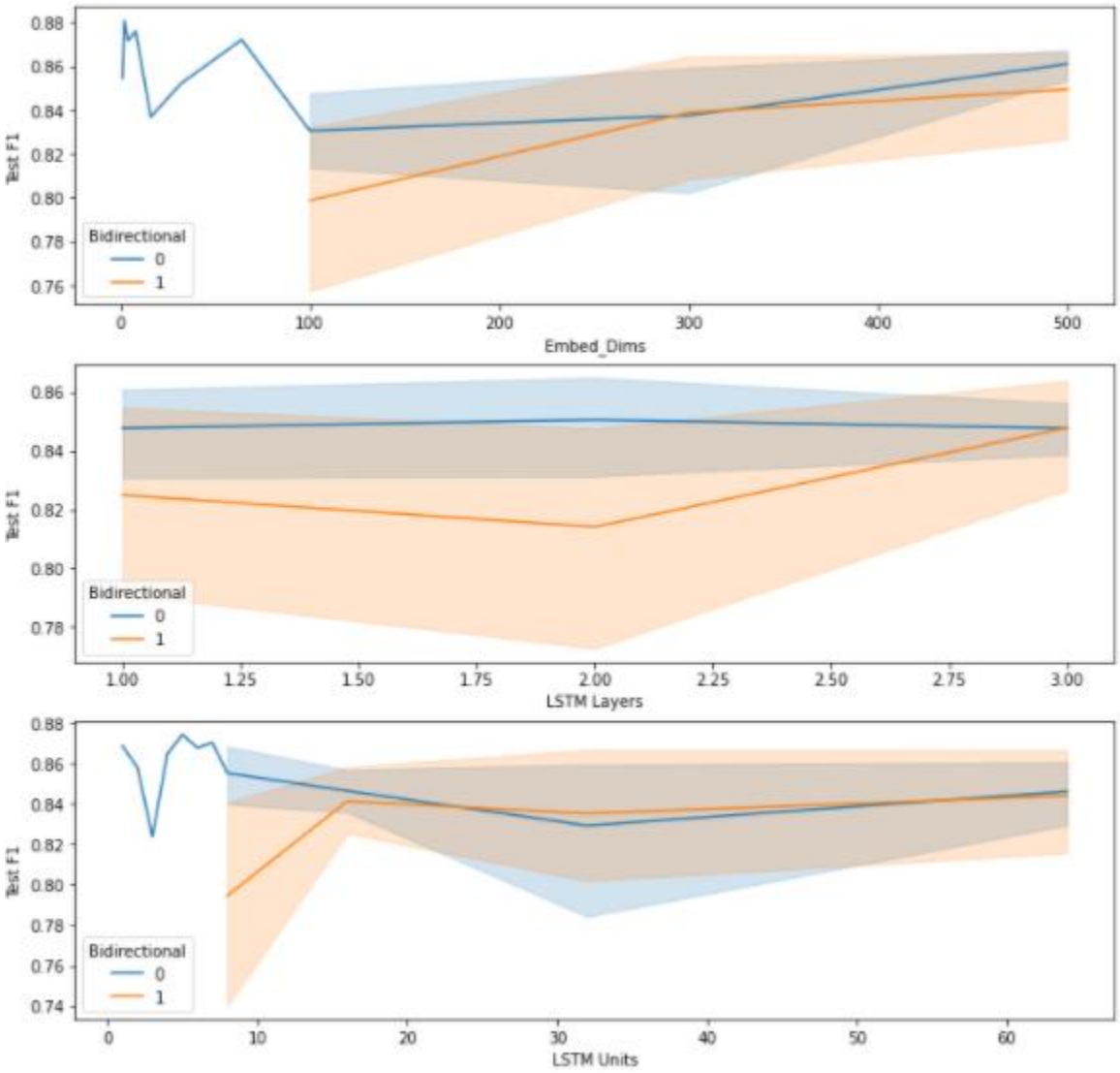

Figure 6

Top 5 LSTM Model Configurations

Embed_Dims	LSTM Layers	LSTM Units	Bidirectional	Epochs	Time	Test Accuracy	Test Precision	Test Recall	Test F1
500	1	8	0	5	80.314639	0.882223	0.877409	0.888137	0.882741
500	1	64	0	5	83.880702	0.884301	0.900678	0.863424	0.881658
300	2	8	0	5	66.540332	0.876769	0.846707	0.919615	0.881656
2	1	8	0	5	20.338643	0.876899	0.852311	0.911290	0.880815
500	1	32	1	5	98.511660	0.883781	0.909015	0.852497	0.879850

Figure 7

GRU Model Performance (F1) against Parameter Changes

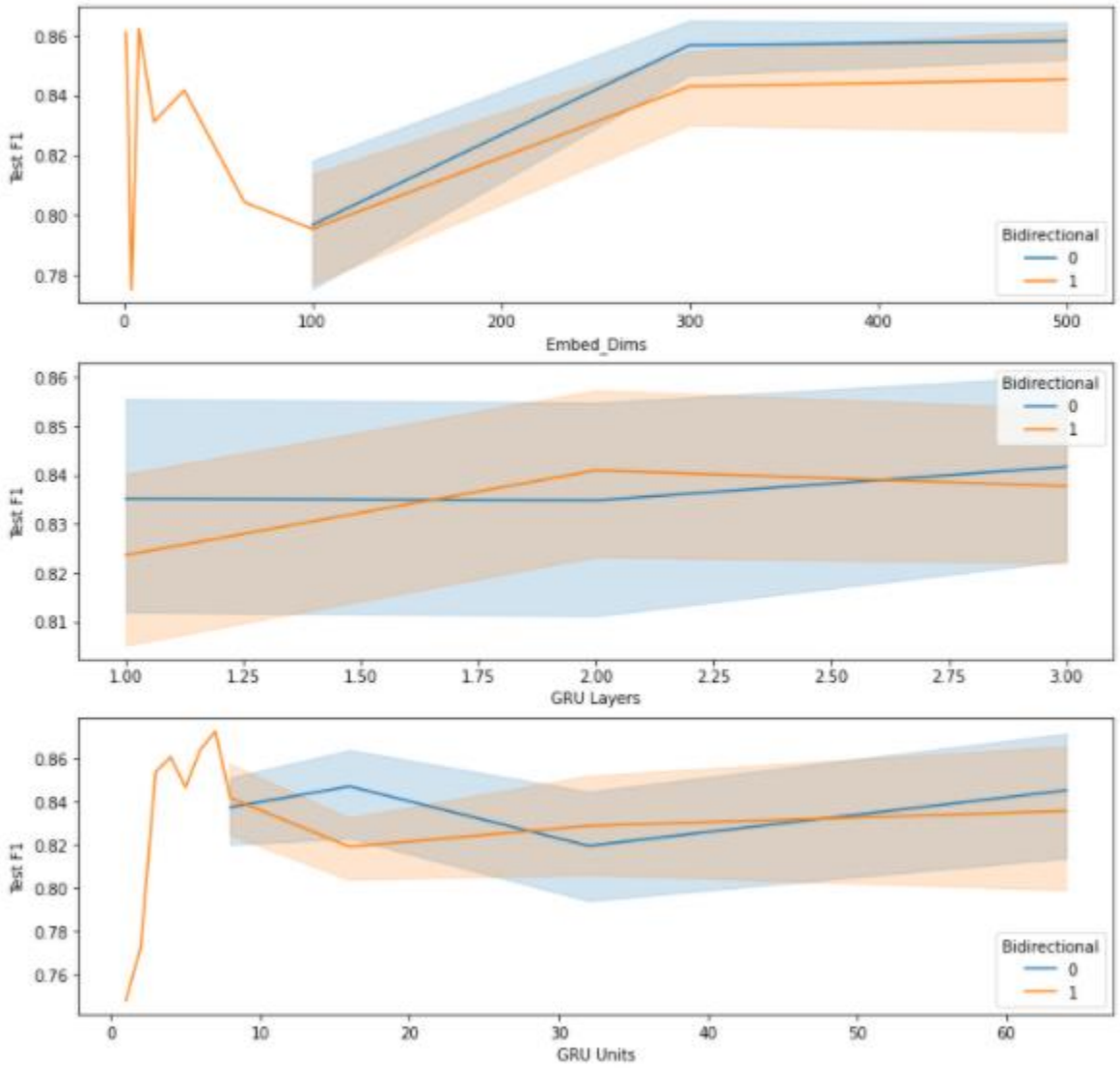

Figure 8

Top 5 GRU Model Configurations

Embed_Dims	GRU Layers	GRU Units	Bidirectional	Epochs	Time	Test Accuracy	Test Precision	Test Recall	Test F1
500	1	8	1	5	89.673430	0.884301	0.901550	0.862383	0.881532
300	1	64	0	5	52.646481	0.884431	0.906663	0.856660	0.880952
500	2	8	1	5	115.990555	0.884560	0.917255	0.844953	0.879621
300	3	64	0	5	78.039325	0.883262	0.914438	0.845213	0.878464
500	1	64	0	5	79.279962	0.883262	0.915374	0.844173	0.878333

Figure 9

Average Performance by Network Type

Type	Bidirectional	Bidirectional	Time	Test Accuracy	Test Precision	Test Recall	Test F1
GRU	0	0	64.901178	0.854251	0.936540	0.761136	0.837226
	1	1	85.024177	0.845885	0.916091	0.765716	0.831018
LSTM	0	0	65.287076	0.859977	0.916448	0.794719	0.848328
	1	1	98.802764	0.846849	0.923294	0.762639	0.828954
RNN	0	0	435.107013	0.804683	0.856024	0.740433	0.789163
	1	1	990.123492	0.794821	0.847972	0.726435	0.777755

Figure 10

F1 against Word Embedding Dimension

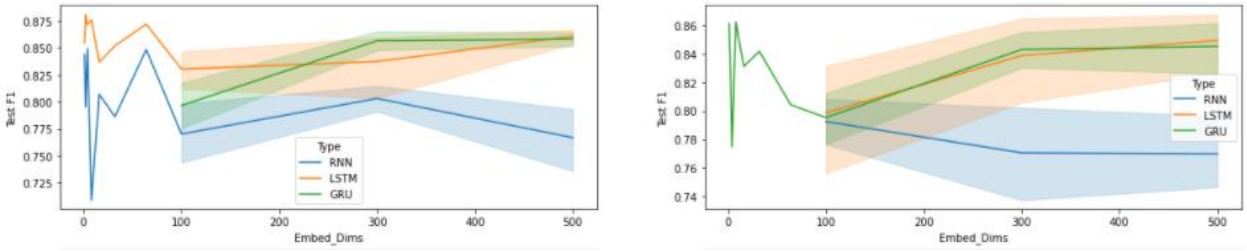

Figure 11

F1 against Layer Count

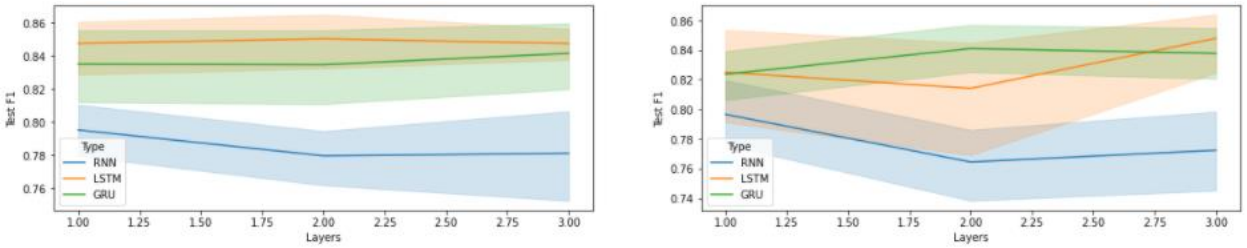

Figure 12

F1 against Memory Units

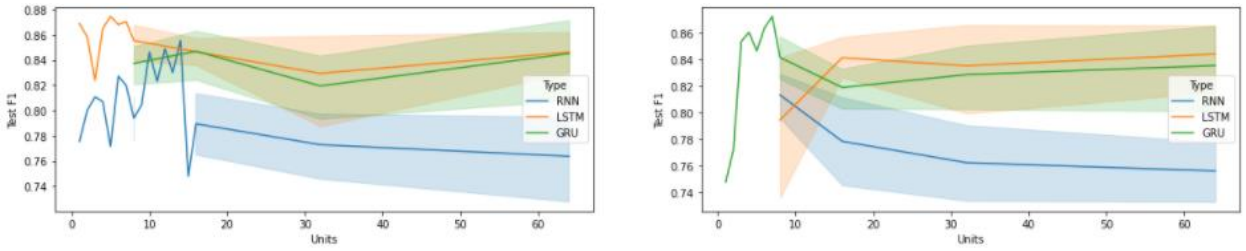

Figure 13

Recommended LSTM Model Accuracy Trend and Confusion Matrix