Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

CSE 4238

Soft Computing Lab

Assignment # 03

Submitted To,

Sanzana Karim Lora
Lecturer
CSE, AUST

Submitted By,

MD. Abu Yousuf Sajal
ID 170104025

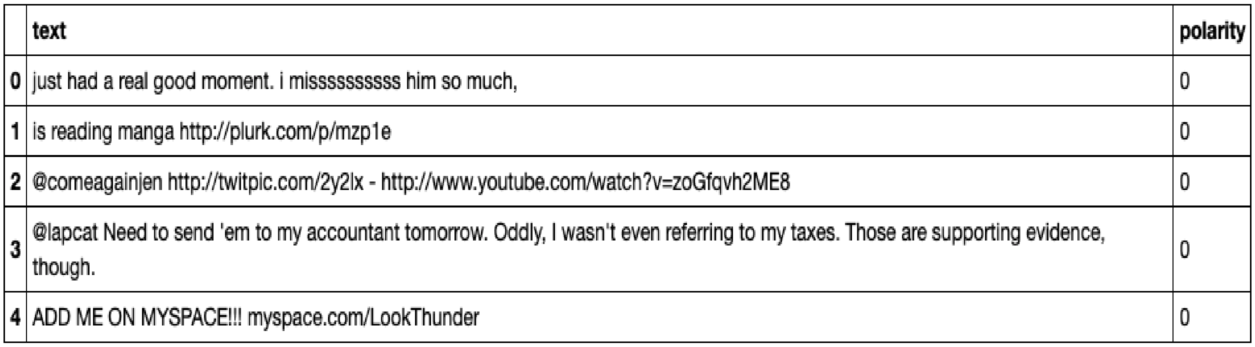
Date of Submission: September 25, 2021

DataSet Analysis

DataSet-2

Structure of Dataset

Text containing every rows along with the polarity each of items in another column.



	text	polarity
0	just had a real good moment. i missssssssss him so much,	0
1	is reading manga http://plurk.com/p/mzp1e	0
2	@comeagainjen http://twitpic.com/2y2lx - http://www.youtube.com/watch?v=zoGfqvh2ME8	0
3	@lapcat Need to send 'em to my accountant tomorrow. Oddly, I wasn't even referring to my taxes. Those are supporting evidence, though.	0
4	ADD ME ON MYSPACE!!! myspace.com/LookThunder	0

pre-processing Dataset

The texts in the dataset including unnecessary words, punctuations, redundant words. So to do a better operation we need to pre-process the text. Hence few tasks had been done to create a more logical text for the experiment.

- Lower Casing
- Removal of Punctuations
- Removal of stop words
- Removal of Frequent words
- Removal of Rare words
- Lemmatization

Some Analytics of Dataset

Top 5 Most Common Words

go	878
get	505
u	499
time	457
make	430

Total Words = 85529
Total Unique Words = 1024
Maximum Words in a Sentence = 50
Minimum Words in a Sentence = 1

Experiment Analysis

Model Used : Bidirectional LSTM
The idea of Bidirectional Recurrent Neural Networks (RNNs) is straightforward.
It involves duplicating the first recurrent layer in the network so that there are now two layers side-by-side, then providing the input sequence as-is as input to the first layer and providing a reversed copy of the input sequence to the second.
In bidirectional LSTM, instead of training a single model, we introduce two. The first model learns the sequence of the input provided, and the second model learns the reverse of that sequence.

Model Architecture

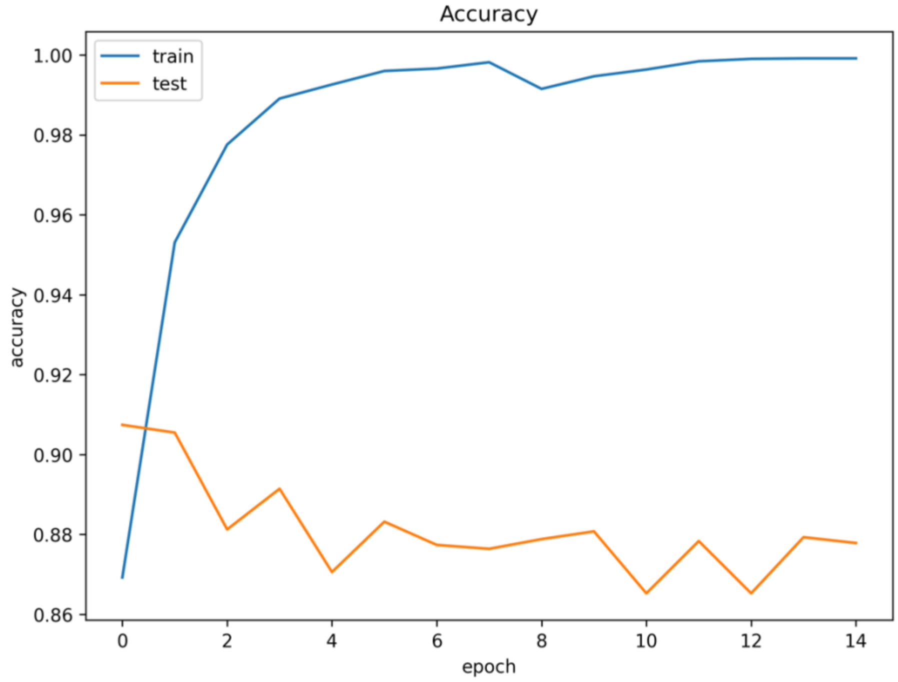
Hyper parameters :
loss = binary cross entropy
optimizer = adam
batch = 32
epoch = 15

Model Summery

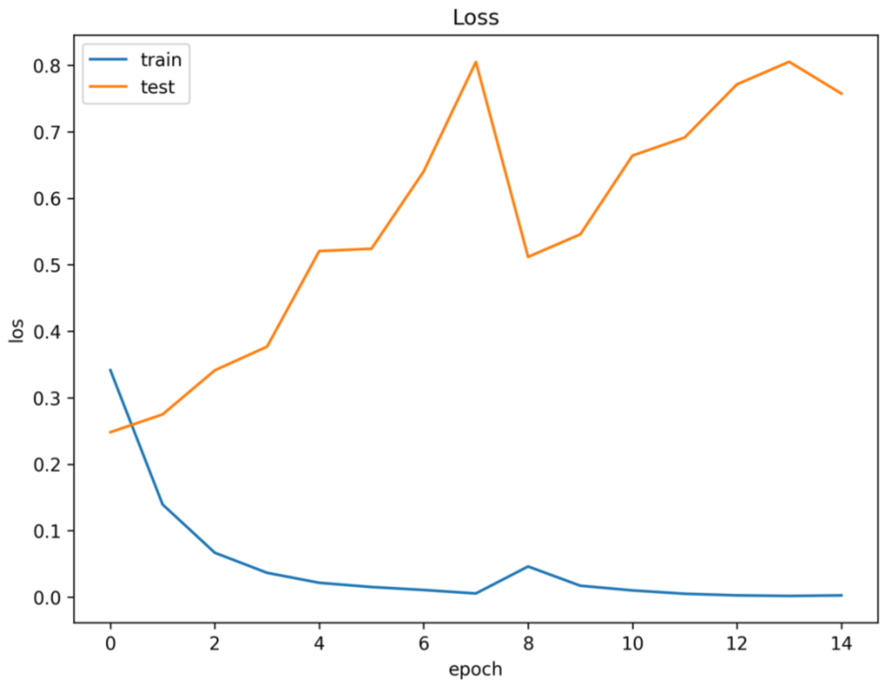
```
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, None)]            0         
_________________________________________________________________
embedding (Embedding)        (None, None, 128)         1311360   
_________________________________________________________________
bidirectional (Bidirectional (None, None, 128)         98816     
_________________________________________________________________
bidirectional_1 (Bidirection (None, None, 128)         98816     
_________________________________________________________________
bidirectional_2 (Bidirection (None, None, 128)         98816     
_________________________________________________________________
bidirectional_3 (Bidirection (None, 128)               98816     
_________________________________________________________________
dense (Dense)                (None, 1)                 129       
=================================================================
Total params: 1,706,753
Trainable params: 1,706,753
Non-trainable params: 0
```

Result Analysis

Accuracy Graph



Loss Graph



Performance Measurement Metrics

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

Accuracy = TP+TN/TP+FP+FN+TN
Accuracy = 0.87

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate.

Precision = TP/TP+FP
Precision = 0.87

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

Recall = TP/TP+FN
Recall = 0.87

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)
F1 Score = 0.87

Confusion Matrix - Well, it is a performance measurement for machine learning classification problem where output can be two or more classes.

1455	126
126	156

GitHub Code