**Assignment 3 NLP Sentiment Analysis Report**

###### Group 2: Vignesh Murali, Ritvik Reddy, Ziqing Lu

***Abstract***

*This report covers NLP sentiment analysis using different models and tools: BOW, GLOVE, LSTM, Amazon, Google, Microsoft and Watson APIs. Confusion matrix of different experiments are compared in the this report. All experiments are based on Financial Dataset covering 2018 Q4 earnings transcript for 12 companies. IMDB dataset is also used in this experiments for transfer learning purpose.*

Experiment 1

* **Experiment 1 [RNN]:**

Best model settings:

optimizer= ‘AdaDelta’

loss=’Poisson’

epochs=100

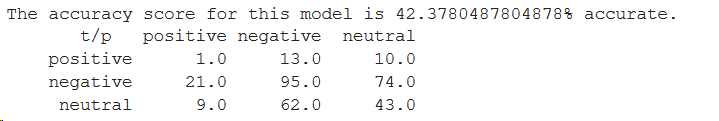
batch\_size=128

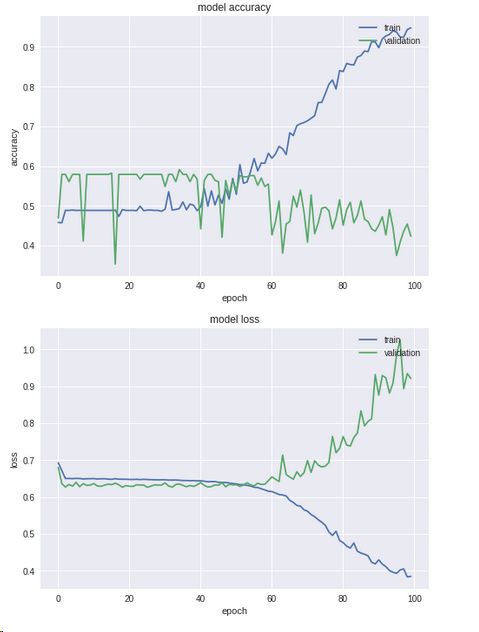
structure:Embedding Layer, LSTM layer, Dense layer

LSTM layer size = 128

Accuracy: train=0.94,test=0.42

Confusion Matrix:



****

Observation: Loss function was set to ‘Poisson’ because other loss functions were giving a negative loss value.

* **Experiment 1 [Bag of Words]:**

Best model settings:

optimizer= ‘Adam’

loss=’categorical\_cross\_entropy’

epochs=2

batch\_size=32

structure:Dense layer, dropout,activation

Accuracy: train=0.8032,test=0.5909

Matrix:

**[ 0 11 15]**

**[ 0 127 41]**

**[ 0 29 105]**

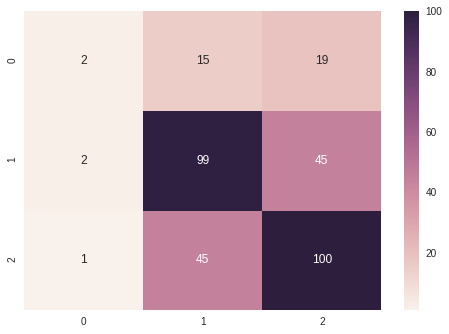
* **Experiment 1 With “GloVe”**

Train the model based on GLOVE embedding layer

**1. Best model:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim | optimizer | learning rate | Accuracy (train) | Accuracy (val) | Accuracy (test) | Notebook |
| 300 | Adam | 0.001 | 0.998 | 0.636 | 0.6128 | GlOVE-CNN.ipynb |

**2. Concusion Matrix:**

****

**3. Observation:**

- Actual Positive: 67.8% accurately classified

- Actual Neutral: 68.4% accurately classified

- Actual Negative: 5.5% accurately classified

- Most negative are falsely predicted

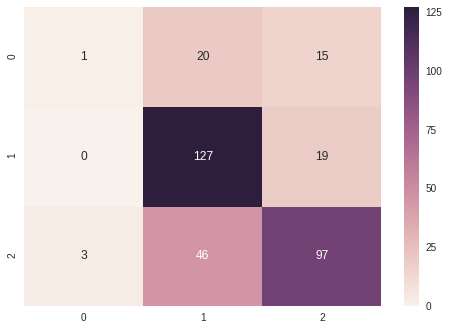
* **Experiment 1 With “GloVe + TextCNN”**

Train the model based on GLOVE embedding layer + CNN

**1. Best Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim | optimizer | learning rate | Accuracy (train) | Accuracy (val) | Accuracy (test) | Notebook |
| 300dim | Adam | 0.1 | 0.961 | 0.71 | 0.6859 | GlOVE-CNN.ipynb |

**2. Confusion Matrix:**

****

**3. Observation:**

- For Negative: 2.7% accuracy

- For Positive: 66.4% accuracy

- For Neutral: 86.9% accuracy

- No Negative Labels are correctly predicted

Experiment 2 Transfer Learning

* **[RNN]**

Best model settings:

Transfer learning has been performed on the IMDB dataset

optimizer= ‘AdaDelta’

loss=’Poisson’

epochs=100

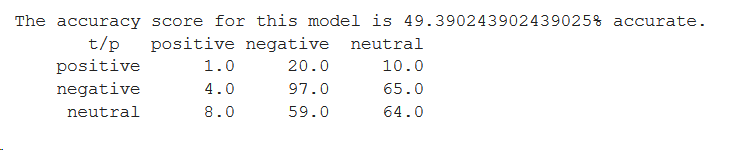
batch\_size=128

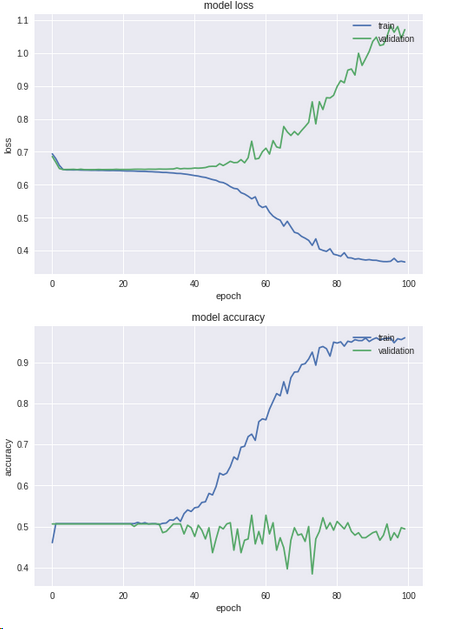
structure:Embedding Layer, LSTM layer, Dense layer

LSTM layer size = 128

accuracy: train = 0.96, test=0.49

confusion matrix:





* **[Bag Of Words]**

Best model settings:

Transfer learning has been performed on the IMDB dataset

optimizer= ‘Adam’

loss=”categorical\_crossentropy’

epochs=50

batch\_size=32

structure:dense relu dropout activation(softmax)

accuracy: train = 0.51, test=0.46

confusion matrix:

**0[ 0 33 0]**

**1[ 0 164 0]**

**2[ 0 131 0]**

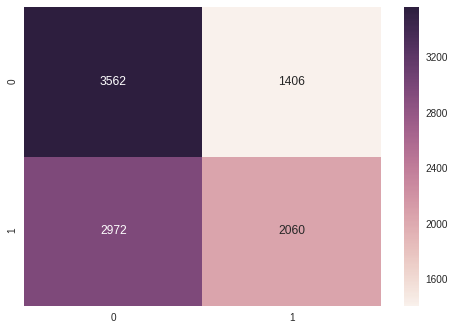
* **Experiment 2 With “GloVe”**

Train IMDB with Glove Embedding Layer

1. **Best Model of IMDB:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | optimizer | learning rate | accuracy (train) | accuracy (test) | Notebook |
| with Glove - IMDB | rmsprop | 0.001 | 0.91 | 0.5622 | Experiment-2-CNN-GLOVE.ipynb |

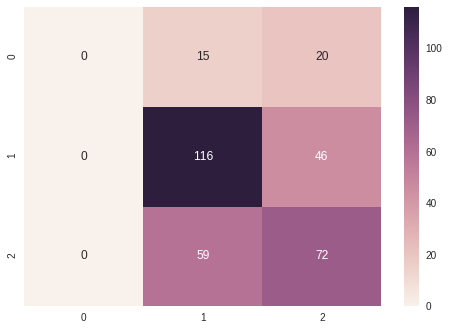
1. **Confusion Matrix of IMDB Model:**

****

1. **Best Model of Transfer Learning Result:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| optimizer | learning rate | Accuracy (train) | Accuracy (val) | Accuracy (test) | Notebook |
| SGD | 0.1 | 0.596 | 0.625 | 0.573 | Experiment-2-CNN-GLOVE.ipynb |

**4. Consusion Matrix:**

****

**5. Observation:**

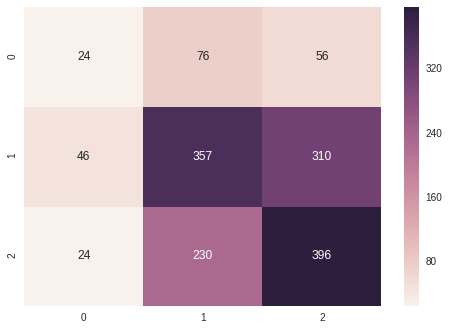
Transfer Learninig Accuracy is lower than training directly on the target data

Experiment 3 With API

**Average of 4 APIs:**

> Accuracy: 0.512

> Confusion Matrix:



> Threshold:

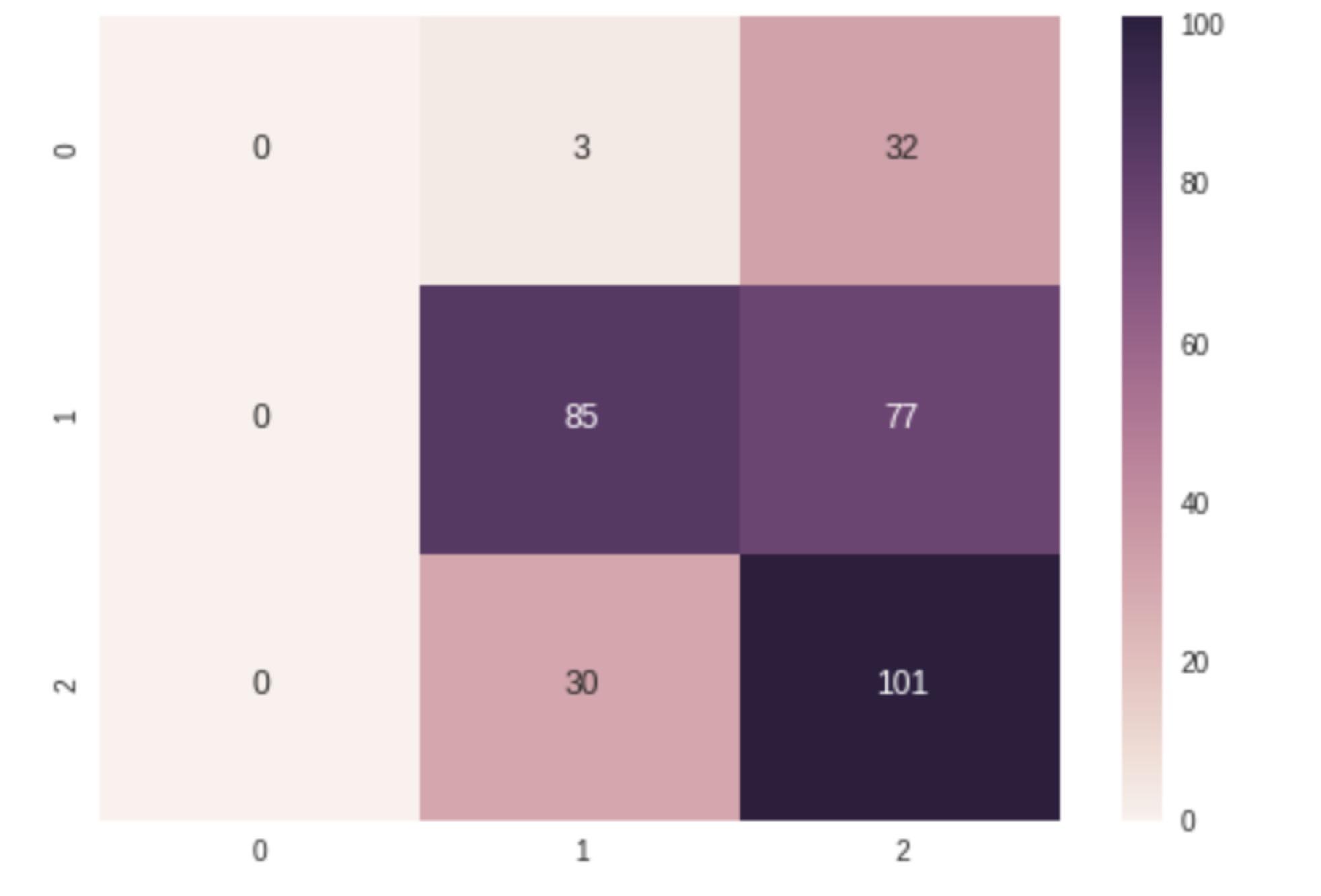
1. > 0.45: positive
2. < 0.30: negative
3. else: neutral

> Observation:Still difficult to correctly predict negative

**IBM**

> Accuracy: 0.474

> Confusion Matrix:

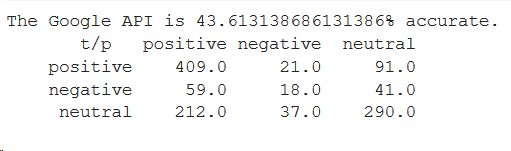


> Observation: IBM cannot give a label if the paragraph is too short

**Google:**

Accuracy:0.4361

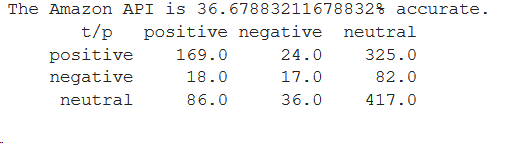
Confusion Matrix:



**Amazon:**

Accuracy:0.3667

Confusion Matrix:



Microsoft:

pos [305, 39, 310]

neg [ 56, 21, 80]

neutral [326, 42, 465]

Experiment 4 AutoML

**AutoSKLearn**

auto-sklearn is an automated machine learning toolkit and a drop-in replacement for a scikit-learn estimator.

Accuracy:0.58

Runtime:60 minutes

Number of models tested: 50

**H2O AutoML**

H2O’s AutoML can be used for automating the machine learning workflow, which includes automatic training and tuning of many models within a user-specified time-limit.

Acc:0.561332

Runtime:5 minutes

Number of models tested: 20

**TPOT**

TPOT is a Tree-Based Pipeline Optimization Tool (TPOT) , one of the very first AutoML methods. The goal of TPOT is to automate the building of ML pipelines by combining a flexible [expression tree](https://en.wikipedia.org/wiki/Binary_expression_tree) representation of pipelines with stochastic search algorithms such as [genetic programming](https://en.wikipedia.org/wiki/Genetic_programming).[[1]](#footnote-0)

Acc: 0.557

Runtime: 60 minutes

Number of generations: 162

Experiment 5 Test

The best model in terms of accuracy was found to be Bag\_Of\_Vectors.

So Decided to run Bag of words on our test i.e GE dataset

* **[Bag Of Words]**

Best model settings:

Transfer learning has been performed on the IMDB dataset

optimizer= ‘Adam’

loss=”categorical\_crossentropy’

epochs=10

batch\_size=32

structure:dense relu dropout activation(softmax)

accuracy: train = 0.98, test=0.71

confusion matrix:

**0[ 6 0 0]**

**1[ 1 1 0]**

**2[ 1 0 8]**

Summary

1. Across all of our experiments, we found it is difficult to correctly predict negative labels, major reason is that there are limited data points for training negative paragraphs. The data set is too imbalanced for a classification problem
2. We build our RNN models with LSTM layers, we noticed that using 1 LSTM yielded better results than using stacked LSTM layers for this dataset.
3. While training our best RNN model on the IMDB dataset we noticed that the model trained much faster and performed a lot better, this is maybe due to quality of data and adequate preprocessing done to the IMDB dataset beforehand by keras.
4. In terms of Training with GloVe Embedding Layer, it delivered better general accuracy to add textCNN layers on top of embedding Layer than to add normal dense layers
5. Transfer Learning did not perform better than training using target dataset, probably due to the fact that financial sentiment differ too much from movie review sentiment
6. Average Score from 4 Sentiment Analysis API did not produce a better result than training on our own

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Exp # | optimizer | learning rate | Accuracy (train) | Accuracy (val) | Accuracy (test) |
| with Glove embedding layer | Adam | 0.001 | 0.998 | 0.636 | 0.6128 |
| with Glove embedding layer + CNN | Adam | 0.1 | 0.961 | 0.71 | 0.6859 |
| with Glove - IMDB transfer to - Financial | SGD | 0.1 | 0.596 | 0.625 | 0.573 |
| Experiment 1 Using 1 LSTM layer | AdaDelta | Adaptive | 0.9483 | 0.4543 | 128 |
| Exp 2 Transfer Learning on EDGAR dataset [1 LSTM layer load wts] | AdaDelta | Adaptive | 0.9597 | 0.4939 | 128 |
| Microsoft Azure API |  |  |  |  | 0.4811 |
| AutoSKLearn |  |  |  |  | 0.58 |
| Average Score - 4 Sentiment Analysis API |  |  |  |  | 0.52 |

1. https://automl.info/tpot/ [↑](#footnote-ref-0)