CptS 570: Machine Learning
Dr. Jana Doppa

Multi-label Classification of TED Talks using Neural Networks.

Final Project Report - Fall 2021
Deep Inamdar
Pallavi Sharma
December 15, 2021

1. Abstract

TED talk is a classic global platform dedicated towards spreading knowledge. This paper focuses on performing Multi-label Text Classification (MLTC) of TED talks based on topics, using Neural Networks. The paper takes a deep dive into discussing the underlying concepts and methodologies behind multilabel classification and text summarization. We have empirical findings for multi-label classification and theoretical study on the Text summarization techniques.

2. Introduction

The motivation behind the paper was the nature of the TED talks. On the TEDx platform, experts share views and experiences on diverse topics succinctly in a time bound environment. For the very reason, TED talk scripts are smaller in size and very different from each other.

Every TED transcript can be bucketed into multiple topics or hashtags. The label is decided on the semantic meaning of the text. The very nature of human dialect makes the MLTC a complex problem. The paper introduces the data set and the problem statement. Thereafter it explains the general methodology to tackle the MLTC using neural networks. Finally, it showcases empirical results found by performing various experiments involving hyperparameter tuning and deciding on evaluation matrix. The dataset, problem statement, and challenges are further explained down the paper.

3. Related Work

There exists a plethora of research on multilabel image and text classification. It has been observed that Recursive Neural Networks are best suited for text classification and Convolutional Neural Networks for the image classification problem. Pretrained word embeddings, proven and tested NLP algorithms, and mastered hyperparameter tuning strategies are key research contributions in this area. Specifically, to TED dataset, there is work related to predicting number of views, sentimental analysis based on comments, and exploratory data analysis. The problem which we handle in this paper is unique and has not been picked up so far.

4. Data set

As of 1984 till 2020-year end, there exist only a 4000+ publicly available TED talks. Our dataset contains 4004 English TED talks. Concerned features include – Title, Description, Full transcript, and list of multiple topics (hashtags) per TED talk. The total unique topics are found to be 427. The average word count is around 3000 words ranging from 500 to 8000 words. Average length of topics associated with each TED talk is 7. Out of the total 19 features, we focused on the mentioned textual features which directly relate to the MLTC task.

5. Problem Statement and Challenges

Problem Statement

The TED dataset contains a feature named ‘Topics’ - a list of areas the specific TED talk is based on. The task is to predict these topics using Recursive Neural Networks (RNN). Solving this problem would help us further propose list of other related TED talks.

Challenges

- Sparce Data - The data set is small which requires critical preprocessing for its effective utilization.
- High Variance - The data set covers multiple topics which makes it harder to learn as the word semantics is different for each data point. This makes the RNN slip into a local minimum often.

- High number of labels - The unique number of labels is too high. This makes predicting the True Positives and False hard leading to poor precision and recall. Performing MLTC on TED is harder than a common sentimental analysis problem, where the text (in the form of comments) is related to a common topic.
- TED talk transcript length - The range of TED talk text length is wide. This makes the weight vector calculations harder as there exist opposing updates nullifying the learning. This leads to overfitting.
- TED talks often contain direct speech and storytelling. This makes the presumption that the frequently occurring words have most impact negative. Introducing an inductive bias on this premise is necessary.

6. Multi-label Text Classification (MLTC)

The jargons and the optimal strategies to solve MLTC are complex. The paper carries out study pertaining to textual representation semantics and propose alterations to gain improved results tailored to our use case. Following are the concepts related to the same.

Bag of words – A preprocessing technique to transform the text into a representational NLP model.

Stemming or Lemmatization – Techniques to treat derivative forms of a word as similar. Lemmatization is an old approach where the last 3 letters of a word are truncated. Risky since the root word might get altered. Stemming is a more robust approach to tackle this task where the root word is kept intact. We performed Stemming.

Stop words – Set of Frequently occurring words which have no direct relevance to the topics. Inbuilt libraries have pre-filtered stop words. E.g., The NLKT corpus has the stopwords library in Keras. We made use of the Stop words corpus from the nltk library.

Tokenization - The string representation of words is label encoded in numeric values. Tokenization can be word-specific or sentence-specific. One-hot encoding, Keras tokenization are few implementations. We performed both word-specific tokenization and sentence-specific tokenization. Due to the scarcity of the data with high variance, sentence embedding technique gave poor results.

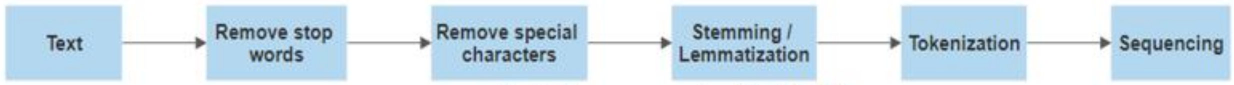
Sequencing – Neural Networks must have fixed length input. Sequencing pads the data-point word vectors to a uniform length. We made use of the average transcript word count to set the padding.

Multilabel Binarizer – Along with the transcript processing, we also need to split the label vector into binary feature matrix.

Word Embeddings - A word embedding is a learned representation for text where words have preserved context and semantics. Word2Vec, GloVe are types of pre-trained word embeddings. These represent a large corpus of words in a n-dimensional weighed vector space. Word scores such as TF-IDF weighed words also help preserve the relative occurrence of words in a text.

Embedding layer – The prerequisite of any Neural Network for NLP is to have a tokenized and weighted word vector. Embedding layer provides the same. Pre-trained embeddings such as GloVe can be fed as weights for better performance and reducing computational cost.

Sentence Embedding – For semantics of a sentence than a word, the embedding is done at sentence level. E.g., Doc2Vec and SentenceBERT.



Text Pre-processing for MLTC

7. Methodology and Experiments

7.1 Logistics

After pre-processing the data, we stored the tokenized word vectors and the multiclass binarized labels in an output file for re-usability. The sequence was padded to 10,000 words which serves as the input to the Neural Network. We made use of the pre-trained 100-dimention GloVe weighted word Embedding layer. This was connected to the Keras Simple Neural Network layer - 100 neurons output layer with tanh activation function. The output of the NN layer was connected to a Dense layer - 20 neurons output layer with sigmoid activation function. We made use of the frequently occurring top 20 labels/topics. The model was trained with a learning rate of 0.1. Evaluation matrices include accuracy, binary accuracy, F1-score, and Precision-Recall. The training data was split into validation set with a ratio 4:1. The train to test ratio was 5:1. We used the Adam optimizer to train the model. Every learning was triggered on 20 epochs.

7.2 Hyperparameter list

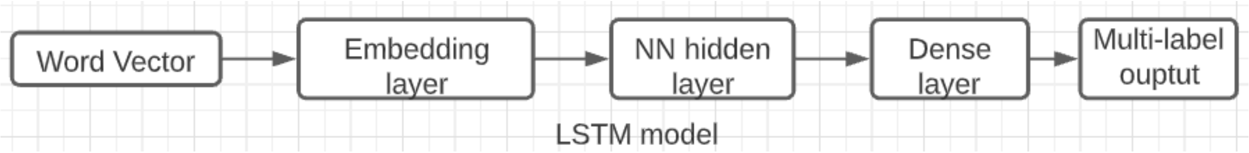
The hyperparameters we performed experiments on include – upper limit of topics, SimpleNN unit size, type of embedding, activation functions, learning rate, optimizers, batch size, word vector padding, vocabulary size, and number of epochs.

7.3 Pre-processing and EDA

- Data cleaning – remove noise from the data. There exist few data points which had no transcripts. Words such as (Applause), (Laughter), (Music) were removed.
- Removal of special characters, numbers and stop words, merging Title-Topics-Transcript columns, Lemmatization, tokenization, and padding. Multilabel binarized encoding was done for the labels.
- Merge the Title, Description, and Transcript column. This introduced redundancy in the data and the specific word corpus for each TED would lead to better results.
- The frequency of topics was plotted, and top 20 topics were selected for the MLTC task. [Appendix: Plot-1]
- Exploratory Data Analysis on frequency of words, average word count per TED talk was done for further reference. This was helpful to tune hyperparameters such as padding and setting the vocabulary size.

7.4 Neural Network model

After the preprocessing of the transcripts and the labels, we construct a RNN model. The RNN model is a 3-layer NN with an Input layer, an Embedding layer, and a RNN hidden layer, followed by an output Dense layer.



8. Experiments and Results

Following are the results of the hyperparameter tuning experiments which were preformed:

8.1 Most frequent topics Iterations – There exist 427 unique topics. We iterated through the top frequently occurring topics in incremental steps starting from 5 till 427. We observed decent accuracy of 72% till 20 topics. The precision was high, but the recall was a bit low. As we go beyond 20 topics, the precision-recall drops down below 0.3 with an accuracy of 60%. This is due to the decreasing number of True Positives and increasing False Positives. The True Positives are minority labels and predicting them is the toughest. [Appendix: Plot-1]

8.2 Vocabulary size Iterations – Vocabulary size is a decisive factor in constructing the Neural Network. The Tokenizer makes use of it to only consider the top frequently occurring vocabulary. From the EDA we knew that there exists 47000 unique words combining all the TED talks. We triggered our tests on a varying vocabulary size ranging from 5, 50, 500, 5000, 50000. We observed best results for a vocabulary of 20000 words. We hypothesize that the evaluation would yield better results for higher vocabularies if the epochs were increased with lower learning rates. [Appendix: Plot-2]

8.3 Padding length Iterations – From EDA we observed that the range of number of words per TED talk goes from 1000 to 7500. Setting a padding size to the mean value of 4000 gave good results, but more than 20% TED talks suffered data loss. To overcome this issue, we implemented two strategies –

1. Keep padding size to minimum – 1000. Split TED talks to multiple TED talks with disjoint transcripts of size 1000 with the same labels. Here, we dissolve the issue of data loss.
2. Keep padding size to maximum – 10000. For TEDs having word vector length less than 10000, append the same text until it matches the size. Here, we dissolve the issue of padded zeroes.

For both the above approaches, there was a rise in F1 score by 10%. Another observation was that the learning curve stopped being stagnant after few epochs. [Appendix: Plot-3]

8.4 Changing learning rate – We tested learning rates from 1, 0.1, 0.001, 0.0001. Best learning curve achieved for 0.1. Dataset has high variances - faster learning rate results in faster weight updates. For LR=1, we get skewed accuracy curve dipping after 7 epochs as it crosses global optima. For lower learning rates, it took multiple epochs to spot visible difference in accuracy and F1 scores. [Appendix: Plot-4]

8.5 Changing activation functions – We tried out Linear, ReLU, Sigmoid, and tanh. The best results were observed when tanh was used for the neural network hidden layer. For the output layer we used Sigmoid which gives the probability between 0 and 1. We used the binary cross entropy loss for weight updates. The tanh-sigmoid duo worked best for as - tanh gave bipolar results between -1 & 1 which set a threshold of 0.5 on sigmoid outputs. Tanh gave a rise of 5% in the accuracy and F1 scores. [Appendix: Plot-5]

8.6 Optimization functions - We tried out Gradient Descent optimizer types such as Adam's optimizer and SDG. Adam's optimizer served better for the dataset. Initially we encountered high accuracy of >0.75 but low stagnant F1 scores <0.25. This was a case of the model hitting a local minimum. The root cause was the lower learning rate and not the optimization function.

8.7 Changing Embedding layer – We worked with having no embedding layer. The accuracy was around 25%. Then we used the Keras embedding layer with no pre-trained weights. This improved the accuracy by 10%. It was only after using the GloVe pre-trained 100-dimensional word embedding that the accuracy spiked almost twice as it was before. Word2Vec and GloVe achieved similar results for the fact that both are weighted word vectorization techniques and preserve contextual and semantic information. [Appendix: Plot-6]

8.8 Epochs – For any combination of hyperparameters, after 10 epochs, the validation and training accuracies won't change a bit except for high learning rates which resulted in a noisy accuracy curve. [Appendix: Plot-7]

9. Evaluation Matrix

- In a multi-label classification problem, relying on accuracy is not a good idea since it takes into consideration only the TP and TN but does not penalize us for the FP and FN. Binary Accuracy keeps a threshold of 0.5 to evaluate regression results on a binary scale. In Keras, implicitly the accuracy measure is the Binary accuracy measure depending on the optimization function used.
- F1-score is a better measure for multi-label classification. Detecting TP and minimizing FP is a challenge here. High Precision is directly proportional to TP as better the TP better the FP. Our data set suffered on Recall more than Precision, since predicting at least 10% of labels correctly gave good TP but majority of the labels were then classified as FP which had an impact on Recall. Though our AUC of the Precision-Recall was low, we tried to maintain an equal Precision-Recall score.
- There is no inbuilt implementation of F1 score in Keras. Overall F1 measure is measured using F1-micro and F1-macro. In F1-micro, the Precision and Recall is calculated on individual data points and then averaged. Whereas in F1-macro, the Precision and Recall is calculated considering the TP, TN, FP, FN for all the data points together and then averaged. To be precise, we implemented the F1-macro measure since our use case demands evaluating overall performance assigning equal weightage to every data point based on most frequent class labels.

10. Future Scope

- Multi-label classification can also be performed using supervised learning techniques. After the BOW is ready, the supervised learning algorithms such as Naïve Bayes or SVM. Comparing the results, we got by the NN approach with the supervised learning results would be interesting to know.
- Use word corpuses from other languages and predict labels for non-EN TED talks using the built model.

In the limited time frame, we could achieve good results for the multi-label classification on TED talks. We intend to take this ahead to perform other NLP tasks such as –

- Propose related TED talks – Now that we found a way to bucket a TED talk in multiple labels, it can be used as a first filter for relevant TED talks. But using just the label will give a plethora of recommendations. We need to combine this with a Ranking algorithm based on Title, Speaker background, Publication date, number of likes, and Description of the TED talk. Our strategy was to perform a K-means clustering on non-textual features. Use the results of the MLTC for the target TED talk. Get the labels of the speaker domain. Find the union of all these labels. Rank the TED talks on the intersection levels with the target labels to spit out top K relevant TED talks. The evaluation matrix would be the F1-score based on the true values of recommendations.
- Predict the Topic – We saw TF-IDF scoring technique which suggests importance of words in a text. Using TF-IDF and NN we wish to achieve this. This is an Abstractive summarization problem - a harder one.
- Text Summarization – We aim to perform an Abstractive Text Summarization task with the ROUGH evaluation matrix.

11. Results

We started off with an accuracy and F1 score as low as 20%. EDA results were used to tune hyperparameters such as vocabulary, padding length, and topic frequency. Introducing Pre-trained Embedding layer GloVe improved the F1 score by 100%. Other experiments to tune hyperparameters related to the Neural Network model yield us better results. Final Outputs were as follows –

Results

Training Accuracy – 0.7132
Training F1 (Macro): 0.6726 Precision: 0.8475 Recall: 0.5728
Validation Accuracy – 0.7058
Validation F1: 0.6216 Precision: 0.6622 Recall: 0.59
Testing Accuracy – 0.7253
Testing F1: 0.55 Precision: 0.5933 Recall: 0.5216
Topics/Labels – 20
Vocabulary – 20000 words
Padding – 10000
Learning Rate – 0.1
Activation Functions – Hidden NN layer: tanh Dense output layer: sigmoid
Optimization Functions – Adam's gradient descent optimizaer
Loss function – Binary cross entropy loss
Embedding layer – GloVe 100-dimensional pre-trained
Hidden layer neurons – 100 units
Epochs - 10

12. Conclusion

The MLTC task on TED talks is not an ordinary sentimental analysis task. It involves multiple labels and having the True Positives correct is a challenge. After doing a thorough theoretical study and then an empirical analysis for the MLTC task on TED talks data set we managed to achieve a bit above average result. The training, validation, and testing accuracy curve was almost equal and above average. This proves that our model does not suffer from overfitting or underfitting. The gap between precision and recall was managed to be kept as low as possible. The data set has high variance and is scarce. The data contains direct speech which does not directly relate to the core topic which deviates the learning. The data suffered from noise where few transcripts were missing and redundant. Still, we performed a thorough EDA, pre-processing, post-processing, and hyperparameter tuning to elevate the performance multifold. We also introduced inductive bias wherever we could to effectively utilize the dataset. We performed cross validation to reduce bias, since the dataset was small. Few other creative approaches to balance the evaluation matrix were incorporated. The experimental results were documented for future references.

We are of the opinion that the sentence embeddings should work better. Given the data set if combined with transcripts on other platforms – may be YouTube and using the TED talk as a test dataset would output better results. It can be concluded that the MLTC task on a challenging data set as TED is possible and can be further improved in future.

13. Summary

NLP is a strong tool to handle tedious tasks such as text labeling and summarization. There has been tremendous research hinged about tackling these issues. The project gave us a deeper insight on the underlying challenges and made us think in the right direction. The project also gave us a better understanding of Neural Networks and the Embedding techniques.

References

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Stanford University. GloVe: Global Vectors for Word Representation. Link

[2] Yun Chen Bo Xiao, Zhiqing Lin, Cheng Dai, Zuochao Li, Liping Yan. 2018. Multi-label Text Classification with Deep Neural Networks. (2018)(IEEE). Link

[3] Vedosi Prace, Santosh Kesiraju. Brno University. 2019. Topic Identification from spoken TED talks. Link

[4] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Elizabeth D. Trippe, Juan B. Gutierrez, Krys Kochut.
University of Georgia. July 2017. Text Summarization Techniques: A Brief Survey.
https://arxiv.org/pdf/1707.02268.pdf

[5] Josef Steinberger and Karel Jezek. University of West Bohemia Pilsen Czech Republic. March 2009. Evaluation Measures for Text Summarization,
https://www.researchgate.net/publication/220106310_Evaluation_Measures_for_Text_Summarization

[6] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Bing Xiang. IBM Watson. August 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond.
https://arxiv.org/pdf/1602.06023.pdf

[7] Wesley T. Chuang and Jihoon Yang, Computer Science Department, UCLA, Los Angeles, CA 90095, USA. Extracting Sentence Segments for Text Summarization: A Machine Learning Approach.
https://dl.acm.org/doi/pdf/10.1145/345508.345566

[8] Teresa Goncalves. Paulo Quaresma. 2004. The impact of NLP techniques in the text multilabel classification problem. Link

[9] Anthony Roussea. Poul Deleglis. Yannik Esteve. University of Le Mans, France. 2014. Enhancing the TED-LIUM Corpus with Selected Data for Language Modeling and More TED Talks. Link

Contributions

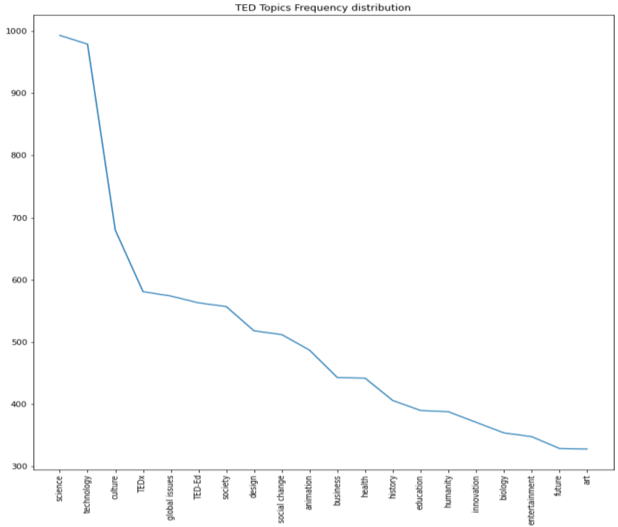
Both the authors of this paper have contributed equally in every aspect of the project be it theoretical study or coding.

Acknowledgements

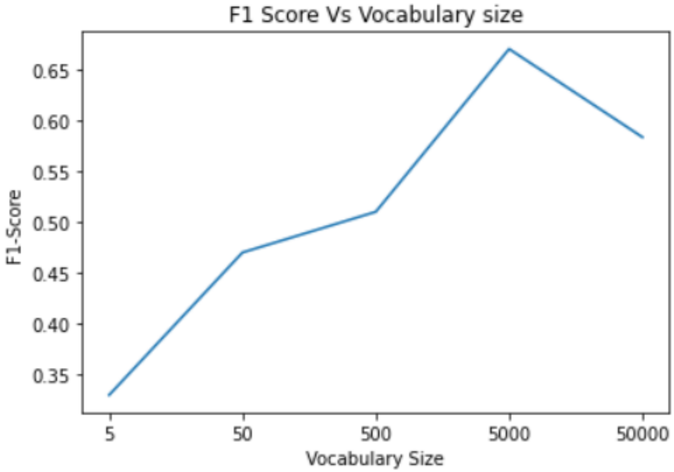
We acknowledge our professor Dr. Jana Doppa for imbibing in us a proper thought process to handle any machine learning task. We would like to acknowledge Jason Brownlee for his excellent blogs on machine learning. Link

Appendix – Results

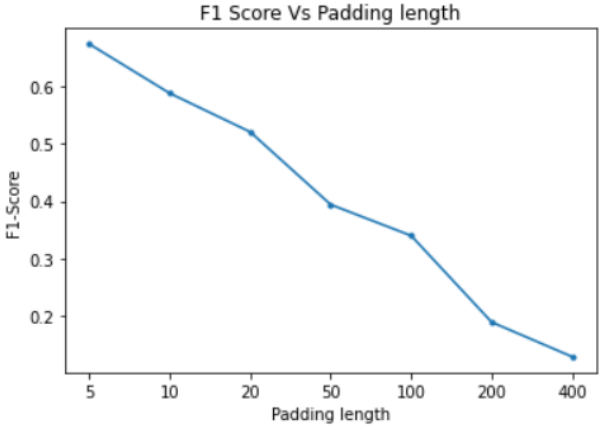
Plot 1 – Topic frequency distribution – Top 20



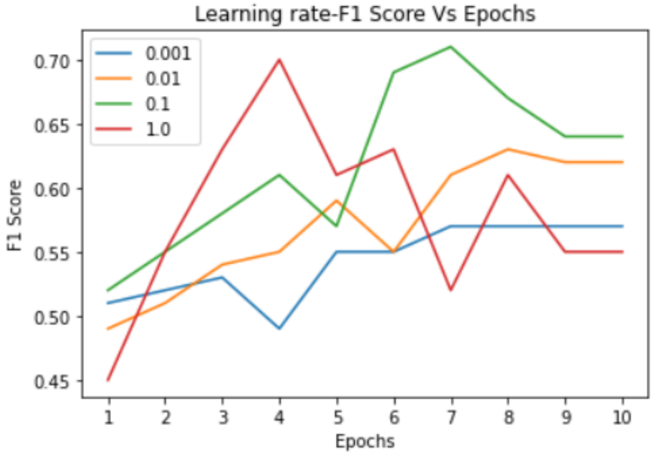
Plot 2 – F1 score vs Vocabulary Size



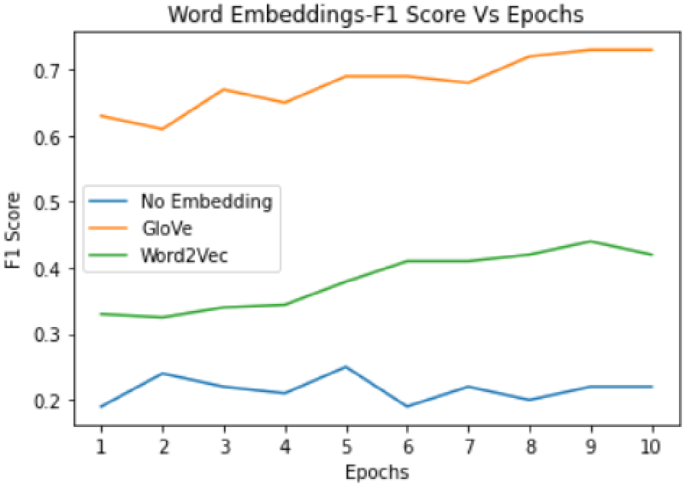
Plot 3 – F1 score vs Padding length



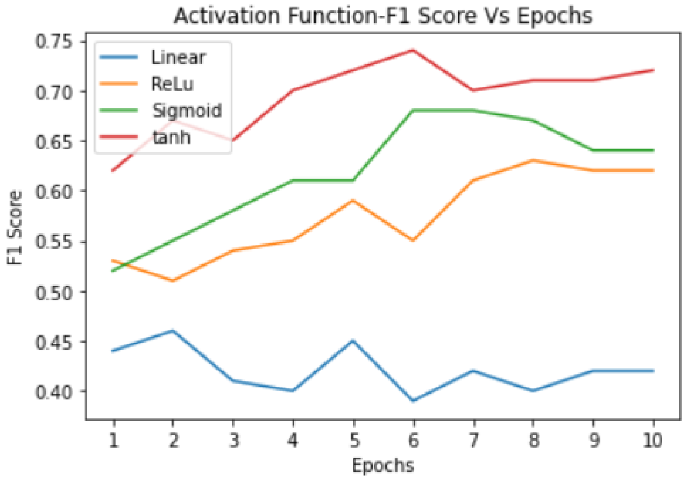
Plot 4 – Learning rate vs F1 score



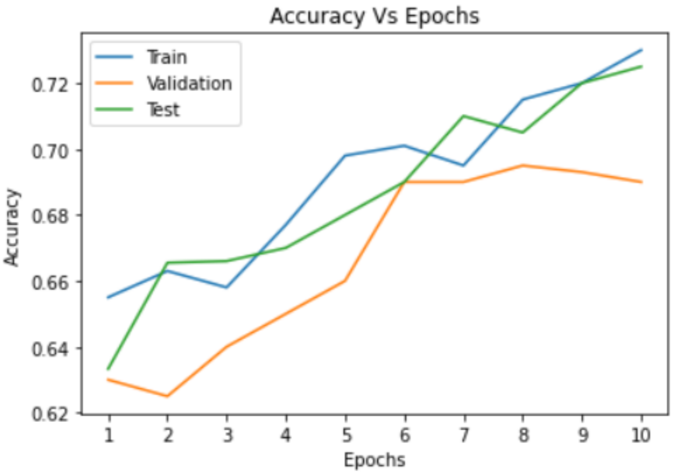
Plot 5 – Activation Function - F1 score vs Epochs



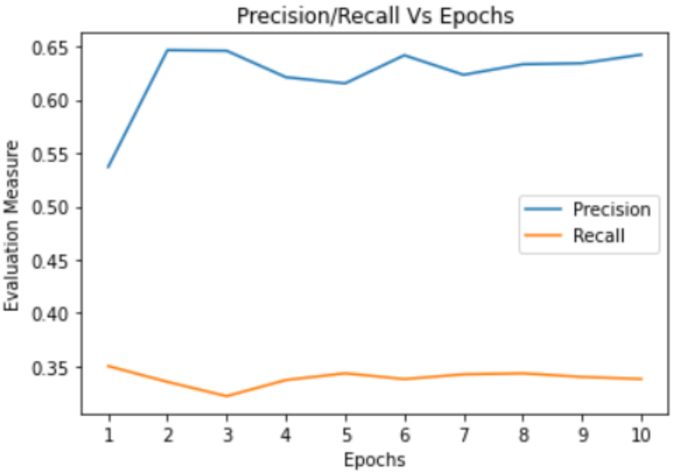
Plot 6 – Embedding layers - F1 score vs Epochs



Plot 7 – Train/Validation/Test Accuracy vs Epochs



Plot 8 – Precision/Recall vs Epochs


Precision/Recall Vs Epochs
Precision
Recall
Evaluation Measure
Epochs