CONVOLVE EPOCH-2

SENTIMENT ANALYSIS OF VERBATIM FEEDBACK

January 3, 2023

Team: IITMOP
Indian Institute of Technology Madras

Shankar Balajee

Niveath Arul

Srinivasan Kidambi

Shivasurya C.M.

Contents

List of Figures

1 Introduction

Sentiment Analysis is the process of computationally identifying and categorizing the emotional tone of a text. It is used in a variety of fields, from marketing to customer service, in order to better understand the attitude of customers, employees, and others towards a company or product. Sentiment Analysis can also be used to track public opinion on political issues or to measure the public's reaction to a news event. There are a number of ways to perform this, however all of them agree upon a fundamental principle, identifying and classifying positive/negative phrases in a sentence. Some methods also assign weights to quantify the ”tone” of the word as such.

2 Problem Statement

Given a labelled dataset of verbatim feedback from various domains, create a working ML/NLP module that would analyse Verbatim Feedback from customers and classify them as positive or negative.

2.1 Description of the dataset

The train data-set has 387 data-points each comprising Job Role, Feedback and Sentiment(1/0). The test data set consists of 130 data-points, for which we have to produce the correct sentiment using our model.

3 Literature review

- Attention Is All You Need
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- RoBERTa: A Robustly Optimized BERT Pretraining Approach
- DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques
- TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification

4 Approach

Our approach was to use a pipeline in which first we encode the Verbatim Feedback using a transformer and make numerical embeddings of words, followed by a classification model that would predict the Sentiment of the Sentence.

5 Model Ideation

The broad steps in our methodology of approach to this problem statement included:

5.1 Embeddings

- Embeddings are a technique used in natural language processing (NLP) to map text to a low-dimensional space. This is done by training a neural network to learn the association between words and a set of numerical features by capturing the meaning and context of each word.
- We used an open-source library called Flair to provide us with various text-embeddings and state of the art models for text-classification. Initially we used a FlairEmbedding, but the embedding was not able to capture the features of contextual text well enough.
- Following this attempt, we ideated upon the usage of Transformers as a solution, given their robust and easy-to-train nature.
- Post various trials on multiple transformer architectures, we finalized on customized RoBERTa embeddings. This decision was guided by the fact that the length of ”Verbatim Feedback” in the dataset is best matched by length of tweets.
- After the generation of suitable word-embeddings, the next step was to make document embeddings which convert the word embeddings into a single embedding for the entire document which in the case is our feedback message.
- We tried both GRUs and LSTMs, and decided to go with the latter given its ability to retain information for a longer distance, in this context.
- This embedding is passed on to the classifier discussed below.

5.2 Classifier

- We initially tried a simple LSTM-based architecture wherein we trained our own word embeddings, followed by a simple MLP classifier. However the model was not robust enough for contextual analysis.
- An attempt at utilising gradient boosting techniques (XGBoost and RandomForest) resulted in sub-par accuracy (F_1 score of 0.603).
- To achieve better scores, the use of a Neural Network classifier was considered. A few dropout layers were added to aid the process of training.

6 Training and Results

Hyper-Parameter Tuning

- Optimizer: AdamW
- Learning-Rate
  - Best convergence was observed with a learning rate of 10^{-6}.
  - Higher learning rate led to over-fitting on the training data. Results are illustrated below.

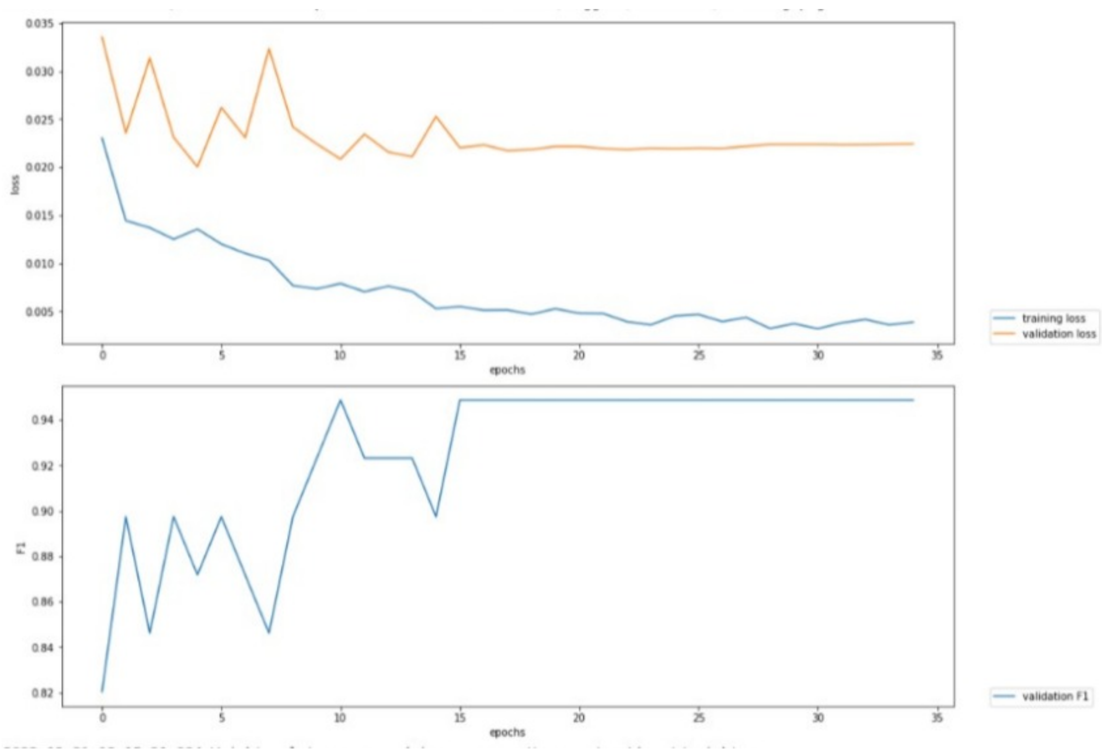

Figure 1: Case of Over-fit model with a learning rate of 0.005

- Scheduler
  - Initially, AnnealonPlateauLR was used with an anneal factor of 0.5 and patience of 3 where the maximum validation F_1 score observed without overfitting was 0.9231.
  - In an attempt to outperform the above mentioned method, CosineAnnealing was considered as an option. This approach yielded the expected outcome, with a score of 0.9744 far surpassing all previous implementations.

Results

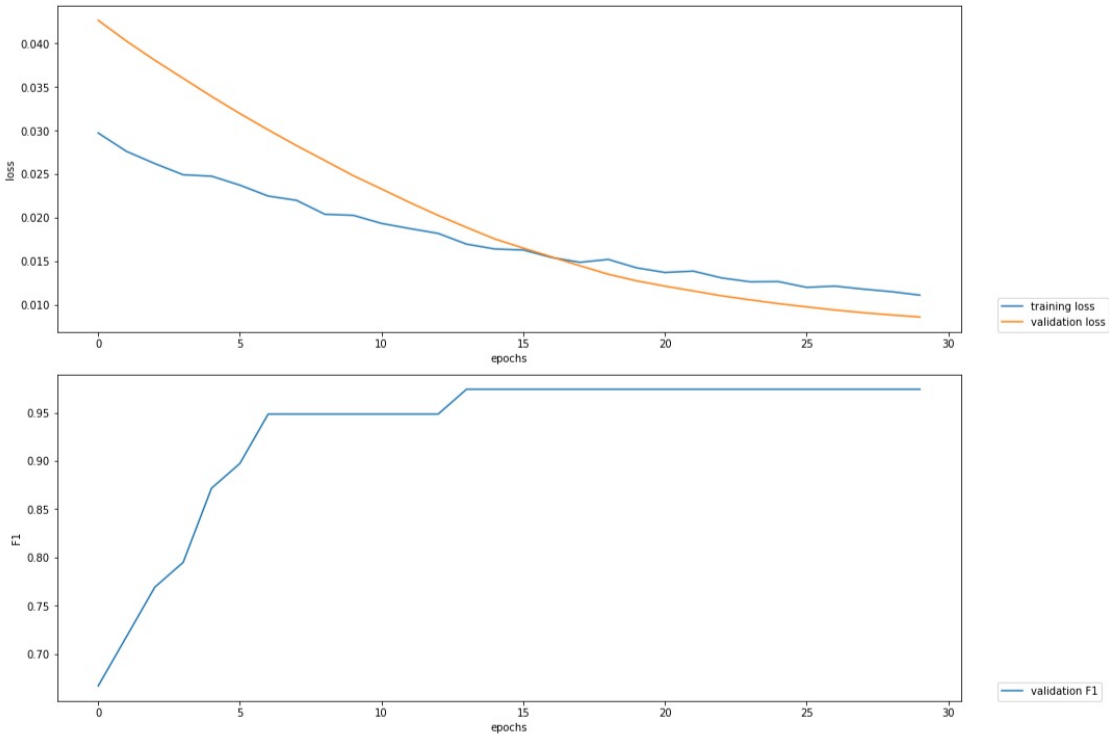

Figure 2: Plot showing the training curve of our best model