DL and GenAI Project Report

Hariharan R, Diploma in Data Science – IITM BS, 21f2000120

1. Executive summary

The problem that has been worked in this project is a multi label classification problem in sentiment analysis. Finetuning pre-trained models is the current state of the art in sentiment analysis. Hence finetuning a range of pretrained models such as BERT, RoBERTa and ALBERT have been experimented. A custom transformer model and LSTM model have also been implemented for comparative analysis. Among all implementations the fine-tuned RoBERTa model has the highest Macro-F1 score (0.876) on the test set. A model that has a larger vocabulary size and has been pretrained on a lot of data is able to perform better in unseen cases.

2. Introduction

2.1 Problem statement : Emotions are complex, often overlapping, and can be expressed in subtle ways through language. Detecting them is an important task in Natural Language Processing (NLP) with applications in mental health support, customer experience analysis, conversational AI etc. In this competition, the challenge is to build models that can classify short text entries into multiple emotion categories. The dataset contains five emotions : anger, fear, joy, sadness, surprise. Each label is binary and each text may have multiple emotions at once. The goal is to predict the correct set of emotions for the unseen test data.

2.2 Project objective : The goal is to train a model that will have atleast 80% performance in the test set. Experimentation with pretrained models and different architectures that are available is to be done to gain a certain exposure in using libraries such as hugging faces and pytorch.

2.3 Report structure : The further sections involve summary of the dataset, exploratory data analysis, data preprocessing and tokenization steps that were taken, modelling and experimentation that were done as a part of exploration, results analysis, conclusion and future works.

3. Dataset & Preprocessing

3.1 Dataset Description : The train dataset has text and 5 emotions. The test dataset contains only the text and the emotion set have to be predicted.

3.2 Exploratory Data Analysis (EDA): The train and test datasets do not contain any null values. The train dataset has 6827 entries and the test dataset is 25 % of the train dataset. When we look at the correlation coefficients as shown in Figure 1, Fear and sadness have a positive correlation of 0.3, Fear and surprise have a positive correlation of 0.2 and Joy has strong negative correlation towards fear and sadness.

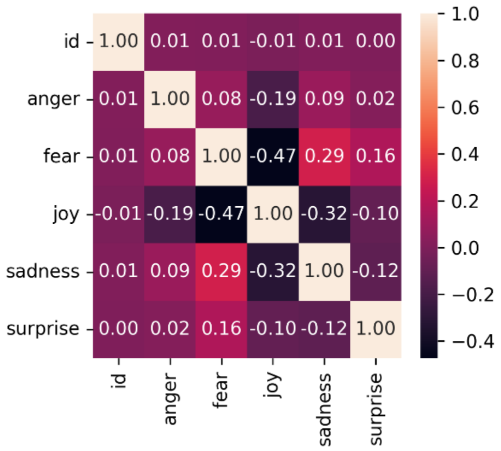

Figure 1 : Correlation Heatmap between Emotions

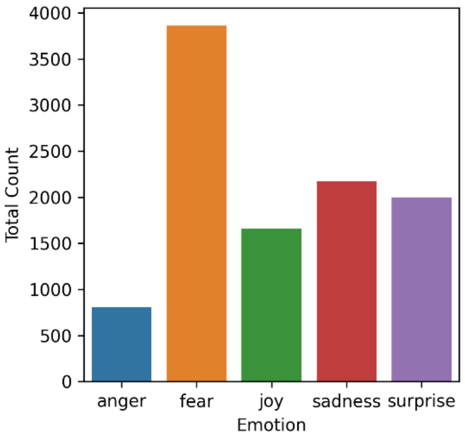

Figure 2 : Emotion Frequency Distribution

Among the Emotions anger seems to have appeared less number of times as shown in Figure 2. The maximum length of the string is 450 and the minimum length of the string is 5

3.3 Data Preprocessing: Using regular expression, text is lowercased, html tags are removed, websites have been removed, only lower case letter and spaces have been kept, extra spaces have also been removed.

4. Tokenization strategy

4.1 Primary Tokenizer Selection : The best performing model has been RoBERTa, Byte Pair Encoding (BPE) is the encoding strategy that has been used. RoBERTa is Robustly Optimized BERT Approach developed by Facebook AI. It improves BERT by training longer with large datasets and removing next sentence prediction. BPE splits words into subword units to handle rare words, misspellings.

4.2 Implementation Details : Tokenizer is got from the AutoTokenizer library of hugging faces. Max length of 64 has been used, padding and truncation are used.

4.3 Experimental Comparison : Varying max lengths has been experimented to find a optimal choice between computational cost and model performance. Max length value of 64 has been chosen as the optimal max length from Figure 2 and Table 1.

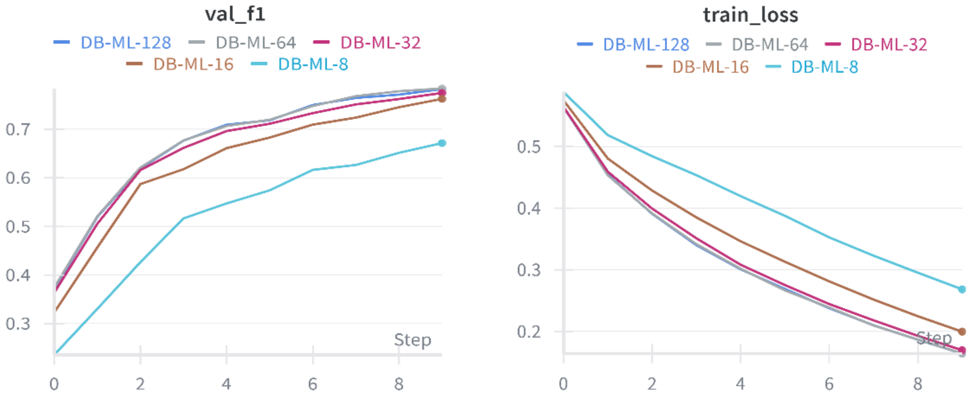

Figure 3 : Comparison of Validation F1 score and Train loss reduction for different ML DB(DistilBERT), ML(Max Length)

Max Length – DistilBERT	Val F-1 at 10th Epoch	Training Time
128	0.7825	6 minutes 7 seconds
64	0.7838	3 minutes 33 seconds
32	0.7745	2 minutes 23 seconds
16	0.7623	1 minute 47 seconds
8	0.6713	1 minute 41 seconds

Table 1 : Comparison of Validation F1 and Training Time for different Max Lengths

The results were similar with DistilRoBERTa model and max length of 64 remains the optimal choice.

5. Modeling & Experimentation

5.1 Model 1 : Fine-Tuned Transformer Model

5.1.1 Architecture : Transformer with L layers. Each block uses A self-attention heads and hidden dimension H. RoBERTa has L=24, A=16, H=1024, 355M parameters (Liu et al., 2019).

5.1.2 Salient Points : RoBERTa uses the encoder of transformer which is good at semantic understanding for classification. It uses multi-head attention and is better at capturing long range dependencies than LSTM.

5.1.3 Fine-Tuning Strategy : The optimizer used was AdamW, with a learning rate of 0.05 and a linear schedule with 10% warmup, for 15 epochs with early stopping on val_F1 score (patience =10) with the pretrained model layers frozen. The pretrained model layers were then unfrozen and trained with a learning rate of 5e-5 for another 15 epochs keeping other parameters constant.

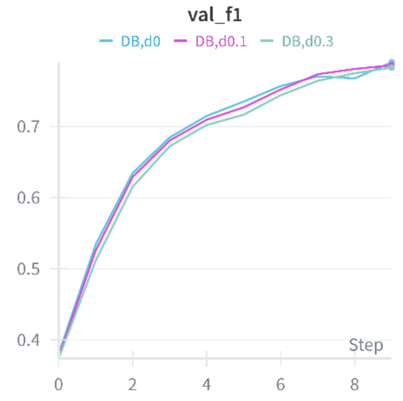

Figure 4 : Varying dropout percentage Val F1 score

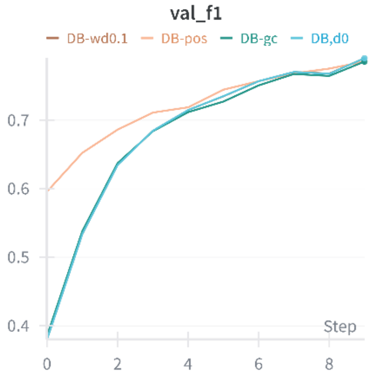

Figure 5 : Comparison of pos weight and gradient clip

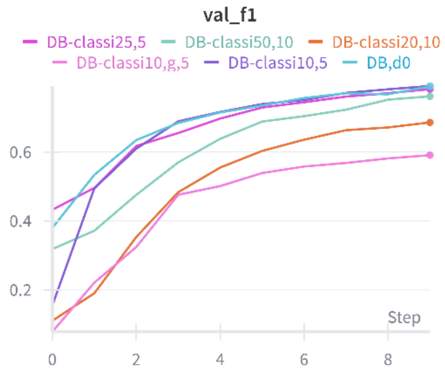

Figure 6 : Different classifier heads performance

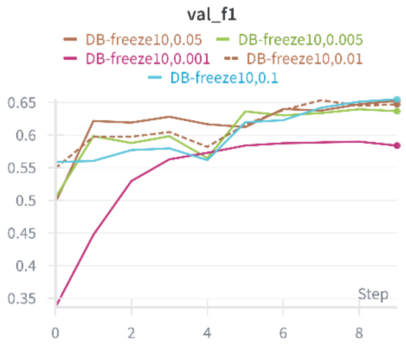

Figure 7 : Different learning rates for training classifier

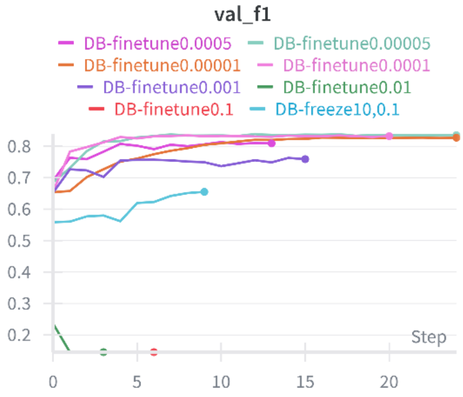

Figure 8 : Different learning rates for finetuning

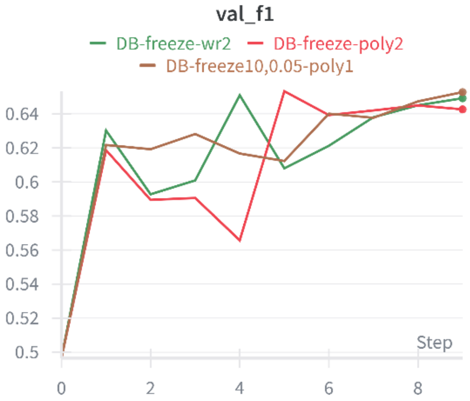

Figure 9 : Different learning rate schedules

Figure	Inference
4	No Dropout has maximum macro F1
5	No gradient clipping, no Pos weight has maximum macro F1
6	Classifier with 10, 5 head performs better than just 5
7	Learning rate of 0.05 is optimal for training classifier
8	Learning rate of 5e-5 is optimal for finetuning
9	Linear schedule is optimal

Table 2 : Inference from experimentation

5.2 Model 2 : Custom Deep Learning Model

5.2.1 Architecture : A Custom Transformer Encoder

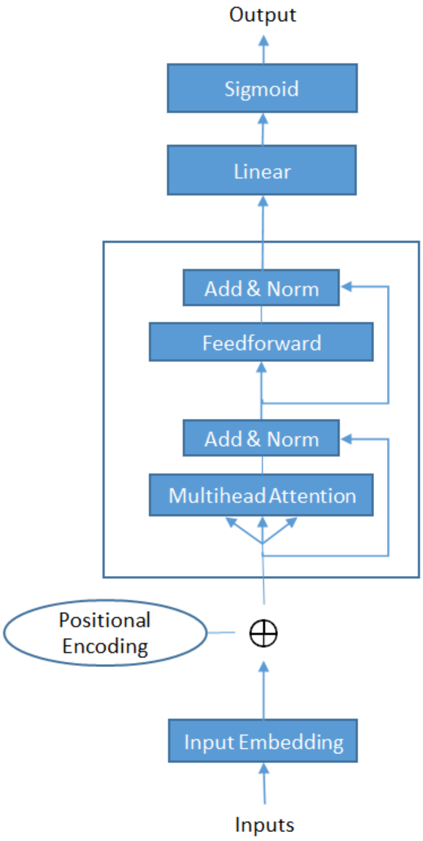

Figure 10 : Custom Transformer Architecture

5.2.2 Salient Points : Embedding dimension = 16, d_model = 16, number of heads= 2, number of layers = 1, Dimension Feed Forward = 16. A small model is able to generalize sufficiently (above 0.7) as the train set has less data.

5.3 Model 3 : RNN-Based Model

5.3.1 Architecture : Bi-LSTM

5.3.2 Salient Points : It is implemented on a custom vocabulary, removing selected stopwords. Embedding dimension = 2000 and hidden dimension = 2000.

6. Performance & Comparative Analysis

6.1 Evaluation Metrics : Macro F1 Score

F1 score is the harmonic mean of precision and recall

$$F1 = \frac{2.Precision.Recall}{Precision + Recall}$$

The F1 score is computed for each label individually and the unweighted average is calculated.

$$Macro\ F1 = \frac{1}{N}\sum_{i=1}^{N} F1_i$$

6.2 Training Performance :

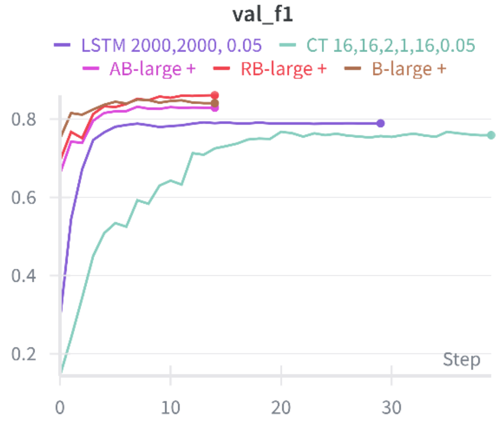

Figure 11 : Validation Macro F1 Comparison

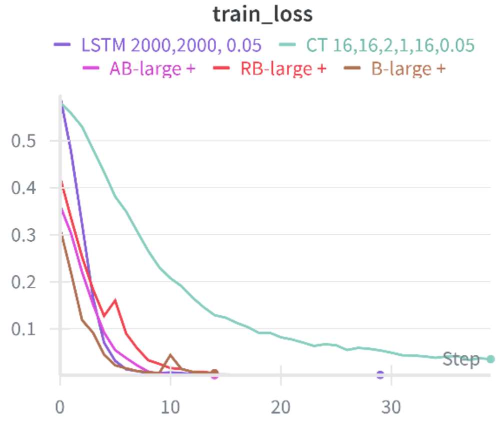

Figure 12 : Train loss reduction comparison

RoBERTa Large has the highest Macro F1 Score on the validation set and the train loss is also minimal

6.3 Comparative Report :

Model	Total	Finetune	Classifier training	Total Epochs	Per Epoch Training Time (sec)	Maximum Val F1	Minimum Train loss
ALBERT Large	39 m 51 s	27 m 47 s	12 m 4 s	15+15	111.1/48.2	0.832	0.001
RoBERTa Large	45 m 28 s	32 m 10 s	13 m 18 s	15+13	128.6/61.3	0.861	0.005
BERT Large	41 m 45 s	31 m 17 s	10 m 28 s	15+10	125.1/62.8	0.850	0.006
LSTM	22 m 6 s	-	-	30	44.2	0.792	0.002
Custom Transformer	1 m 17 s	-	-	40	1.925	0.767	0.035

Table 3 : Performance Comparison of different models

6.4 Kaggle Performance : The final score achieved is 0.876 with a rank of 42

7. Conclusions & Future Work

7.1 Key Learnings : Pretrained models outweigh custom models and fine tuning requires a lot of experimentation.

7.2 Challenges Faced : Experimenting with large pretrained models is computationally expensive. Experimenting with distilled models was a good choice as they train faster and is a representative of the large model.

7.3 Areas for Improvement : Experimenting with larger models such as Albert x large and Albert xx large could lead to improvements. Using upsampling techniques for rare emotion could also lead to improvements. Exploring ensemble methods could improve the macro-f1 score. Training a model with pretrained backbone and 5 heads could also improve the macro-f1 score.

8. References

8.1 Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
arXiv:1810.04805, 2018.

8.2 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al.
RoBERTa: A Robustly Optimized BERT Pretraining Approach.
arXiv:1907.11692, 2019.

8.3 Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R.
ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.
arXiv:1909.11942, 2019.

8.4 Sanh, V., Debut, L., Chaumond, J., & Wolf, T.
DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.
arXiv:1910.01108, 2019.

8.5 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al.
Attention Is All You Need.
NeurIPS, 2017.

8.6 Harris, C. R., Millman, K. J., van der Walt, S. J., et al.
Array programming with NumPy.
Nature, 585, 2020.

8.7 Paszke, A., Gross, S., Massa, F., et al.
PyTorch: An Imperative Style, High-Performance Deep Learning Library.
NeurIPS, 2019.

8.8 McKinney, W.
Data Structures for Statistical Computing in Python.
Proc. of the 9th Python in Science Conference, 2010.

8.9 Van Rossum, G., & Drake, F. L. *Python 3 Reference Manual.* CreateSpace, 2009.

8.10 Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al.
Transformers: State-of-the-Art Natural Language Processing.
arXiv:1910.03771, 2020.

8.11 Biewald, L.
Experiment Tracking with Weights and Biases.
wandb.com, 2020.