Embeddings at BLP-2023 Task 2: Optimizing Fine-Tuned Transformers with Cost-Sensitive Learning for Multiclass Sentiment Analysis

S.M Towhidul Islam Tonmoy
Islamic University of Technology, Gazipur , Bangladesh
towhidulislam@iut-dhaka.edu

Abstract

In this study, we address the task of Sentiment Analysis for Bangla Social Media Posts, introduced in first Workshop on Bangla Language Processing (Hasan et al., 2023a). Our research encountered two significant challenges in the context of sentiment analysis. The first challenge involved extensive training times and memory constraints when we chose to employ oversampling techniques for addressing class imbalance in an attempt to enhance model performance. Conversely, when opting for undersampling, the training time was optimal, but this approach resulted in poor model performance. These challenges highlight the complex trade-offs involved in selecting sampling methods to address class imbalances in sentiment analysis tasks. We tackle these challenges through cost-sensitive approaches aimed at enhancing model performance. In our initial submission during the evaluation phase, we ranked 9th out of 30 participants with an F1-micro score of 0.7088 . Subsequently, through additional experimentation, we managed to elevate our F1-micro score to 0.7186 by leveraging the BanglaBERT-Large model in combination with the Self-adjusting Dice loss function. Our experiments highlight the effect in performance of the models achieved by modifying the loss function. Our experimental data and source code can be found here.[1]

1 Introduction

Sentiment analysis is an important task in natural language processing that involves automatic detection of expressed opinions within text. The proliferation of online social media interactions has led to a surge in textual content, necessitating strategies to address associated challenges.

While sentiment analysis for high-resource languages like English has made significant progress, low-resource languages like Bangla are still lagging behind. Low resource languages have intricate sentence structures and grammatical rules, making the development of systems resource-intensive. Achieving optimal model performance requires substantial annotated data, leading to longer processing times as data volume increases. Additionally, when performing multiclass sentiment analysis, there is a common challenge related to class imbalance, which can lead to models exhibiting bias towards particular classes. Previous studies have concentrated on improving the quantity of training data instances, although this approach can extend the duration of model training.

Numerous studies have been undertaken to advance the development of linguistic resources for the Bangla language. (Islam et al., 2021) introduced SentNoB dataset for multiclass sentiment analysis task. (Patra et al., 2015) summarized the sentiment analysis task for three Indian language , namely Bangla, Hindi and Tamil. They showed the results for shared task on binary sentiment analysis and introduced the SAIL dataset. (Rezaul Karim et al., 2020) introduced the BengFastText dataset which was able to capture semantics of Bangla words. They experimented their corpus with traditional ML algorithms and also utilized MConv-LSTM network to tackle the binary sentiment analysis task. (Tripto and Ali, 2018) introduced Bangla language corpus from Bangla youtube comments. (Rahman et al., 2018) focused on aspect based sentiment analysis and introduced the research community with ABSA cricket and restaurant datasets. But all of this datasets had class imbalances in their classes. (Hasan et al., 2020) and (Alam et al., 2021) compiled all the previously mentioned datasets and benchmarked their results with different traditional and transformer based models. The ongoing challenge lies in the escalating fine-tuning time due to the increasing data volume. This study seeks to enhance fine-tuned transformer model efficiency

[1] https://github.com/towhidultonmoy/Bangla-Multiclass-Sentiment-Analysis-Shared-Task-.git

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 340–346
December 7, 2023 ©2023 Association for Computational Linguistics

by employing cost-sensitive learning to tackle class imbalance problem. Our contributions can be summarized as follows:

- Cost sensitive learning improves the performance of most of the transformer based models. We perform an extensive series of experiments involving SOTA transformer models, exploring various loss functions.
- The best F1-micro score was achieved with **BanglaBERT-Large** variant combining it with self adjusting dice loss.
- Additionally, we examine the impact of diverse preprocessing techniques on model performance.

2 Related Work

2.1 Sentiment Classification with Deep Learning

In the context of text classification for sentiment analysis in Bangla, researchers have utilized a range of models, from traditional ones to the latest prompt-based large language models (LLMs).

(Rahman et al., 2018) employed SVM, RF, and KNN models to perform ABSA in Bangla. They achieved F1 scores of 0.37 and 0.42, respectively, using TF-IDF features on their ABSA cricket and restaurant datasets. (Rezaul Karim et al., 2020) explored a comprehensive set of models, including LR, NB, SVM, KNN, GBT, RF, MConv-LSTM, and MAE. They achieved impressive results with MConv-LSTM, attaining an MCC of 0.746 and an AUC of 0.87 for sentiment analysis in Bangla using BengFastText embeddings. (Hasan et al., 2023b) delved into zero- and few-shot in-context learning for sentiment analysis in Bangla. They compared Open LLMs like Flan-T5 and GPT-4 against fine-tuned models, where BanglaBERT outperformed others with a weighted F1 of 69.39. They utilized SentNoB and introduced the MUBASE dataset, which included Facebook posts and tweets. (Alam et al., 2021) conducted a comparative analysis of Bangla NLP tasks using transformer models, achieving an 82.0 weighted F1 using XLM-RoBERTa on various publicly available datasets. In their study, (Hasan et al., 2020) conducted comparative sentiment analysis on Bangla text using classical algorithms and deep learning models. BERT and XLM-RoBERTa demonstrated strong performance on different datasets, with an average weighted F1 of 0.671 and 0.653, respectively.

2.2 Handling Class Imbalance

(Hasib et al., 2023b) present a system that employs RUS and SMOTE to balance the dataset. Their approach utilizes a range of machine learning and deep learning models, with BERT reaching a maximum accuracy of 99.04% in balanced datasets and 72.23% in imbalanced datasets. Another noteworthy contribution by (Hasib et al., 2023a) introduces MCNN-LSTM, a novel fusion of CNN and LSTM for news text classification. After balancing the dataset using the Tomek-Link algorithm, their model attains remarkable performance, achieving a 98% F1-score and 99.71% accuracy compared to prior research. (Rafi-Ur-Rashid et al., 2022) address class imbalance using various models for binary sentiment analysis, achieving 0.94 accuracy with their CNN model on the original corpus, employing a comprehensive approach that includes data augmentation, focal loss functions, outlier detection, data resampling, and hidden feature extraction across diverse datasets. Lastly, (Ashrafi et al., 2020) introduce BERT-based deep learning models for Bangla NER while addressing class imbalance with a modified cost-sensitive loss function. Their proposed models yield 8% enhancement in F1 MUC score compared to previous Bangla NER research.

3 Dataset

3.1 Data Description

The dataset for this shared task is a combination of two sources: SentNoB (Hasan et al., 2020) and MUBASE (Hasan et al., 2023b). Table 1 reports the number of samples in the train, validation and test sets for each class. The dataset distribution reveals a noticeable class imbalance across the training, validation, and test sets.

Class	Train	Validation	Test
Negative	15767	1753	3338
Positive	12364	1388	2092
Neutral	7135	793	1277
Total	35266	3934	6707

Table 1: Class-wise Dataset Distribution in Train, Validation, and Test Sets.

4 System Overview

Recent developments in NLP have seen the emergence of pre-trained transformer models, based on the transformer architecture proposed by (Vaswani et al., 2017). These models consistently achieve state-of-the-art performance across a wide range of NLP tasks.

In our study, we initially fine-tuned multiple pre-trained transformer models using the default cross-entropy loss as our baseline approach. Subsequently, we aimed to enhance model performance through cost-sensitive learning, which effectively addresses class imbalances and mitigates biases towards the majority classes.

4.1 Finetuning Pre-trained Language Models (PLMs)

We selected various pre-trained models and fine-tuned them for our baseline. These models include Bangla-Bert (Bhattacharjee et al., 2022), Bangla-GPT2(Flax Community, 2023), Indic-BERT (Kakwani et al., 2020) and mBERT (Devlin et al., 2018). We employed cross-entropy loss and the AdamW optimizer for fine-tuning. Details regarding the hyperparameter values used for training the baseline and subsequent models can be found in the Appendix.

4.2 Cost Sensitive Learning

A prominent challenge we encountered with our dataset was class imbalance, a common issue in machine learning tasks. However, conventional methods like oversampling and undersampling were not feasible in our case due to their drawbacks, which involve increased training times and reduced performance, respectively. Thus, we explored the hypothesis that modifying the loss function could potentially enhance model performance without the need for additional data.

To elevate our model's performance beyond the baseline, we introduced various loss functions, namely, the self-adjusting dice loss (Li et al., 2019), focal loss (Lin et al., 2017), and F1-micro loss. These alternative loss functions were employed as part of our strategy to address class imbalance and improve overall model performance. Details about this loss functions are mentioned in the appendix C

5 Experiments and Results

We explored various model and custom loss function combinations as described in Section 4. In this section, we outline the evaluation for the shared task competition, with the F1-micro score as the key performance metric. Our model assessments were conducted on the test set, and, as outlined in Section 6, we noted improved model performance without text preprocessing as mentioned in appendix 6.2. Table 2 presents the test set results, trained upon dataset B . Details about the dataset are mentioned in A.

In our initial experimentation with transformer models, we fine-tuned each model using the default cross-entropy loss function. Among the models in our baseline study, BanglaBERT-Large stood out, achieving the highest F1-micro score of 0.7101. Subsequently, we investigated the impact of cost-sensitive loss functions on model performance. We implemented focal loss, self-adjusting dice loss, and F1-micro loss. Notably, for two models, BanglaBERT-Large and mBERT, these alternative loss functions led to significant improvements compared to the baseline approach.

For BanglaBERT-Large, self-adjusting dice loss produced the best result, with an F1-micro score of 0.7186, surpassing all other transformer models used in our research. For mBERT, focal loss resulted in improved performance, achieving an F1-micro score of 0.6606. Other loss functions for these two models also outperformed the baseline, as shown in the table 2.

However, for BanglaGPT2, incorporating cost-sensitive loss functions did not enhance model performance; the baseline approach yielded the highest F1-micro score at 0.6788. Regarding the IndicBERT model, self-adjusting dice loss improved performance compared to the baseline cross-entropy loss, achieving an F1 score of 0.6263. However, focal loss and F1-micro loss did not yield performance improvements for this model.

6 Ablation Study

In the scope of our study, we conducted a sequence of experiments to understand key factors affecting our model's performance.

6.1 Impact of Combining Training and Validation Set

To evaluate the merging of training and development sets, we analyzed two datasets: Dataset A and Dataset B (the consolidated dataset). We then assessed their impact on the designated test dataset. Appendix A offers a detailed data distribution analysis for both datasets, and Table 3 summarizes the effect of these datasets on the performance of the most promising combinations from Table 2.

Label	Word Unigram Overlap
Negative	খারাপ (Bad), দোষ (Fault), ধর্ষণ (Rape), নিষিদ্ধ (Prohibited), যুদ্ধ (War), হামলা (Attack), গুম (Disappearance), ভুয়া (Fake), ধ্বংস (Destroy), প্রত্যাহার (Withdrawal), কষ্ট (Suffering), হত্যা (Murder), শান্তি (Peace), উন্নয়ন (Development), অবৈধ (Illegal), ভয় (Fear), ধন্যবাদ (Gratefulness), পরিবর্তন (Change), প্রাণহানি (Homicide), অভিযোগ (Complaint)
Neutral	ভয় (Fear), গুরুত্বপূর্ণ (Important), ভুল (Mistake), জয় (Victory), ঘুম (Sleep), হামলা (Attack), খারাপ (Bad), ধ্বংস (Destruction), উন্নয়ন (Development), গুম (Loss), ধর্ষণ (Assault), অবৈধ (Illegal), ভুয়া (Destruction), ধন্যবাদ (Gratefulness), দোষ (Fault), যুদ্ধ (War), কষ্ট (Suffering), প্রিয় (Favorite), আলহামদুলিল্লাহ (Gratitude), সুন্দর (Beautiful)
Positive	পরিবর্তন (Change), ঘুম (Sleep), প্রিয় (Favorite), আলহামদুলিল্লাহ (Gratitude), শান্তি (Peace), হত্যা (Murder), সুন্দর (Beautiful), খারাপ (Bad), উন্নয়ন (Development), গুরুত্বপূর্ণ (Important), ধ্বংস (Destruction) , খারাপ (Bad), ধন্যবাদ (Gratefulness) , নিষিদ্ধ (Prohibited), প্রাণহানি (Homicide), অভিযোগ (Complaint), নিষিদ্ধ (Prohibited), প্রত্যাহার (Withdrawal), যুদ্ধ (War), জয় (Victory)

Figure 1: Example of word unigram overlaps among label categories with English translations. Here distinct colors are used to emphasize concurrent words: 🟩 color denotes common words across **all** labels, 🟥 denotes common words between **Negative** and **Neutral** labels, 🟦 color denotes common words between **Negative** and **Positive** labels, and 🟨 denotes common words between **Neutral** and **Positive** labels.

Model	Loss Function	F1
BanglaBERT	Cross Entropy Loss	0.7101
	Focal Loss	0.7177
	SA Dice Loss	**0.7186**
	F1 Micro Loss	0.7126
Bangla GPT2	**Cross Entropy Loss**	**0.6788**
	Focal Loss	0.6757
	SA Dice Loss	0.6569
	F1 Micro Loss	0.6707
mBERT	Cross Entropy Loss	0.6497
	Focal Loss	**0.6606**
	SA Dice Loss	0.6528
	F1 Micro Loss	0.6581
IndicBERT	Cross Entropy Loss	0.6166
	Focal Loss	0.6062
	SA Dice Loss	**0.6263**
	F1 Micro Loss	0.6145

Table 2: F1-micro score on the Competition Test Set for Various Transformer Models Trained with Dataset B

6.2 Impact of Different Text Processing Techniques

In our study, we performed two crucial text preprocessing steps: **1)** removing emojis and **2)** eliminating punctuation marks. We assessed the effects of each step independently and when applied together. We've summarized the results in Table 4, using the acronyms: **P1** (for Step 1), **P2** (for Step 2), **All** (for Both Steps), and **None** (for No Preprocessing). This analysis sheds light on how these preprocessing methods impact our research outcomes.

Model	Loss Function	Dataset	F1
BanglaBERT	SA Dice Loss	A	0.7067
		B	**0.7186**
Bangla GPT2	Cross Entropy Loss	**A**	**0.6833**
		B	0.6788
mBERT	Focal Loss	A	0.6446
		B	**0.6606**
IndicBERT	SA Dice Loss	A	0.6230
		B	**0.6263**

Table 3: Impact of Diverse Datasets on Optimal Transformer Model Combinations. **Dataset A**: Original Training Set, **Dataset B**: Combined Train and Validation Sets.

7 Error Analysis

Table 2 present the performance results of BangaBERT-Large, which, notably, outperformed all other methods in our experiments. This section delves into a quantitative error analysis employing a confusion matrix, as displayed in Figure 2, focusing on the top-performing model. Our analysis reveals a distinct pattern of misclassification occurring primarily between the 'neutral' and 'negative' classes.

In Appendix D, Table 7 demonstrates the subpar performance observed in the 'neutral' class. Despite our diligent efforts to mitigate class imbalance

Dataset	BanglaBERT with SA Dice Loss
P1	0.7182
P2	0.7088
All	0.7106
None	**0.7186**

Table 4: F1-micro score for Different Preprocessing Techniques on Dataset B: Combined Train and Validation Sets

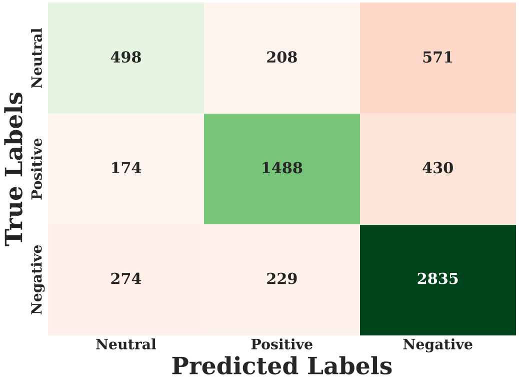

Figure 2: Confusion Matrix of Best Performing Model

through a cost-sensitive loss function, the model continues to encounter difficulties in distinguishing between 'neutral' and 'negative' labels.

Furthermore, this misclassification is influenced by semantic similarities between words across different classes. Figure 1 visually represents the common unigrams across various labels, highlighting the areas where the model exhibits errors, especially when there are concurrent words between the 'negative' and 'neutral' labels.

8 Conclusion

This research paper primarily emphasizes the enhancement of transformer-based models' performance through the application of cost-sensitive learning techniques, aimed at alleviating issues related to class imbalance and overfitting. Among various combinations of transformers and loss functions explored, the BanglaBERT model utilizing the self-adjusting dice loss exhibited the highest F1 score of 0.7186 on the test dataset. Although the combination of cost-sensitive techniques with transformer models led to notable enhancements in performance, it's important to highlight that the model's effectiveness still falls short, especially when it comes to the 'neutral' class.

Limitations

In this research, we chose a cost-sensitive approach as an alternative to augmentation of the training dataset, recognizing its resource-intensive demands in GPU resources and training time. Our objective was to investigate how modifying loss functions could improve the performance of fine-tuned transformer models, presenting a more resource-efficient route to better outcomes.

Despite our experiments demonstrating several strategies for enhancing fine-tuned transformer model performance, we acknowledge the model's ongoing challenge in accurately classifying less frequent classes. This limitation directs our future research towards optimizing loss function hyperparameters and assessing their effectiveness across various model architectures and datasets as a promising avenue for improvement.

References

Firoj Alam, Md Arid Hasan, Tanvir Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Imranul Ashrafi, Muntasir Mohammad, Arani Shawkat Mauree, Galib Md Azraf Nijhum, Redwanul Karim, Nabeel Mohammed, and Sifat Momen. 2020. Banner: a cost-sensitive contextualized model for bangla named entity recognition. *IEEE Access*, 8:58206–58226.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Flax Community. 2023. gpt2-bengali (revision cb8fff6).

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment classification in bangla textual content: A comparative study. In *23rd International Conference on Computer and Information Technology (ICCIT)*.

Khan Md Hasib, Sami Azam, Asif Karim, Ahmed Al Marouf, FM Javed Mehedi Shamrat, Sidratul Montaha, Kheng Cher Yeo, Mirjam Jonkman, Reda Alhajj, and Jon G Rokne. 2023a. Mcnn-lstm: Combining cnn and lstm to classify multi-class text in imbalanced news data. *IEEE Access*.

Khan Md Hasib, Nurul Akter Towhid, Kazi Omar Faruk, Jubayer Al Mahmud, and MF Mridha. 2023b. Strategies for enhancing the performance of news article classification in bangla: Handling imbalance and interpretation. *Engineering Applications of Artificial Intelligence*, 125:106688.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Proc. of MIKE*, pages 650–655. Springer.

Md Rafi-Ur-Rashid, Mahim Mahbub, and Muhammad Abdullah Adnan. 2022. Breaking the curse of class imbalance: Bangla text classification. *Transactions on Asian and Low-Resource Language Information Processing*, 21(5):1–21.

Md Rahman, Emon Kumar Dey, et al. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Md Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced Bengali language based on multichannel convolutional-lstm network. *arXiv*, pages arXiv–2004.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *Proc. of ICBSLP*, pages 1–6. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Dataset

We conducted experiments using two datasets, A and B, as described in our ablation study. The table 5 shows the number of examples in each split.

For dataset A, we utilized the original training and test sets. In dataset B, we combined the training and validation sets into a single unified training set, while keeping the test set unchanged.

Split	Class	Dataset A	Dataset B
Train	Negative	15767	17520
	Positive	12364	13752
	Neutral	7135	7928
Test	Negtaive	3338	3338
	Positive	2092	2092
	Neutral	1277	1277

Table 5: Class wise Dataset Distribution in Dataset A and Dataset B

B Model Training

In this section, we provide the hyperparameter values we used during fine tuning our models to facilate the reproducibility of our results at a later time. The acronyms correspond to:

- **LR** : Learning Rate
- **BS** : Batch Size
- **EP** : Epoch
- **WD** : Weight Decay
- **MP** : Mixed Precision

- **TML** : Tokenizer Max Length
- **ES** : Early Stopping
- **ESP** : Early Stopping Patience
- **FL** : Focal Loss (Gamma , Alpha)

Hyperparameter	BanglaBERT	BanglaGPT2	mBERT	IndicBERT
LR	2E-5	2E-5	2E-5	2E-5
BS	20	1	20	20
EP	20	20	20	20
WD	0.02	0.02	0.02	0.02
MP	True	True	True	True
TML	200	200	200	200
ES	True	True	True	True
ESP	3	3	3	3
FL	2,4	2,4	2,4	2,4

Table 6: Hyperparameter and Fine-Tuning Settings for Various Transformer Models in Our Experiment

C Loss functions

C.1 Self-adjusting Dice Loss

The Self-adjusting Dice Loss(Li et al., 2019) was introduced as an objective function for handling imbalanced datasets in NLP. It derives from the original dice coefficient, an F1-oriented metric for measuring set similarity. This loss function, based on a modified dice coefficient, was reported to yield superior F1 scores compared to models trained with cross-entropy loss.

$$DiceLoss = 1 - \frac{2(1-p_{n1})^{\alpha} \cdot (p_{n1}) \cdot y_{n1} + \gamma}{(1-p_{n1})^{\alpha}(p_{n1}) + y_{n1} + \gamma} \quad (1)$$

Here, for the $\mathrm{n_{th}}$ training instance, $\mathrm{p_{n1}}$ is the predicted probability of positive class and $\mathrm{y_{n1}}$ is the ground truth label. The loss function also has two hyperparameters, alpha and gamma, which we tuned for our models.

C.2 Focal Loss

In order to focus on hard, wrongly classified samples, Focal Loss applies a modulating term to the cross-entropy loss. Given the crossentropy loss formula:

$$CrossEntropyLoss(p_t) = -\alpha_t \cdot \log(p_t) \quad (2)$$

the focal loss formula is as follows:

$$FocalLoss(p_t) = -\alpha_t \cdot (1-p_t)^{\gamma} \cdot \log(p_t) \quad (3)$$

where α and γ are the focusing hyperparameter. The higher the hyperparameter, the more the focal loss function will focus on wrongly classified samples.

C.3 F1 micro loss

We transformed the F1-micro score metric into an F1-micro loss specific to our task. This loss function optimizes the F1-micro score and prioritizes overall performance across all classes, offering a more balanced evaluation of a model's capabilities in scenarios involving class imbalance.

D Error Analysis

Class	Precision	Recall	F1
Neutral	0.53	0.39	0.45
Negative	0.77	0.71	0.74
Positive	0.74	0.85	0.79

Table 7: Classification Report of Best Performing Model