PCIC at SMM4H 2024: Enhancing Reddit Post Classification on Social Anxiety Using Transformer Models and Advanced Loss Functions

Leon Hecht, Victor Martinez Pozos, Helena Gomez Adorno,
Gibran Fuentes Pineda, Gerardo Eugenio Sierra Martínez, Gemma Bel Enguix
leon.hecht@comunidad.unam.mx
National Autonomous University of Mexico

Abstract

We present our approach to solving the task of identifying the effect of outdoor activities on social anxiety based on reddit posts. We employed state-of-the-art transformer models enhanced with a combination of advanced loss functions. Data augmentation techniques were also used to address class imbalance within the training set. Our method achieved a macro-averaged F1 score of 0.655 in the test data, exceeding the mean F1 score of the shared task of 0.519. These findings suggest that integrating weighted loss functions improves the performance of transformer models in classifying unbalanced text data, while data augmentation can improve the model's ability to generalize.

1 Introduction

This paper addresses Task 3 of the 9th Social Media Mining for Health (SMM4H) (Xu et al., 2024) workshop at ACL 2024, which focuses on analyzing the impact of outdoor activities on social anxiety through the lens of Reddit posts. Despite advances in natural language processing, accurately classifying such nuanced data poses significant challenges due to linguistic variability and data imbalances.

Previous studies have shown that there exist various possibilities to address data imbalance in classification tasks. For example, in (Shaikh et al., 2021), additional samples for underrepresented classes were generated. In (Hasib et al., 2023), Random Under-Sampling and Synthetic Minority Oversampling Techniques were employed. Other authors proposed to tackle the class imbalance problem by introducing loss functions that focus on the underrepresented classes (Lin et al., 2017).

Our study builds on these works by incorporating a novel combination of weighted loss functions within a transformer model to address the challenge of imbalanced data.

2 Task Description

The aim of this task is to classify into four classes if an outdoor activity mentioned in a post had a positive, neutral, negative, or unrelated impact on the person's social anxiety symptoms. The outdoor activity mentioned in every text was given as a keyword in an extra column, so that the dataset consisted of the columns `'id'`, `'text'`, `'keyword'` and `'label'`. The training dataset includes 1800 posts, and the validation and test set includes 600 posts each.

The keyword is highly important in the classification task since a post can be highly negative, but still, in only one sentence, the user mentions the positive effect of an outdoor activity. In this case, even though the entire text itself was negative, it should be classified as positive. So, it was important to somehow link the classification task to the keyword.

Another challenge was to tackle the highly imbalanced classes. The training dataset containing 1800 samples has 1131 texts for the class `'unrelated'`, 395 for class `'neutral'`, 160 for class `'positive'` and only 114 for class `'negative'`.

3 Methodology

For the experiments run to solve task 3, different transformer models of the huggingface library were used. A small model (DistilBert), two medium size models (RoBERTa and XLNet-base), and a larger model (XLNet-large) were employed (Sanh et al., 2019; Liu et al., 2019; Yang et al., 2019). These models were modified in some experiments to use a combined loss function. This combined loss function consists of the `Focal-Loss`, designed to address class imbalance by increasing the importance of hard-to-classify examples, `Weighted-Cross-Entropy` Loss, which assigns different weights to classes based on

Proceedings of The 9th Social Media Mining for Health Applications (#SMM4H) Workshop and Shared Tasks, pages 63–66
August 15, 2024 ©2024 Association for Computational Linguistics

Run	Class distribution	LR	DistilBert	RoBERTa	XLNet-base	XLNet-large
0	(400, 97, 231, 71)*	5e-6	0.47	0.55	0.49	0.49
1	(400, 97, 231, 71)*	5e-6	0.54	0.55	0.54	0.55
2	(400, 97, 231, 71)*	5e-6	0.47	0.56	0.55	**0.60**
3	(796, 464, 530, 415)*	5e-6	0.49	0.52	0.51	0.57
4	(1000, 580, 1000, 415)*	5e-6	0.51	0.56	0.56	0.53
5	(2000, 928, 1590, 747)*	5e-6	0.52	0.53	0.54	**0.54**

Table 1: Results of the different experiments run to monitor the influence of each modification on the model performance. *Class distribution has the order (unrelated, negative, neutral, positive). Note: LR stands for Learning Rate.

their representation in the training data, and `Weighted-Smooth-Cross-Entropy` Loss. This variation adds smoothing to the class labels to improve generalization (Lin et al., 2017). To combine these three loss functions, the mean of the three values is computed after each batch.

The preprocessing of the text data included cleaning the text of any URLs, extra whitespaces and performing a conversion of the emojis used to text using the python library `'emoji'` (Teahoon and Wurster, 2024). Apart from that, only the sentence(s) in each text containing the keyword and their previous and next sentence were used. This adaptation was made to reduce the text's length to prevent a substantial part of the text from being truncated in the tokenization process. The previous and next sentences were included to provide more context for the keyword phrases. The keyword was appended to the input text by using the model-specific separator token.

For training and validation, the training dataset (1800 instances) was divided into 70% for training and 30% for validation with a seed of 42. The macro-averaged F1 score was evaluated on the validation dataset (600 instances) to test the model performance. As a termination criterion, the F1 score was used for early stopping with a patience of 6 epochs. For the learning rate, the value of 5e-6 was the best result of a hyperparameter optimization and was therefore chosen for the experiments. The batch size was set to either 16 or 32 depending on the max-length parameter of the tokenizer due to hardware restrictions.

For some experiments, data augmentation was performed to address the class imbalance. On one hand, an augmentation by paraphrasing was used. On the other hand, augmentation was performed by punctuation insertion, random deletion, random insertion, and random swapping (from now on referred to as ’traditional augmentation’).

For the paraphrasing task, a finetuned model from the huggingface repository was used, which is based on the T5-base model and finetuned on paraphrases generated by ChatGPT (Vorobev and Kuznetsov, 2023). For the punctuation insertion augmentation, the punctuation marks [’,’, ’.’, ’!’, ’?’, ’;’] were inserted at a random position in the text with a frequency of 10% in relation to the number of words in the text. In the random deletion augmentation, each word in the text is deleted with a probability of 20%. In the random-insertion augmentation, four random words in the text were chosen, and then, using the wordnet of the library `'nltk'`, synonyms for these words were searched (Bird et al., 2009). One of the synonyms found was randomly inserted in the text for each of the four words. For the random swapping task, two random words were chosen in the text and then swapped.

Table 2 shows the hardware and software environment with which all experiments were run.

4 Results and Discussion

First, a baseline classification (Run 0) was run with the four different models. In this experiment, the original loss function of the model was used, and no data augmentation was performed.

For the following experiments, we aimed to evaluate the effect of the combined loss function (Run 1), the max-length parameter of the tokenizer (Run 2), data augmentation using paraphrasing (Run 3), and data augmentation using the aforementioned traditional augmentation (Run 4). Finally, an experiment was run where all these methods were combined, using the combined loss function, increased value for the max-length parameter, and both data augmentation methods (Run 5). The results of these experiments are shown in Table 1.

Operating System	Linux
GPU	Nvidia RTX A5000 24GB (1)
CUDA Version	12.1
Deep Learning Framework	PyTorch 2.2.0
Transformers Library Version	4.39.0
Python Version	3.10.12

Table 2: Specifications of the hardware and software environment used in the experiments.

Analyzing the experiment results, we can observe that, with exception of the RoBERTa model, all modifications outperformed the baseline or at least had the same performance. Using the combined loss function, improved the F1 score in the experiments for all models except of RoBERTa, with a mean absolute improvement of 0.06. This combined loss function is used throughout the following experiments.

Changing the max-length parameter of the tokenizer from 128 to 256, yielded the best result in all experiments for XLNet-large, lifting the F1 score up from 0.55 to 0.60. The motivation of this experiment surged from an analysis that we conducted, showing that 334 texts (18.5%) of the tokenized text lengths in the training dataset surpass the previously set length of 128 for the XLNet tokenizer (see figure 1). For comparability, the same max-length parameter was set for the experiments with DistilBert and RoBERTa, even though these models work with a different tokenizer and tokenized text length might differ.

Augmentation by paraphrasing only showed a slight improvement for one model, XLNet-large. At the same time, traditional augmentation showed a mixed effect on the F1 score across the different models. In the experiment with XLNet-large, traditional augmentation seems to have a negative impact on the F1 score, decreasing it from 0.55 to 0.53.

In the following experiment, all of the previous modifications were combined. Hence, the combined loss function was used, the max-length parameter of the tokenizer was set to 256, and the data were augmented using both augmentation methods. Despite the slightly decreased performance on the validation data in this experiment, it is possible that due to the augmentation, the model's ability to generalize improves. Therefore, this set-up with the model XLNet-large was used to obtain the predictions on the test set (0.655 F1 score) and was sent to the task organizers.

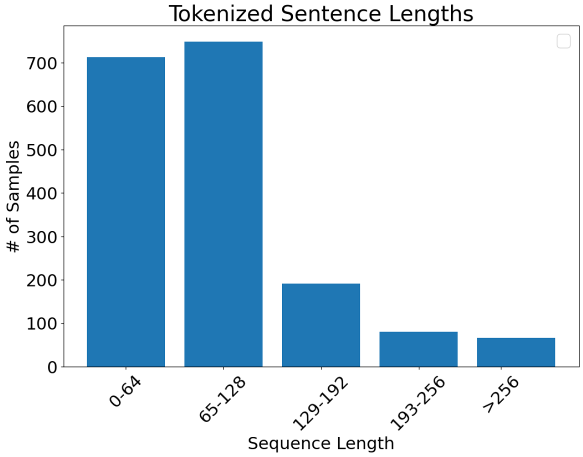

Figure 1: Sequence length analysis of tokenized texts (XLNet tokenizer)

5 Conclusion

This paper presents our approach to solving task 3 of the SMM4H 2024 workshop, which consists of a unique pre-processing to avoid truncating important information and providing only the key phrases of the text to the model. Furthermore, the XLNet model was adapted to use a combined loss function, and data augmentation was performed to address the class imbalance and improve generalizability. The keyword related to the outdoor activity was appended to the input text using the separator token. This setup has resulted in a macro-averaged F1 score of 0.655 on the test data, outperforming the mean of 0.519.

6 Acknowledgments

This work was carried out as part of PAPIIT project IT100822, IN104424 and the project CF2023-G64, which is funded by '*Consejo Nacional de Humanidades, Ciencias y Tecnologías* (CONAHCYT)'. We also want to thank Ricardo Villareal and Rita Rodríguez for supporting us with the computing resources.

References

S Bird, E Klein, and E Loper. 2009. *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

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

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Sarang Shaikh, Sher Muhammad Daudpota, Ali Shariq Imran, and Zenun Kastrati. 2021. Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models. *Applied Sciences*, 11(2).

K. Teahoon and K. Wurster. 2024. emoji: A python library to perform operations on emojis in text data. `https://carpedm20.github.io/emoji/docs/index.html`. Accessed: 2024-05-13.

V Vorobev and M Kuznetsov. 2023. A paraphrasing model based on chatgpt paraphrases.

Dongfang Xu, Guillermo Lopez Garcia, Lisa Raithel, Rolland Roller, Philippe Thomas, Eiji Aramaki, Shuntaro Yada, Pierre Zweigenbaum, Sai Tharuni Samineni, Karen O'Connor, Yao Ge, Sudeshna Das, Abeed Sarker, Ari Klein, Lucia Schmidt, Vishakha Sharma, Raul Rodriguez-Esteban, Juan Banda, Ivan Flores Amaro, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2024. Overview of the 9th social media mining for health applications (#SMM4H) shared tasks at ACL 2024. In *Proceedings of The 9th Social Media Mining for Health Research and Applications Workshop and Shared Tasks*, Bangkok, Thailand. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.