From Pain to Perspective:
Self-reported Chronic Pain and Patient-Providers Sentiment Classification

Beatrice Brown-Mulry, Ipsa Yadav, Peibo Zhang

Self-Reporting Chronic Pain Classification

In this study, we undertook the task of fine-tuning a RoBERTa classifier to identify self-reported chronic pain status in social media posts. Our methodology involved aggregating three distinct datasets (training, testing, and development) into a unified corpus, comprising 3,299 posts. Of these, 2,765 were labeled as non-self-reporting chronic pain (denoted as '0'), and 534 were self-reported chronic pain posts (labeled as '1'). The dataset was split into training, validation, and testing subsets in a 60%, 20%, and 20% ratio, respectively, while ensuring a balanced distribution of labels between subsets.

Tokenization and encoding were performed using the RoBERTa base tokenizer from the Hugging Face Transformers library. This tokenizer truncates or pads inputs to 256 tokens—the model's maximum length—and converts all text to lowercase. Additionally, we employed data loaders for efficient batch processing during training and evaluation, a strategy advantageous for handling larger datasets in future studies. The posts used to train the classifier were preprocessed in a preceding study so no additional preprocessing was applied. We plan to integrate a similar preprocessing pipeline in the forthcoming stages of the project in order to refine the larger unprocessed chronic pain dataset.

Our model architecture adhered to the standard configurations provided in the Hugging Face Transformers library. This setup included a pre-classifier linear layer with ReLU activation and a final classifier layer. The output undergoes a Sigmoid activation function, aligning with the requirements of binary classification tasks. The model's performance was evaluated on a test dataset, and metrics such as Accuracy, AUROC, Precision, Recall, and F1 Score were reported. Due to the class imbalance in the data, F1 Score was prioritized in the evaluation and hyperparameter tuning process. We employed Binary Cross-Entropy Loss as our loss function and utilized the Adam optimizer with an initial learning rate of 0.00005. The model was trained for 30 epochs and achieved a preliminary F1 Score of 77.59% before hyperparameter tuning (**Table 1**).

Accuracy	AUC	F1	Precision	Recall
91.98%	93.68%	77.59%	72.58%	83.33%

Table 1. Pre-Tuning Model Performance

To further enhance model performance, we tuned two critical parameters: dropout proportion and learning rate. Recognizing the interplay between these parameters, we aimed for a balance to ensure both generalizability and stable learning progress. Our approach employed Bayesian optimization for hyperparameter tuning, offering a more efficient method compared to random or grid searches. This technique iteratively tests hyperparameters by balancing exploration and exploitation, thereby narrowing down the search space more effectively. The outcomes of this process, including the impact of varying hyperparameters on model performance, are visually represented in a heatmap (**Figure 1**). Post-tuning, the best-performing hyperparameter combination yielded an improved F1 Score of 79.26%, which is comparable with the benchmark F1 score of 83% from the previous study. The model performance metrics after hyperparameter tuning are shown in **Table 2**.

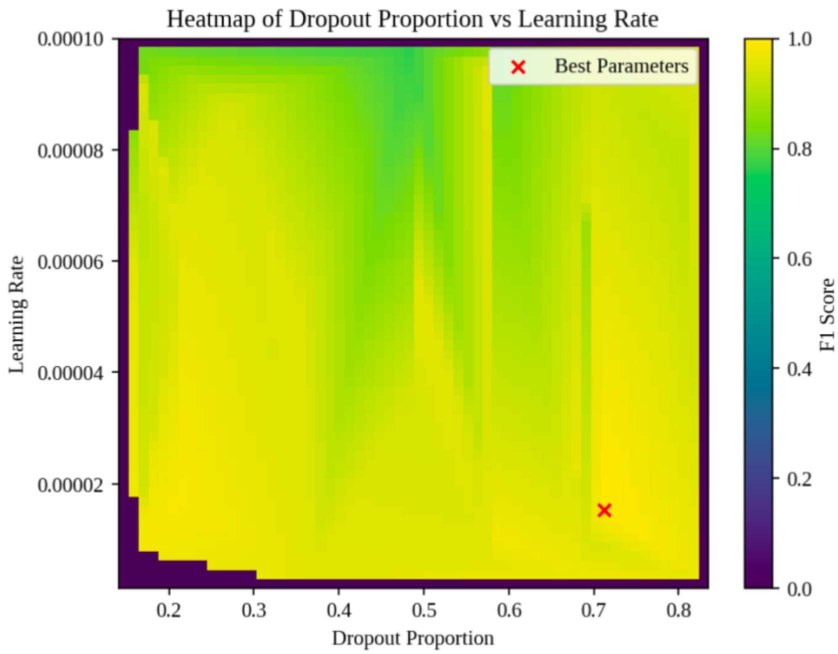

Figure 1. Hyperparameter Heatmap

Accuracy	AUC	F1	Precision	Recall
93.19%	95.50%	79.26%	78.90%	79.63%

Table 2. Post-Tuning Model Performance