GEORGE BROWN COLLEGE
SCHOOL OF COMPUTER TECHNOLOGY
APPLIED A.I. SOLUTIONS

APPLIED MATHEMATICAL CONCEPTS FOR DEEP LEARNING

Professor: Dr. Mahdieh Khalilinezhad

Seyed Amir Kiarash Abbasi

Mohammad Ardalanasl

Feb 05, 2024

TABLE OF CONTENTS

1.0 INTRODUCTION

This project aimed to provide hands-on experience in developing a Natural Language Processing (NLP) model using Transformers. We focused on leveraging various versions of BERT for text classification across diverse databases. Our objective was to broaden the applicability of pre-trained models to classify text data from multiple sources. To enhance model performance, we incorporated Particle Swarm Optimization (PSO), a widely adopted optimization technique, to search for optimal hyperparameters. Due to time and hardware constraints, we employed multiple datasets to evaluate the effectiveness of hyperparameter optimization strategies.

1.1 Data

For training and evaluating the pre-trained model, we utilized two distinct datasets sourced from different websites. These datasets consist of text samples paired with corresponding labels. The datasets used are as follows:

- **Dataset-1**: Named "dair-ai/emotion," this dataset encompasses approximately 20,000 records for emotion detection, categorized into 6 classes [1].
- **Dataset-2**: Named "CNN News Articles from 2011 to 2022," this dataset comprises approximately 4,000 records categorized into 6 classes [2].

1.2 Preprocessing Data

Prior to model training, we performed preprocessing on the datasets. This involved removing irrelevant columns, cleaning the data, and converting it into NumPy arrays (X and y) for compatibility with the model. We then partitioned each dataset into three subsets: the Training set, the Validation set, and the Test set, using an 80%, 10%, and 10% split of the total data, respectively.

Furthermore, we utilized the "Auto Tokenizer" provided by the BERT model to process the text data. This step involved converting each record in the dataset into input layers suitable for the model's input format.

2.0 METHODOLOGY

In this project, we propose two main approaches for fine-tuning the pre-trained model:

2.1 The Pre-trained Model + A Classification Layer

For this approach, we employed a BERT model with an untrained 6-class SoftMax layer. The objective was to train this new layer using various datasets. The format of the pre-trained model, utilized for text classification purposes, is illustrated in Figure 2.

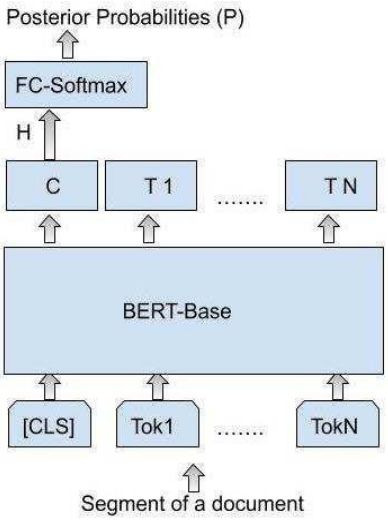

Figure 1. The format of the pre-trained model + a classification layer

2.2 The fine-tuning of the model using PSO

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique inspired by the social behavior of bird flocking or fish schooling. The algorithm maintains a population of candidate solutions, referred to as particles, which move through the search space to find the optimal solution. Each particle's position in the search space represents a potential solution, and its movement is guided by its own experience and the collective behavior of the swarm. One of the key advantages of PSO is its simplicity and ease of implementation compared to other optimization algorithms. PSO does not require gradient information, making it well-suited for optimization problems where derivatives are not readily available or are expensive to compute.

In this project, we employed PSO for fine-tuning the model by searching the possible domain of hyperparameters and tracking the improvement in the results using a predefined cost function. The algorithm of the PSO model for searching the domain is depicted in Figure 1. In each iteration of the PSO model, we trained and validated the model, and the cost function was calculated as (1/F1)^2, where F1 is the F1 score of the validation set.

After obtaining the optimal values for hyperparameters, we used these values for training and testing the pre-trained classification model.

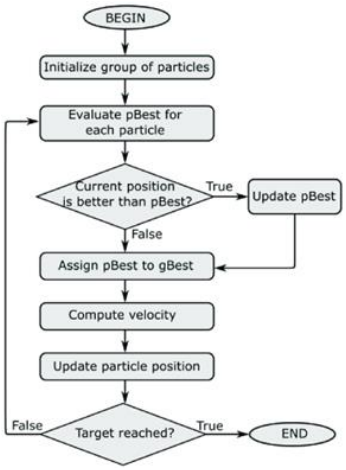

Figure 2. Flowchart of PSO method

3.0 RESULTS

In this section, we present the outcomes of our validation and testing endeavors across various datasets using pre-trained models.

3.1 The results of the PSO-based hyperparameter tuning model

In this project, we utilized the PSO method to determine the optimal hyperparameters for our model. As mentioned earlier, this method operates through a multi-agent approach that requires several iterations to converge on optimal results.

Upon executing the PSO method, we obtained the most favorable results for each strategy. Subsequent adjustments to the resulting hyperparameters led us to select the following set for fine-tuning our model:

Table 1. Selected Hyperparameters for Model Tuning

Learning Rate	Batch Size	Weight Decay	No. Epochs
8.10E-03	220	2.50E-04	100

Following the selection of these parameters, we proceeded to fine-tune the pre-trained model, which incorporates a 6-class SoftMax classifier.

3.2 The results of the model with optimal parameters

We employed the optimal parameters to train the BERT and Electra models alongside a 6-class SoftMax layer, resulting in the following accuracy, F1 score, and loss values across different databases:

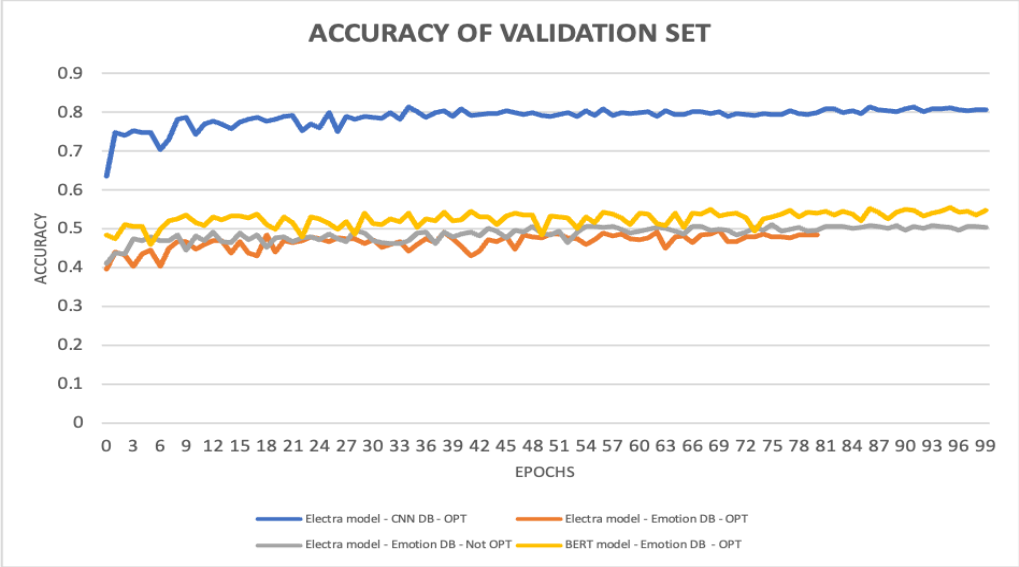

Figure 3. Accuracy trends in the validation set

This figure illustrates the accuracy of the model on the CNN dataset and Emotion Detection dataset, demonstrating the improvement in accuracy achieved by using the optimal parameters. The BERT model with the optimal hyperparameters yielded the best results.

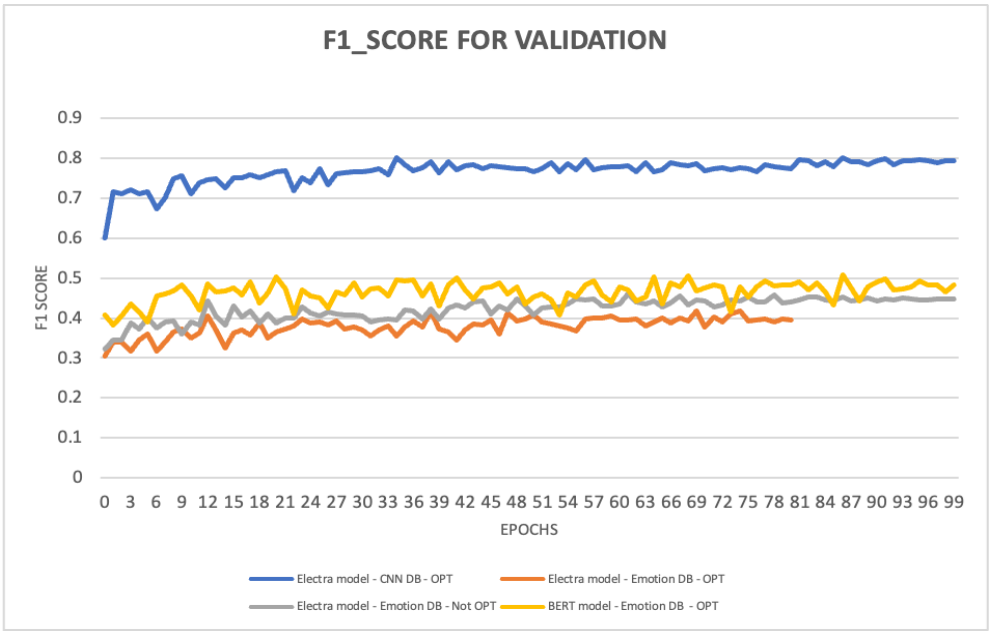

Figure 4. F1 score trends in the validation set

In this figure, we observe the F1 score for the BERT and Electra models on the Emotion Detection and CNN datasets. The results show improvement in all cases, with the BERT model with the optimal hyperparameters achieving the best performance.

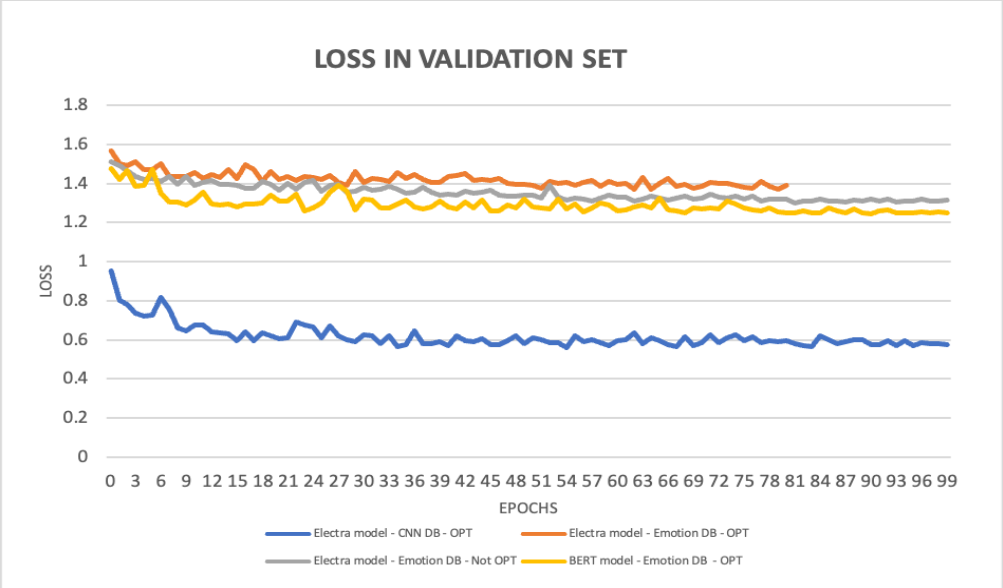

Figure 5. Loss trends in the validation set

This graph depicts the loss values of the previously defined models, demonstrating a decrease in loss across all cases. The BERT model with the optimal hyperparameters achieved the best results. The decline in the loss value in the CNN dataset indicates a rapid convergence to the optimal value.

Following model training, we evaluated the performance of the model on the test sets to assess its generalization capabilities:

Table 2. Test set results using parameters 1

	Accuracy	F1 Score	Loss Value
Electra model - Emotion DB - Not OPT	0.4985	0.4458	1.2926
BERT model - Emotion DB - OPT	0.5410	0.4779	1.2272
Electra model - CNN DB - OPT	0.7647	0.7426	0.6622

In this table, we observe that the results in the CNN dataset and Emotion Detection dataset are quite close to their corresponding validation results. This indicates that the model could effectively manage the overflow condition. Additionally, while the range of values in the CNN dataset is reasonable, the values in the Emotion Detection dataset exhibit less satisfactory outcomes.

Employing the optimal parameters in the Emotion Detection dataset enhances the range of parameters, but the final results do not fall within the acceptable ranges. This highlights a potential issue with the dataset, suggesting that a larger volume of data may be required to achieve higher accuracy in this dataset.

4.0 CONCLUSIONS

This project aimed to develop a Natural Language Processing (NLP) model using Transformers, focusing on leveraging various versions of BERT for text classification across diverse datasets. By incorporating Particle Swarm Optimization (PSO) for hyperparameter optimization and employing multiple datasets, we sought to enhance model performance within the constraints of time and hardware resources.

Our methodology involved two main approaches for fine-tuning the pre-trained model: utilizing a BERT model with an untrained 6-class SoftMax layer and employing PSO for hyperparameter tuning. The PSO method, a population-based stochastic optimization technique, successfully identified optimal hyperparameters, leading to improvements in model performance.

The results of our model training and testing demonstrated promising outcomes. We observed significant improvements in accuracy, F1 score, and loss values across different datasets when using the optimal parameters. However, challenges were encountered in achieving satisfactory results with the Emotion Detection dataset, indicating the potential need for a larger volume of data to improve accuracy.

In conclusion, this project provides valuable insights into the development and optimization of NLP models using state-of-the-art techniques. Further research and experimentation, particularly with larger datasets, could lead to enhanced performance and broader applications of these models in real-world scenarios.

5.0 REFERENCES

[1] Dair-ai, Hugging Face, https://huggingface.co/datasets/dair-ai/emotion

[2] CNN News Articles from 2011 to 2022, Kaggle, https://www.kaggle.com/datasets/hadasu92/cnn-articles-after-basic-cleaning