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

DEEP LEARNING-I

Professor: Dr. Mo Fadaee

Group 12

Seyed Amir Kiarash Abbasi
Mohammad Ardalanasl

Feb 02, 2024

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

In this project, our objective was to gain practical experience in developing a Natural Language Processing (NLP) model using Transformers. To achieve this, we utilized a fine-tuned version of BERT, a pre-trained model specifically tailored for Amazon reviews. Our goal was to extend the applicability of this model to perform sentiment analysis on review data from various websites. Additionally, we employed Particle Swarm Optimization (PSO), a widely used optimization technique, to search for optimal hyperparameters for our model. Given the constraints of time and hardware resources, we adopted multiple strategies to address the challenges of hyperparameter optimization. Ultimately, we present two sets of hyperparameters and conducted training and validation using these settings. Subsequently, we evaluated the performance of our models on diverse review datasets from different websites to assess their generalization capabilities.

1.1 Problem statement

In this report, we have utilized the 'LiYuan/amazon-review-sentiment-analysis' model [1], which is a fine-tuned model trained specifically on Amazon review data. However, it is important to note that this model has not been trained or tested on reviews from other websites. Therefore, the objectives of our project are twofold: (1) to evaluate the performance of the pre-trained model when applied to review data from different websites, and (2) to enhance the model's performance by optimizing its hyperparameters. We aim to achieve this by training and testing the model with review data from diverse sources. Additionally, we provide two sets of optimal hyperparameters that have been trained and compared with the performance of the original model.

What is 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. The position of each particle is updated iteratively based on its current position, its best-known position (personal best), and the best-known position found by any particle in the swarm (global best). This global exploration and exploitation process allows PSO to efficiently search for optimal solutions in complex search spaces.

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. Additionally, PSO is known for its ability to effectively balance exploration and exploitation of the search space, which can lead to the discovery of high-quality solutions. However, PSO's performance may be sensitive to its parameter settings, such as the number of particles, the inertia weight, and the cognitive and social parameters. Careful tuning of these parameters is essential to ensure the convergence and effectiveness of the algorithm. Overall, PSO has been widely applied in various fields, including engineering, finance, and machine learning, due to its effectiveness in solving optimization problems. The flowchart of the PSO method is shown in the following figure.

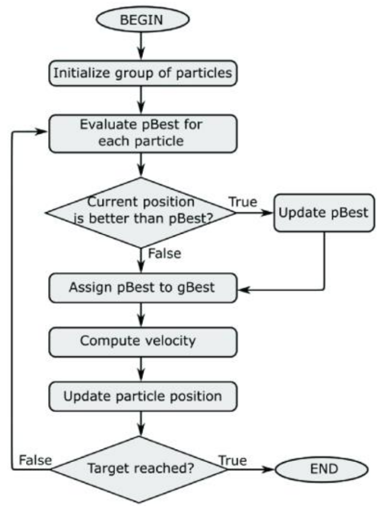

Figure 1. Flowchart of PSO method

How to use PSO?

In this project, we used the "GlobalBestPSO" method for implementing the PSO method. The "GlobalBestPSO" method is a variant of the Particle Swarm Optimization (PSO) algorithm that focuses on a global exploration of the search space to find the optimal solution. In this method, each particle in the swarm adjusts its position based on the best-known position of any particle in the entire swarm (global best). This global best position is updated as particles explore the search space, allowing the swarm to collectively converge towards promising regions of the solution space. The "GlobalBestPSO" method is particularly effective for problems where a global optimum is sought, as it encourages the entire swarm to converge towards the same solution. This method is widely used in optimization problems where the objective is to find a single optimal solution rather than multiple solutions or a diverse set of solutions. for more information about the details of implementing the PSO in Python, the readers can see the documentation of the “Pyswarms” library [6].

1.2 Data

In this project, we utilized several review datasets to train and evaluate the BERT model. These datasets contain review text and corresponding scores ranging from 1 to 5, sourced from different websites. The datasets used are as follows:

- Dataset-1: Named "LoganKells/amazon product reviews video games", this dataset comprises approximately 50,000 records of Amazon reviews related to video games products [2].

- Dataset-2: Named “Amazon Musical Instruments Reviews”, this dataset comprises approximately 10,000 records [3]
- Dataset-3: Named “Google Maps Restaurant Reviews”, this dataset comprises approximately 1100 records [4]
- Dataset-4: Named “Trip Advisor Hotel Reviews”, this dataset comprises approximately 20500 records [5]

Preprocessing Data

Prior to training the model, we conducted preprocessing on all datasets. This involved removing irrelevant columns, cleaning the data, and converting it into NumPy arrays (X and y) for use in the model. Subsequently, we partitioned each dataset into three main subsets: the Training set, the Validation set, and the Test set. These subsets were divided in a ratio of 80%, 10%, and 10% of the total data, respectively. Additionally, we processed the text data using the "AutoTokenizer" provided by the BERT model. This step involved converting each record of the dataset into input layers suitable for the model's input format.

1.3 Tasks

For this project, we leveraged a fine-tuned benchmark model tailored for sentiment analysis on Amazon reviews. Our tasks involved using this model in three main scenarios:

(1) A benchmark model for evaluating sentiment analysis on various databases.

(2) Training a new layer on reviews from different websites and comparing its performance with the benchmark model.

(3) Analyzing the impact of increasing the number of layers on the model's performance.

Additionally, we explored the use of this model as a cost function for a Particle Swarm Optimization (PSO) method to search for optimal hyperparameters, including Learning Rate, Batch Size, and Weight Decay Ratio.

2.0 METHODOLOGY

In this project, we present three main models for training and evaluating results based on the LiYuan model:

2.1 The fine-tuned BERT benchmark model (LiYuan)

We utilized a BERT model with a pre-trained 5-class SoftMax layer (LiYuan) to establish a benchmark performance on different databases.

The LiYuan model, named "distilbert-base-uncased-finetuned-mnli-amazon-query-shopping," is a fine-tuned version of the nlptown/bert-base-multilingual-uncased-sentiment model specifically tailored for sentiment analysis on Amazon US Customer Reviews. It is based on the Bert-base-multilingual-uncased model, which has been fine-tuned for sentiment analysis on product reviews in six languages: English, Dutch, German, French, Spanish, and Italian. The LiYuan model predicts the sentiment of reviews as a number of stars (ranging from 1 to 5) and is intended for use in sentiment analysis tasks related to product reviews in any of the aforementioned languages.

The developers of this model claim that in the fine-tuning process, the head of the model was replaced with customer reviews, and it was trained on a dataset comprising 17,280 rows for training and 4,320 rows for validation. The model's performance was then evaluated on a held-out test set containing 2,400 rows. It achieved a loss of 0.5202942490577698 and an accuracy of 0.8 on the evaluation set.

The developer team also indicates that while the LiYuan model is effective for sentiment analysis tasks related to Amazon product reviews in multiple languages, it is important to note that its performance may be limited when applied to domains outside of Amazon products. This limitation stems from its training specifically on Amazon review data, which may not generalize well to other domains.

2.2 The BERT Model With A New Layer Training

For this model, we used a BERT model (LiYuan) with an un-trained 5-class SoftMax layer. The objective was to train this new layer using reviews from various websites and compare its performance with the benchmark model.

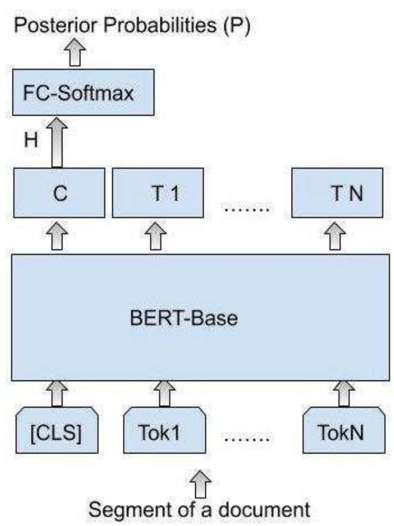

Figure 2. The format of the BERT model with a new layer

2.3 The BERT Model with Two New Layers Training

In this model, we employed a BERT model (LiYuan) with two un-trained layers. The aim was to train these new layers using reviews from diverse websites, compare the results with the benchmark model, and analyze the impact of increased layer depth on the final results. The first layer comprised 256 nodes with a ReLU activation function and a dropout ratio of 0.1, while the second layer consisted of 5 nodes with a SoftMax activation function.

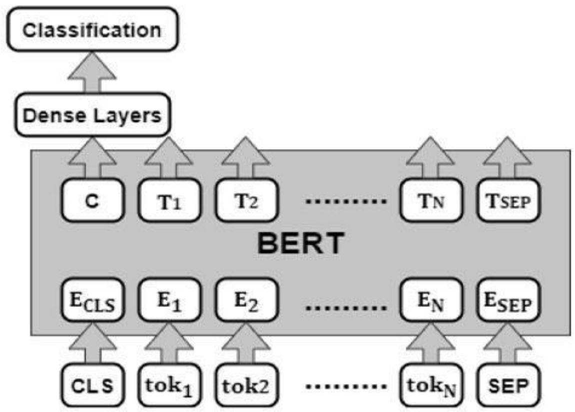

Figure 3. The format of the BERT model with two new layers

2.4 PSO-Based Hyperparameter Tuning:

This model utilized a BERT model (LiYuan) with an un-trained 5-class softmax layer, acting as a cost function for a PSO method. The goal was to search for optimal values of hyperparameters such as Learning Rate, Batch Size, and Weight Decay Ratio using the PSO algorithm.

The optimal hyperparameters obtained from this method are applied to the other aforementioned BERT-base models to enhance their performance. The flowchart below illustrates the model, incorporating the GlobalBestPSO optimizer and the cost function.

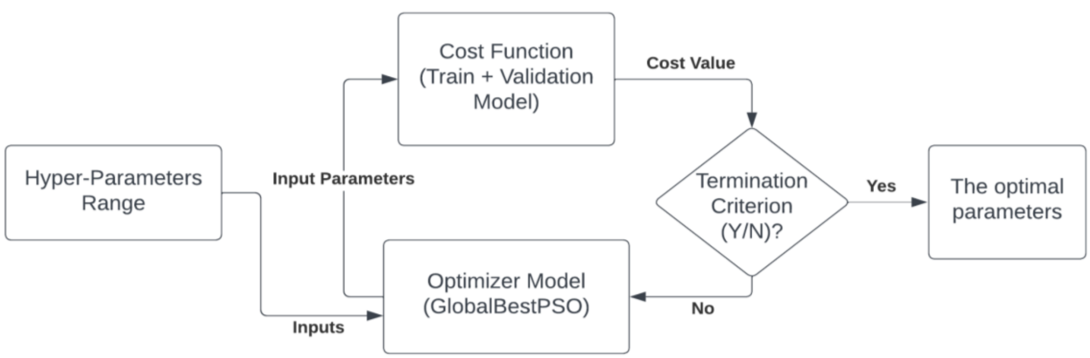

Figure 4. Flowchart of Optimization Model

Cost function

The choice of cost function is a critical factor influencing the effectiveness of optimization methods. We experimented with various parameter combinations for the cost function, ultimately selecting the function (1/F1)^2. In this equation, F1 represents the F1 score of the validation set. During the optimization process, our optimizer aims to minimize the value of the cost function, which consequently leads to an improvement in the F1 score.

3.0 RESULTS

In this section, we present the results of our validation and testing efforts across various datasets using predefined models.

3.1 The test results of the BERT benchmark model (LiYuan)

We obtained the BERT benchmark model from the "Hugging Face" website and employed its classifier to assess our diverse datasets. We allocated 10% of each dataset as a test set for evaluating the model, and the outcomes are summarized in Table 1.

Table 1. Model -1- Test Results

	Test Results			
	No. Records	Accuracy	F1 Score	Loss
Amazon Instruments	10261	0.743665	0.738918	0.643378
Amazon Video Games	49988	0.684937	0.672192	0.794072
Google Map	1100	0.445455	0.384341	1.269553
TripAdvisor	20491	0.538537	0.462444	1.064925

Upon reviewing the table, it becomes evident that despite the author's claims regarding the benchmark model, its accuracy in analyzing Amazon reviews falls below the 80% mark. Moreover, the accuracy and F1 scores for Google Maps and TripAdvisor datasets notably lag behind those of Amazon. These findings suggest that the benchmark model, having been trained solely on Amazon review data, struggles to effectively classify reviews from other platforms.

3.2 The results of the PSO-based hyperparameter tuning model

In this project, we employed the PSO method to identify the optimal hyperparameters for our model. As previously mentioned, this method operates through a multi-agent approach that necessitates numerous iterations to converge on optimal results. Due to the scale of both the model and the datasets, as well as the constraints at the time of the project, we opted to utilize a subset of records from the Amazon product review dataset for video games [2] in our application of the PSO method. While this approach introduces certain errors and uncertainties into the model, we accepted these limitations in pursuit of the project's educational objectives.

Given the multitude of parameters to consider, we pursued two distinct strategies:

Table 2. Strategies for PSO Parameter Optimization

Strategies	No. records	No. particles	No. iters	No. epochs
1	1000	5	10	3
2	5000	5	10	1

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 sets for fine-tuning the models:

Table 3. Selected Hyperparameters for Model Tuning

Strategies	Learning rate	Batch size	Weight decay	No. epochs
1	2.50E-04	140	6E-05	>10
2	6.50E-04	128	5.00E-05	>10

Subsequently, we employed each strategy to fine-tune the BERT model, which incorporates a 5-class SoftMax classifier.

The results of the model with Parameters 1 and 2

We utilized the parameters from strategies 1 and 2 to train the BERT model alongside a 5-class SoftMax layer, resulting in the following accuracy, F1 score, and loss values across different databases:

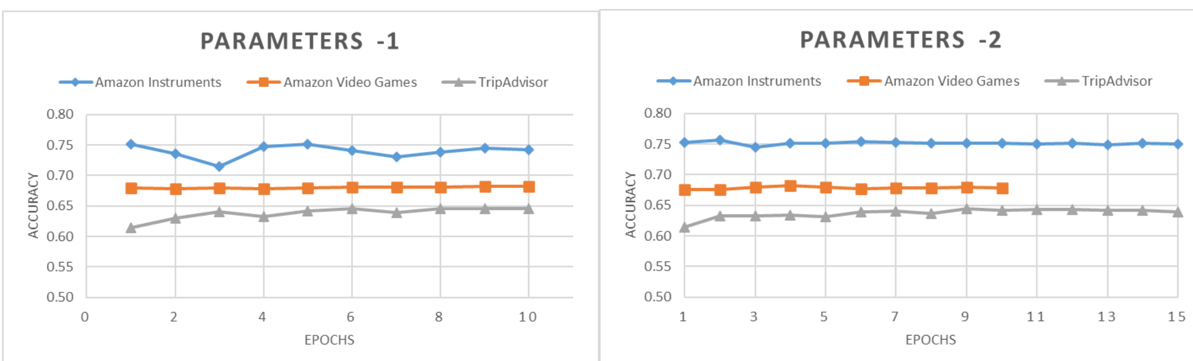

Figure 5. Accuracy trends in the validation set using parameters 1 and 2

In strategy 1, the accuracy of the model demonstrates a steady increase across epochs. Notably, while the accuracy for TripAdvisor remains low, it reaches a more reasonable value of approximately 0.75 for Amazon (musical instrument products).

In strategy 2, the accuracy exhibits a slight upward trend in the validation set, with varying behavior across different datasets. The model performs at around 75% accuracy for Amazon (instruments) but drops to less than 65% for TripAdvisor reviews.

Comparing the accuracy of parameters 1 and 2 reveals no significant change in overall model performance. Instead, the impact of these parameters appears to vary depending on the dataset, highlighting the model's greater sensitivity to epoch count and data size than to these specific parameters.

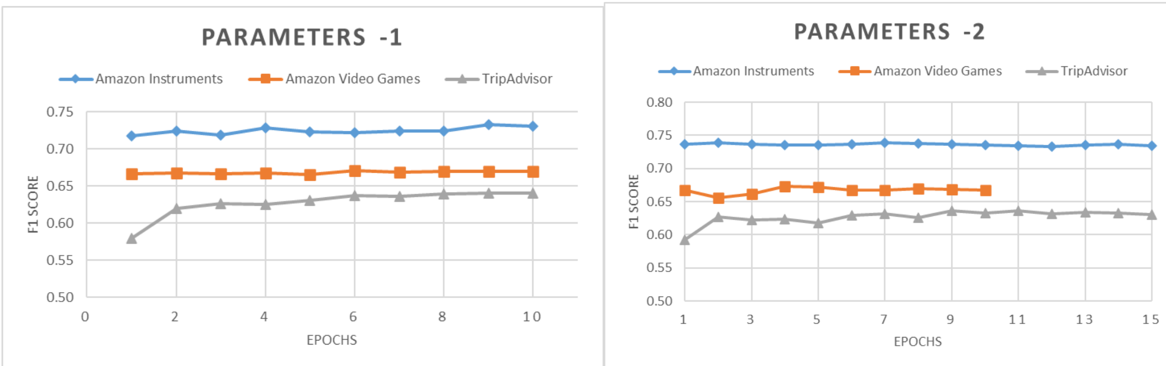

Figure 6. F1 score trends in the validation set using parameters 1 and 2

The F1 scores exhibit a similar pattern, with lower performance observed in non-Amazon databases compared to Amazon review databases.

In strategy 1, while the F1 values show a slight increase, the magnitude of change varies across databases. While there is an acceptable increase in the TripAdvisor database, the improvement in the Amazon (video games) database is marginal.

In strategy 2, the F1 scores follow a similar pattern to those in strategy 1. Only one dataset achieves an F1 score above 70%, with minimal variation observed as the epoch count increases.

Comparing the F1 scores of parameters 1 and 2 reveals that each model's performance varies across datasets. While parameter 2 yields better results for Amazon (video games), parameter 1 performs better for TripAdvisor. Thus, it is challenging to determine which parameter set is more efficient.

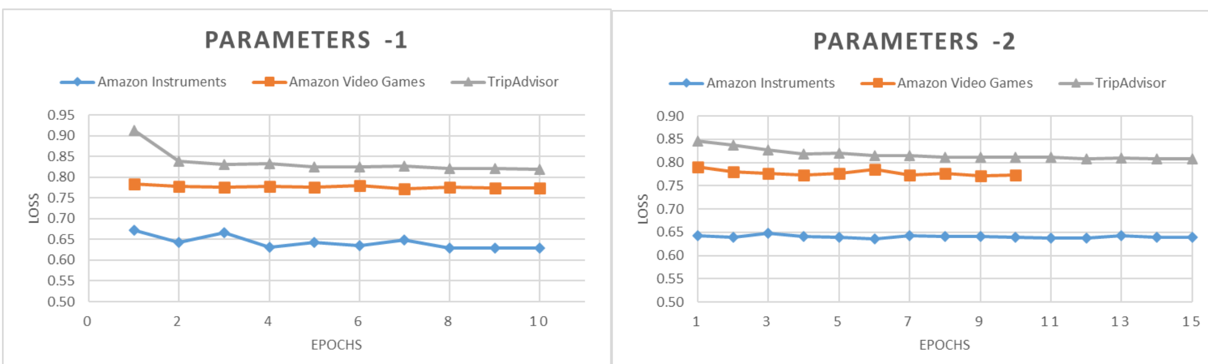

Figure 7. Loss trends in the validation set using parameters 1 and 2

In both cases, the loss values exhibit a gradual decrease with increasing epochs, suggesting that further epochs may lead to lower losses. Comparing the two graphs, we observe better performance for Amazon (instruments) with parameter 1, while parameter 2 yields better performance for TripAdvisor. Following model training, we evaluated its performance on a test set to assess its generalization capabilities:

Table 4. Test set results using parameters 1

	Test Results			
	No. Records	Accuracy	F1 Score	Loss
Amazon Instruments	10261	0.759259	0.746208	0.603521
Amazon Video Games	49988	0.693939	0.682645	0.758794
TripAdvisor	20491	0.624878	0.614431	0.842508

Table 5. Test set results using parameters 2

	Test Results			
	No. Records	Accuracy	F1 Score	Loss
Amazon Instruments	10261	0.757310	0.743704	0.610735
Amazon Video Games	49988	0.692939	0.682316	0.750689
TripAdvisor	20491	0.620488	0.606876	0.862526

The test results, including accuracy, F1 score, and loss value, closely align with the validation results, indicating proper model training. However, increasing the number of epochs could potentially lead to even better results. Based on the F1 score criterion, parameter 2 appears slightly superior to parameter 1 in the test results.

Comparing the Benchmark and New Model

When comparing the test results of the benchmark model with the fine-tuned model featuring one SoftMax layer, notable improvements are observed. In the case of Amazon data, the accuracy has increased by approximately 1.2% to 1.8%, the F1 score by about 0.6% to 1.5%, and the loss has decreased by approximately 2.5% to 5.1%. For non-Amazon data, the improvements are more significant, with an accuracy increase of about 15%, an F1 score increase of around 31%, and a loss reduction of about 19%. These results demonstrate the success of the fine-tuning process. Despite the benchmark model being trained with a large number of epochs, the new model, with optimized hyperparameters, outperformed it. Furthermore, this approach showcases its effectiveness in tuning non-Amazon data, suggesting its potential applicability for fine-tuning models for review data from various websites.

Table 6. Comparing the Benchmark and New Model

	Test Results		
	Accuracy	F1 Score	Loss
Amazon Instruments	1.8%	0.6%	-5.1%
Amazon Video Games	1.2%	1.5%	-5.5%
TripAdvisor	15.2%	31.2%	-19.0%

4.0 CONCLUSIONS

In this project, we aimed to expand the applicability of a pre-trained BERT model tailored for Amazon reviews to analyze sentiments across various review datasets from different websites. Additionally, we employed the Particle Swarm Optimization (PSO) method to search for optimal hyperparameters, enhancing the model's performance. Our evaluation revealed that while the benchmark BERT model struggled to classify reviews from websites other than Amazon, the PSO-based hyperparameter tuning improved its performance across diverse datasets. The comparison of different strategies for PSO parameter optimization highlighted the nuanced impact of hyperparameters on model performance, emphasizing the need for careful consideration and tuning. Furthermore, the analysis of model performance using different parameter settings underscored the model's sensitivity to dataset characteristics, with no clear superiority between the tested parameter sets.

Overall, our project demonstrates the potential of leveraging pre-trained models like BERT in conjunction with optimization techniques such as PSO to enhance their performance in real-world applications. However, it also underscores the importance of understanding the limitations and nuances of such models when applied beyond their original scope. Further research could focus on refining the PSO-based hyperparameter tuning process and exploring additional strategies to improve the model's generalization capabilities across diverse datasets and domains.

5.0 REFERENCES

[1] LiYuan, Hugging Face, https://huggingface.co/LiYuan/amazon-review-sentiment-analysis/tree/main

[2] LoganKells, Hugging Face, https://huggingface.co/datasets/LoganKells/amazon_product_reviews_video_games

[3] Eswar Chand, Kaggle, https://www.kaggle.com/datasets/eswarchandt/amazon-music-reviews?resource=download

[4] Deniz Bilgin, Kaggle, https://www.kaggle.com/datasets/denizbilginn/google-maps-restaurant-reviews

[5] Larxel, Kaggle, https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews

[6] Pyswarms, https://pyswarms.readthedocs.io/en/latest/api/pyswarms.single.html