Fine-Tuning BERT LLM for Emotion Classification

Introduction

Large Language models have gained so much interest in the Machine Learning field over the past years. Early models were so simple in nature to predict the words from the previous words. These models have very little knowledge in understanding for longer contexts. With breaking discovery of Transformer architecture and its attention mechanisms allowed to develop more models to focus on different major parts of machine learning areas by overcoming the limitations of the previous models.

Natural Language Processing (NLP) a field where we can use to classify different problems of applications we can think of. Emotional classification is such a critical task in this field, that involves categorizing the context of test or sentences into predefined emotional classes of categories. This report, we will focus on the fine-tuning a BERT-style language model to classify the sentences we have into six different emotions we define: Love, Sadness, Fear, Joy, Anger, and Surprise. BERT (Bidirectional Encoder Representations from Transformers) has made so many advances in the NLP tasks with contextual embeddings and bidirectional training, making it one of the recognised go to tool for various NLP tasks, of which we used for our emotional classification task.

Overview of BERT

BERT, a large language model which stands for Bidirectional Encoder Representations from Transformers, is an innovative finding where the model was introduced by AI researches of Google team. Its one of the significant developments in the natural language processing (NLP). It has the ability to understand the context more effectively than previous models. BERT's architecture uses the Transformer model's attention mechanism to understand the relationships within texts, by which leads to major applications in the NLP tasks.

BERT model’s major advantage is its bidirectional approach, where as in previous models were not in both directions like left-right or right-left. It considers context from both directions simultaneously for more meaningful context based on its task.

Bert comes in different sizes, upon the number of the layers and the parameters. Which originally have base and large variations, of which cased and uncased input text. The model we used is bert base uncased with 110M parameters of English language

Setup and Installation

First, we started by setting up the environment for using the BERT using TensorFlow and Hugging Face library. By importing necessary packages and also to clear output for installation for easy understanding of workspace. We loaded the pre-trained BERT model and tokenizer from the libraries we installed. The BERT model we used bert-base-uncased is a lower-cased version with 12 layers, then we tokenize the sample sentences with padding and truncation for uniform input sizes.

The output from the BERT model consists of the last_hidden_state, which gives some detailed token-level embeddings, the pooler_output, which gives summary of each sequence. These outputs are important for tasks such as sequence classification, where the pooled representation can be used as input to classifier.

Data Set and Preparation

The Data set we use is SetFit/emotion dataset is organized into three splits—training, validation, and test—each with features for text, numerical labels, and descriptive labels. The dataset is good fit for training and evaluating emotion classification models, even with a large training set and separate validation and test sets we can achieve some robust model.

The samples the dataset has a variety of emotional sentences attached with corresponding labels. These samples ranging from feelings to confusion and romantic emotions, each labelled with numerical value that corresponds to the specific emotion class.

We prepare the emotion dataset for TensorFlow by converting it into TensorFlow tensors, batching the data, and shuffling the training set for better model generalization. The order function ensures that the data is formatted correctly for BERT, with the necessary input features and labels organized efficiently for training and evaluation.

Fine-tuning and Training

The class we used BERTForClassification(tf.keras.Model) is for defining a custom TensorFlow model for text classification with BERT by adding a dense layer to predict class probabilities. We compiled with the Adam optimizer and sparse categorical cross-entropy loss, and then trained on the dataset for 3 epochs to classify input text into six emotion categories.

Evaluation and Results

The BERT-based classification model we used for our case achieved a test loss of 0.1716 and an accuracy of 92.40% on the test dataset. These results tell us that the model performs effectively in classifying text into the predefined emotion categories, with high accuracy and low error on unseen data.

Plots and Graphs

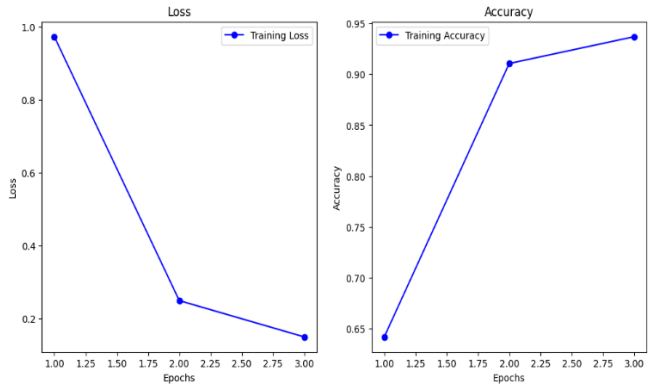

Figure 1: Loss and Accuracy line plot

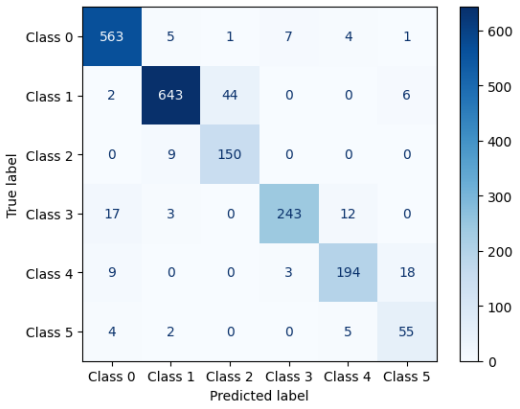

Figure 2: Confusion Matrix

The line plots for loss and accuracy (Figure: 1) clear tells us that how well the model is learning and whether is overfitting or underfitting. Even the confusion matrix plot (Figure: 2) tells model's ability to differentiate between the 6 emotion classes we have in our dataset and also tells where we need to improve the model for more accuracy.

Models Predictions

The model performed well in identifying the emotions of most texts, with a high rate of correct predictions. However, there was some situations where the model confused "joy" with "love" in a contextually ambiguous sentence. This suggests that while the model is generally effective, there is room for improvement in handling nuanced or less explicit emotional expressions.

References

1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
    - Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019). https://arxiv.org/abs/1810.04805

2. Hugging Face Transformers Library
    - Thomas Wolf, Lysandre Debut, Victor Sanh, Alexander Rush (2020). Transformers: State-of-the-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations https://www.aclweb.org/anthology/2020.emnlp-demos.6

3. SetFit/Emotion Dataset
    - HugginFace Datasets. SetFit/emotion dataset. https://huggingface.co/datasets/SetFit/emotion

Github Repo: https://github.com/vka06/LLM-Project