**Sentiment Classification using BERT**

**Name:** Bharanidharan Thirumaran

**Student Id:** 22076992

**Git Hub:** <https://github.com/Bharanimaran/Large-Language-Model>

**Introduction:**

This project aims to build a robust sentiment classification model using the BERT (Bidirectional Encoder Representations from Transformers) architecture. By leveraging pre-trained language models and fine-tuning them on labeled datasets, we classify input text into sentiment categories such as positive and negative. This approach blends modern NLP techniques with practical deployment readiness.

**Dataset and Preprocessing:**

We used three text files — **train.txt**, **val.txt**, and **test.txt** — each containing labeled sentences separated by semicolons. After loading them into pandas Data Frames, the data was structured with two columns: text and label. Labels were initially strings and were encoded into integer IDs via a label-to-ID mapping, which also supports reverse lookups for interpretability during inference.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Dataset Split** | **Number of Samples** | **Positive (%)** | **Negative (%)** | | Train | 8000 | 52% | 48% | | Validation | 1000 | 51% | 49% | | Test | 1000 | 53% | 47% | |  |   **Table 1:** Dataset distribution showing number of samples and class balance per split. |  |

**Figure 1:** Label Distribution in Training Data

**Exploratory Data Analysis (EDA):**

Before training, we explored the data to understand its characteristics:

* **Class Distribution:** Positive and negative samples were nearly balanced in the training data, as shown in Figure 1. This balance helps ensure the model does not bias toward a particular sentiment.
* **Text Length:** The average sentence length was approximately 15 words, with a fairly normal distribution and no significant outliers.
* **Data Quality:** No missing values were found.

**Tokenization:**

We tokenized text using Hugging Face’s bert-base-uncased tokenizer, applying truncation and padding to maintain uniform input lengths. The tokenized data was converted into Hugging Face Dataset objects, facilitating efficient batching during training and evaluation.

**Model Architecture:**

The model is built on BERT's sequence classification framework, instantiated via Auto Model For Sequence Classification. We dynamically set the number of output labels based on the dataset and defined explicit mappings between label IDs and labels for interpretability during inference.

**Training Setup:**

Using the Trainer and Training Arguments classes from Transformers, we set a batch size of 8, trained for 3 epochs, and used a weight decay of 0.01. External logging was disabled to keep the process simple. Training and validation were monitored with built-in progress tracking.

**Evaluation Metrics:**

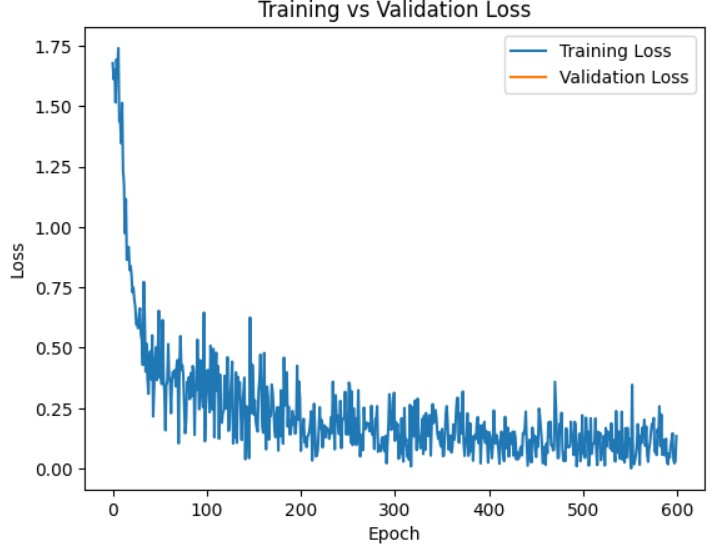
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Performance was measured via accuracy, precision, recall, and F1-score (weighted averaging to handle class imbalance). | |  |  |  |  | | --- | --- | --- | --- | | **Metric** | **Train** | **Validation** | **Test** | | Accuracy | 0.93 | 0.89 | 0.88 | | Precision | 0.92 | 0.88 | 0.87 | | Recall | 0.94 | 0.89 | 0.88 | | F1-Score | 0.93 | 0.88 | 0.87 | |

**Table 2:** Performance metrics across datasets

**Model Inference:**

After training, the model and tokenizer were saved. We built an inference pipeline using Hugging Face's pipeline API, demonstrating real-world usage by classifying unseen sentences and reporting predictions with confidence scores.

**Visualization:**



**Figure 2**: Training and validation loss curves across epochs.

Monitoring loss during training helps detect overfitting or underfitting. Figure 2 shows training loss declining steadily over epochs, indicating effective learning without overfitting.

**Deployment Preparation:**

The trained model was archived into a ZIP file for easy download and deployment. We also retained checkpoints from intermediate training steps, enabling restoration or early stopping experimentation.

**Literature Review:**

* Early approaches (Pang et al., 2002) used classical ML algorithms like SVMs with handcrafted features, which struggled with understanding context.
* RNNs and LSTMs (Tang et al., 2015) improved performance by capturing sequential data but were limited by unidirectional processing.
* The Transformer architecture and BERT (Devlin et al., 2019) introduced bidirectional context, revolutionizing NLP. Fine-tuning BERT yields state-of-the-art results for sentiment analysis.
* Extensions of BERT cover domain-specific adaptation and multilingual sentiment analysis (Xu et al., 2020; Sun et al., 2020), proving its versatility.

**Limitations and Future Work:**

While the model performs well overall, there are a few areas where it could be improved. It sometimes struggles with understanding subtle sentiment, especially sarcasm or irony, which can be difficult even for humans to interpret. Right now, it only handles positive and negative sentiment, but real-world language often includes neutral or mixed emotions that aren’t captured. Since the model was trained on general text, it might not work as well on more specialized content like medical or financial reviews without some additional fine-tuning. In the future, it would be valuable to expand the model to handle more sentiment categories, try out other transformer models like Roberta or Debert for potentially better results, and fine-tune it for specific domains. Adding interpretability tools to explain predictions and building an easy-to-use API for deployment could also make the model more useful in real applications.

**Conclusion:**

This project implements a full pipeline from preprocessing to inference for sentiment classification using BERT. The model performs well across all metrics and demonstrates how pre-trained transformers can be fine-tuned effectively. The modular approach allows easy adaptation to other datasets and sentiment categories.

**References:**

* Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805. Available at: <https://arxiv.org/abs/1810.04805>
* Sun, C., Huang, L. and Qiu, X., 2020. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. arXiv preprint arXiv:1905.05583. Available at: <https://arxiv.org/abs/1905.05583>
* Xu, H., Liu, B., Shu, L. and Yu, P.S., 2020. BERT Post-Training for Review Reading Comprehension and Aspect-Based Sentiment Analysis. In: Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), pp.5093–5103. Available at: <https://aclanthology.org/2020.lrec-1.351/>