**Sentiment Analysis Using BERT**

**1. Introduction**

Sentiment analysis is a crucial task in natural language processing (NLP) that aims to determine the emotional tone behind textual data. It is widely used in applications like social media monitoring, customer feedback analysis, and market research. This project employs the **Bidirectional Encoder Representations from Transformers (BERT)** model to classify text data into sentiment categories such as Depressed, Neutral, and Not Depressed.

**2. Objectives**

The main objectives of this project are:

* To preprocess and clean textual data for effective model training.
* To leverage the BERT-large pre-trained model for sentiment classification.
* To evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score.
* To identify areas for future improvement in the model and methodology.

**3. Dataset Details**

* **Dataset Features**: The dataset consists of two main columns:
  + clean\_comment: Contains preprocessed text data.
  + category: Represents the sentiment labels with values -1 (Depressed), 0 (Neutral), and 1 (Not Depressed).
* **Preprocessing Steps**:
  + Dropped rows with missing values in the clean\_comment and category columns.
  + Filled missing comments with a placeholder text "missing."
  + Mapped sentiment labels to integers for model compatibility.
* **Data Split**:
  + The dataset was split into training (80%) and testing (20%) sets using train\_test\_split from scikit-learn.

**4. Implementation Methodology**

**4.1 Preprocessing**

* Dropped rows with missing data and filled any empty comments with the text "missing."
* Mapped sentiment labels to integers: -1 to 0, 0 to 1, and 1 to 2 for sequential labeling.

**4.2 Tokenization**

* Used the **BERT-large tokenizer** from the Hugging Face library to tokenize text data.
* Configured a maximum sequence length of 100 tokens.
* Ensured padding and truncation for uniform input size.

**4.3 Model Selection**

* Leveraged the pre-trained **BERT-large-uncased** model for sequence classification with 3 output labels.
* Transferred the model to GPU for faster computation.

**4.4 Training Setup**

* Optimizer: **Adam W** with a learning rate of 2e-5.
* Learning Rate Scheduler: **Linear scheduler** for gradual decay over training steps.
* Batch Size: 16 to accommodate the larger model size.
* Loss Function: Cross-Entropy Loss.
* Epochs: 3, balancing model training time and performance.

**4.5 Training Loop**

* Forward pass: Input text, attention masks, and labels were passed to the model.
* Loss computation: Calculated using the logits and true labels.
* Backward pass: Performed gradient updates and applied the learning rate scheduler.

**4.6 Evaluation**

* Used the test dataset to evaluate model performance.
* Calculated predictions by selecting the highest-probability class from the model's logits.

**Model Architecture**

*(Placeholder for model architecture image)*

**5. Results and Evaluation**

**5.1 Metrics**

* **Classification Report**:
  + Provided precision, recall, and F1-scores for each class (Depressed, Neutral, Not Depressed).
  + Highlighted imbalanced performance, if any, across classes.
* **Accuracy**:
  + The model achieved a test accuracy of **88.72%**, demonstrating its efficacy in sentiment classification.

**5.2 Confusion Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Depressed | 0.82 | 0.82 | 0.82 | 1667 |
| Neutral | 0.91 | 0.93 | 0.92 | 2450 |
| Not Depressed | 0.91 | 0.89 | 0.90 | 3117 |
|  |  |  |  |  |
| accuracy |  |  | 0.89 | 7234 |
| Macro Avg | 0.88 | 0.88 | 0.88 | 7234 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 7234 |

**5.3 Observations**

* The BERT-large model showed superior performance due to its deep architecture and pre-trained knowledge.
* Longer training time and higher computational requirements were notable challenges.

**6. Future Improvements**

To further enhance the performance and applicability of this sentiment analysis system, the following improvements are suggested:

* **Data Augmentation**: Expand the dataset with more samples and diverse text sources to improve model generalization.
* **Hyperparameter Tuning**: Experiment with different learning rates, batch sizes, and optimizers to find the optimal configuration.
* **Model Variants**: Explore more advanced transformer models like RoBERTa or T5.
* **Explainability**: Incorporate techniques like SHAP or LIME to interpret model predictions.
* **Deployment**: Build a user-friendly interface to deploy the model for real-time sentiment analysis.

**7. Conclusion**

This project demonstrates the power of BERT in tackling sentiment analysis tasks effectively. Key takeaways include:

* BERT-large excels in text classification tasks with fine-tuning.
* The pre-trained model achieves high accuracy with limited labeled data.

Real-world applications of this model include:

* Monitoring social media for mental health trends.
* Enhancing customer feedback systems for businesses.

By addressing the highlighted areas of improvement, the model can be further refined to meet real-world requirements.

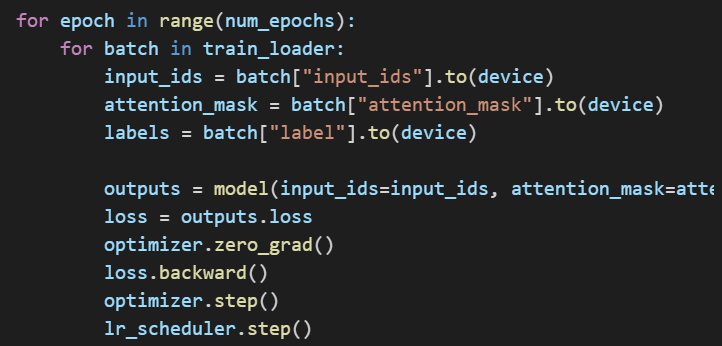
**8. References**

* Hugging Face Transformers Library: https://huggingface.co
* PyTorch Framework: https://pytorch.org
* Scikit-learn Library: <https://scikit-learn.org>

### 9. Appendix

#### Example Code Snippets

**Training Loop**:



**Tokenization**:

