Name: Shreyas Dongre

Email: shreyasgsms@gmail.com

Phone: +91 7738296515

TASK-1

Sentiment Analysis with BERT

The year 2018 marked a significant milestone in Natural Language Processing (NLP) with the advent of transfer learning. This report details the process of implementing sentiment analysis using BERT, a powerful language model developed by Google. The model is hence a pre trained model and the goal is to predict the sentiment (positive, mixed, or negative) of product reviews from an Amazon dataset by fine tuning the model.

Data Preprocessing:

The dataset consists of product reviews from Amazon, stored in a newline-delimited JSON file. Each document in the file includes a "title," "body," and "rating." The task is to predict the sentiment based on the review's title and body. The labels are mapped to three categories: positive (4-star and 5-star reviews), mixed (3-star reviews), and negative (1-star and 2-star reviews).

Baseline Model:

To establish a baseline, a simple Logistic Regression classifier from Scikit-learn is trained using grid search to find the optimal hyperparameter (C). The best baseline classifier achieves an accuracy of 62.67%.

BERT Model:

a. Model Architecture:

BERT (Bidirectional Encoder Representations from Transformers) is chosen for transfer learning. The "transformers" library from HuggingFace is used for easy implementation in PyTorch. BERT models for English include bert-large-uncased (largest), bert-base-uncased, and bert-base-cased.

The English BERT-base model is selected, which uses the uncased version (lowercased text, no accents).

b. Data Preparation:

Each document is represented as a BertInputItem object, including input ids, input mask, segment_ids, and label id. Input ids represent subword units shared across languages, with [CLS] token added at the document's beginning.

Input mask indicates which parts of the input BERT should consider. Segment_ids are used for tasks with multiple input sequences (not applicable in this sentiment analysis task). Label id represents the sentiment label for each document.

c. Training Process:

As this is a fine-tuning task, most of the model parameters are frozen except final layers which are required for fine tuning. AdamW optimizer with a base learning rate of 5e-5 is employed. Gradient Accumulation is utilized to maintain small batches fitting into GPU memory while benefiting from larger effective batch sizes. WarmupLinearScheduler adjusts the learning rate during training, starting small and gradually increasing before decreasing.

5. Evaluation Metrics:

The evaluation method computes precision, recall, and F-score for each batch during training, with a full classification report generated for the test set. Training is performed for a maximum of 100 epochs, with early stopping based on the development set's loss.

6. Results:

BERT achieves an accuracy of around 70% on the test data, surpassing the baseline classifier by approximately 8%. This improvement underscores the efficacy of BERT's transfer learning, enabling higher accuracy even with limited labeled data.

7. Challenges:

Choosing an appropriate BERT model and tokenizer required consideration of trade-offs between size, practicality, and language-specific features. Handling multilingual tokenization nuances, especially for languages with distinct character sets.

Settling down to filter all data to lower case does hamper the performance. Some sarcastic reviews which might be showing a negative polarity but written in a sarcastic way might get mis classified because of all lower-case data. Say for example the review on a game is negative: “I DIED playing this game.” Capitalizing word ‘died’ doesn’t mean that the user liked it. It means that the user hated the game and is trying to convey that but sarcastically. These type of edge cases might get mis classified. Converting whole strings to lowercase might remove this context which would have otherwise helped the model to predict correctly. Further fine tuning, data augmentation is needed to cover all edge cases. Optimizing hyperparameters, such as the learning rate and patience for early stopping, was crucial for effective model training.

The implementation of sentiment analysis using BERT demonstrates the power of transfer learning in NLP. BERT significantly outperforms a baseline logistic regression model, showcasing its ability to generalize knowledge from pretraining to various downstream tasks.

The model's accuracy and its flexibility on test data validates the effectiveness of transfer learning, making it a promising approach for sentiment analysis and similar applications with limited labeled data.