CS 59000 | Natural Language Processing

OBSERVATIONS CRITERIA:

Platform and Environment: The script was developed in a Windows 11 environment using Python version 3.11.6. Due to version mismatches and library corruption issues on the local machine, execution was carried out on Google Colab.

Introduction: The script presents an interactive session allowing users to select one of four functions aimed at leveraging pretrained transformer models from Hugging Face for sentiment analysis on the IMDB dataset.

Approach: The script comprises four functions, each serving a distinct purpose:

batch_predict_sentiment: Predicts sentiment for the test dataset using batch processing to reduce computational load.
calculate_accuracy: Computes the accuracy of the models based on predicted sentiments and actual labels.
predict_gpt2: Utilizes the GPT-2 model and tokenizer to predict sentiment for the given test dataset.
predict_roberta: Utilizes the RoBERTa model and tokenizer for sentiment prediction on the provided test dataset.
main: Acts as the main control block, offering an interactive session for users to select functions and interact with the script.

RESULTS:

The table below summarizes the accuracy achieved by each model:

Model Name	Accuracy
abhishek/autonlp-imdb-roberta-base-3662644	93%
mnoukhov/gpt2-imdb-sentiment-classifier	91%
bert-base-uncased	50%

SUMMARY OF ANALYSIS OF RESULTS:

Performance:

RoBERTa demonstrated superior accuracy compared to the other models, achieving 93%. GPT-2 showcased competitive performance with 91% accuracy, despite being primarily designed for natural language generation. However, BERT's accuracy was notably lower at 50%, indicating potential issues in model initialization or fine-tuning.

Hyperparameters:

Each model may have been trained with specific hyperparameters such as batch size, learning rate, and number of training epochs. These hyperparameters significantly influence model performance and are crucial for achieving optimal results.

Architecture Differences:

RoBERTa, GPT-2, and BERT employ distinct architectures and training objectives, leading to differences in model performance. RoBERTa utilizes a masked language model objective, GPT-2 employs an autoregressive model, and BERT combines masked language modeling and next sentence prediction.

Fine-Tuning Considerations:

Fine-tuning pretrained models on task-specific datasets enhances their performance. RoBERTa and BERT may have undergone fine-tuning on the IMDB dataset, contributing to their higher accuracies compared to GPT-2, which may not have been fine-tuned for sentiment analysis on the same dataset.

Preprocessing and Tokenization:

Variations in tokenization schemes and preprocessing techniques can impact model performance. Each model employs a specific tokenizer tailored to its architecture, potentially affecting how input text is processed and encoded.

Limitations:

Tokenizer Selection and Preprocessing Variability:

The choice of tokenizer, such as the AutoTokenizer from Hugging Face, may influence model performance and tokenization strategies. Different tokenizers have varying

approaches to tokenization, special character handling, and subword splitting, which can introduce inconsistencies in preprocessing and affect model predictions.

Domain and Dataset Dependency:

The performance of pretrained models may vary across different datasets and domains, limiting the generalizability of results. It's crucial to evaluate model performance across diverse datasets to assess their robustness and generalizability.

Model Initialization and Fine-Tuning Requirements:

Fine-tuning pretrained models on task-specific datasets is often necessary to achieve optimal performance. Models like BERT may require fine-tuning on downstream tasks, as indicated by warning messages regarding uninitialized weights. Adequate fine-tuning is essential for adapting pretrained models to specific tasks and improving their accuracy and effectiveness in real-world applications.

Future Utilization:

Future research directions may involve exploring ensemble techniques, hyperparameter tuning, and domain-specific adaptations to further enhance model performance and applicability. Leveraging pretrained models from repositories like Hugging Face can expedite project timelines and facilitate experimentation with various architectures for optimal model selection.