Performance Analysis - Offensive Language Classification

Project Overview

This performance analysis document presents the evaluation metrics of the machine learning models developed for detecting offensive content in online feedback. The models classify text into various labels such as "toxic," "abusive," "vulgar," "menace," "offense," and "bigotry." The results highlight the model's accuracy and overall performance across different metrics.

Dataset Description

The dataset used for this task includes the following files:

1. **train.csv** (Labeled Training Data):
    - `id` : Unique identifier for each comment.
    - `feedback_text` : The comment that needs to be classified.
    - `toxic` : Binary label indicating whether the comment is toxic.
    - `abusive` : Binary label for severe toxicity.
    - `vulgar` : Binary label for obscene language.
    - `menace` : Binary label for threats.
    - `offense` : Binary label for insults.
    - `bigotry` : Binary label for identity-based hate.
2. **test.csv** (Unlabeled data for prediction):
    - The same structure as `train.csv` but without the labels for predictions.

The dataset is structured such that multiple labels may be active for a single comment.

Performance Metrics

The following metrics were used to evaluate the models:

- **Accuracy**: The proportion of correct predictions over the total number of predictions.
- **Precision**: The ratio of true positives to the total predicted positives.
- **Recall**: The ratio of true positives to the total actual positives.
- **F1-Score**: The harmonic mean of precision and recall.
- **AUC-ROC Curve**: The area under the Receiver Operating Characteristic curve, measuring the model's ability to distinguish between classes.

Model Evaluation Results

The model’s performance was evaluated using the following:

- **Baseline Model (Logistic Regression/Random Forest)**: The initial model gave baseline results, providing insights into the overall feasibility of the approach.
- **Advanced Models (LSTM/GRU)**: These models captured sequential dependencies in the text, which led to improvements in accuracy and recall.
- **Transformer-Based Models (BERT/XLM)**: These models provided the best results in terms of accuracy, recall, and F1-score due to their ability to capture contextual relationships in the text.

Model Performance Summary

The models performed as follows:

- **Logistic Regression Accuracy**: 0.72
- **Random Forest Accuracy**: 0.75
- **LSTM Accuracy**: 0.80
- **BERT Accuracy**: 0.85

Observations

- The transformer-based models performed significantly better than the baseline models due to their ability to handle long-range dependencies and context in the feedback text.
- The class imbalance in the dataset was a challenge, but advanced techniques such as data augmentation and model fine-tuning helped mitigate this issue.
- Hyperparameter tuning, including adjusting the learning rate and batch size, contributed to improved model performance.

Repository Structure

Repository/ ├── task/ │ ├── model1_implementation.[ipynb|py] │ └── model2_implementation.[ipynb|py] | |__ Performance_Analysis_Report.pdf | └── README.md

Installation Instructions

1. Clone the repository:

```
git clone <repository_link>
```

2. Install the required dependencies:

```
pip install -r requirements.txt
```

3. Run the Jupyter notebooks for model evaluation:
    - `model1_implementation.ipynb` : Baseline and advanced models.
    - `model2_implementation.ipynb` : Transformer-based models.

License