Advanced Deep Learning
AIGC 5500
Final Project
Comparative Analysis of Deep Learning Models for Sentiment Analysis on Yelp Reviews

Project Overview

In this project, you will develop, train, and compare two deep learning models, **LSTM** and **DistilBERT**, to analyze sentiment in Yelp reviews of restaurants or hotels. Your task is to determine the overall sentiment (positive, negative, neutral) expressed in each review and compare the strengths and weaknesses of these models on this dataset. By focusing on subjective opinions in reviews, you will explore how well each model captures sentiment through various text lengths and vocabulary specific to the hospitality industry.

Project Objectives

1. Build and preprocess a dataset of Yelp reviews, focusing on sentiment classification.
2. Develop and train **two models**—one based on LSTM and one based on DistilBERT.
3. Evaluate the performance of each model on unseen Yelp reviews, analyzing accuracy, precision, recall, and F1 score.
4. Analyze and compare the strengths of each model in terms of their performance on different review types (e.g., short vs. long reviews, specific vocabulary usage).
5. Visualize model interpretability and classification patterns for each model.
6. Conduct hyperparameter tuning to optimize model performance.

Detailed Project Instructions

1. **Data Collection and Preprocessing**
    - Use a Yelp review dataset with sentiment labels or preprocess an available dataset of Yelp reviews, labeling the sentiment (positive, negative, neutral).
    - Clean the text data, remove irrelevant characters or symbols, and tokenize the text as appropriate for each model.
    - Consider using word embeddings for the LSTM model and token embeddings for DistilBERT.

- Split the data into training, validation, and test sets, ensuring a balanced distribution of sentiment labels.

2. **Model Development**
    - **LSTM Model**: Build a Long Short-Term Memory (LSTM) model designed for sequence learning. Define an appropriate architecture, considering embedding layers, hidden units, and regularization layers.
    - **DistilBERT Model**: Implement a DistilBERT model for sentiment classification. Use pre-trained weights and fine-tune the model on your dataset.
    - Ensure that both models are set up to classify reviews into three sentiment categories: positive, negative, and neutral.
3. **Hyperparameter Tuning**
    - Experiment with hyperparameters for each model, such as learning rate, batch size, and dropout rates, to identify the best configuration.
    - Document the impact of different hyperparameters on each model's performance metrics.
4. **Model Evaluation**
    - Evaluate each model's performance on the test set using the following metrics:
        - **Accuracy**
        - **Precision**
        - **Recall**
        - **F1 Score**
    - Use a confusion matrix to visualize each model's performance across different sentiment classes.
5. **Interpretability and Analysis**
    - Compare each model's performance on different types of reviews, such as:
        - **Short vs. Long Reviews**: Analyze how each model performs on reviews of varying lengths.
        - **Vocabulary Analysis**: Examine how vocabulary influences model predictions, identifying words or phrases that each model associates strongly with each sentiment class.
    - Utilize interpretability tools (e.g., LIME, SHAP) to visualize and understand how each model classifies sentiment within reviews.

6. **Visualization and Reporting**
    - Create visualizations that illustrate model interpretability, such as attention weights in DistilBERT or specific word influences in LSTM.
    - Summarize findings with visual comparisons, tables, and graphs that clearly depict the results and insights.

Deliverables

Your final submission should include the following components:

1. **Professional Report (PDF format)**: A structured report that includes:
    - **Introduction**: Overview of the project, goals, and significance of sentiment analysis.
    - **Dataset Description and Preprocessing**: Summary of the dataset, data preprocessing steps, and the rationale behind each choice.
    - **Model Descriptions**: Explanation of the LSTM and DistilBERT models, including architectures and hyperparameter choices.
    - **Results and Findings**: Detailed results from training, validation, and testing, presented in tabular and graphical formats.
    - **Analysis and Discussion**: Interpretation of results, comparison of model strengths and weaknesses, and insights gained from interpretability analysis.
    - **Conclusion**: Summary of key findings, final remarks on model effectiveness, and possible future improvements.
    - **References**: Citations for any external resources, papers, or tools referenced in the project.
2. **Code Submission**:
    - **Python Scripts and Notebooks** (.py or .ipynb files): Submit well-documented scripts for all code used in the project. Include comments to explain the purpose and function of each section.
    - **ReadMe File**: A brief ReadMe explaining how to execute the code, dependencies required, and any setup instructions.
    - **Output Display**: Ensure that your .ipynb notebook shows the results directly within the notebook for ease of review.
3. **Group Responsibility Breakdown**

- A brief outline indicating each team member’s role and contributions, from data preprocessing to model analysis.

Evaluation Criteria

Your project will be evaluated based on the following:

1. **Technical Implementation**: Completeness, correctness, and efficiency of your LSTM and DistilBERT models, as well as hyperparameter tuning.
2. **Analysis and Insights**: Depth of your comparative analysis, clarity of insights into model performance, and effectiveness in visualizing interpretability.
3. **Code Quality and Documentation**: Code readability, structure, and comments that clarify each part of the project.
4. **Report Quality**: Professional presentation, organization, and clarity of your written report.
5. **Teamwork**: Clear evidence of each group member’s contributions and roles.s