Assignment 1 (Part 2):

Natural Language Processing

Uzair Ahmad

Text Classification Assignment: Movie Review Sentiment Analysis

Objective:

The goal of this assignment is to implement and compare three text classification algorithms—Naive Bayes, Logistic Regression, and Multilayer Perceptron (MLP)—on the NLTK Movie Reviews dataset. You will explore the impact of using both raw Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) as feature representations.

Tasks:

1. Data Preparation (3 marks):

- Load the NLTK Movie Reviews dataset.
- Here's how you can obtain the IMDb movie reviews dataset:

  Download the Dataset:

  - You can download the dataset from the NLTK library. NLTK provides a convenient interface to access this dataset.

```
import nltk
# Download the IMDb movie reviews dataset
nltk.download('movie_reviews')
```

  Access the Dataset:

  - Once you have downloaded the dataset, you can access the movie reviews and their corresponding labels using the following code:

```
from nltk.corpus import movie_reviews

# Access the movie reviews and labels
documents = [(list(movie_reviews.words(fileid)), category)
             for category in movie_reviews.categories()
             for fileid in movie_reviews.fileids(category)]

# Shuffle the documents to ensure a balanced distribution of positive and
negative reviews
import random
random.shuffle(documents)
```

  Explore the Dataset:

- Take a look at the structure of the dataset and sample reviews to understand its characteristics.

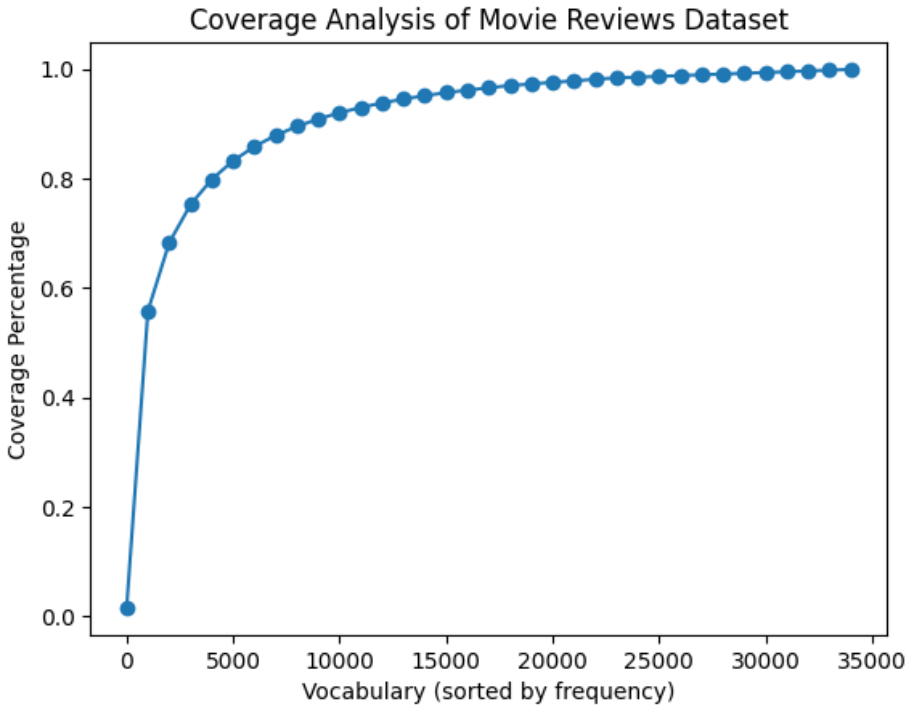
```
# Print the first review and its label
print("Sample Review:", documents[0][0][:10])  # Displaying the first 10 words for brevity
print("Label:", documents[0][1])
```

This IMDb movie reviews dataset is suitable for sentiment analysis tasks, and it provides a good balance between positive and negative reviews.

- Preprocess the dataset by tokenization (use nltk punkt tokenizer), stemming/lemmatization, and removing stop words.

2. Coverage Analysis Insights (2 marks):

- Conduct a coverage analysis to identify the percentage of unique words covered by the preprocessing steps.
- Visualize the coverage analysis with the y-axis representing coverage percentage and the x-axis representing the number of tokens (words) considered. Use a line plot for clarity.



- Discuss the insights gained from the coverage analysis. Consider questions such as:
  - How does the coverage change with the number of tokens considered?
  - At what point does the coverage seem to stabilize?
  - Are there diminishing returns in terms of coverage as the number of tokens increases?

Rationalization for Vocabulary Choice :

- Discuss the rationale for choosing a specific vocabulary size for modeling. Consider factors such as:
  - The trade-off between a larger vocabulary (more words) and computational efficiency.
  - The impact of rare or very common words on the model's generalization.
  - The need to balance informativeness and model complexity.

- Any specific considerations for the chosen algorithms (Naive Bayes, Logistic Regression, MLP) in terms of vocabulary size.

3. Algorithm Implementation (6 marks):

a. **Naive Bayes:**
- Implement a Multinomial Naive Bayes classifier.
- Train and test the model using both TF and TF-IDF as feature representations.

b. **Logistic Regression:**
- Implement a Logistic Regression classifier.
- Train and test the model using both TF and TF-IDF.

c. **Multilayer Perceptron (MLP):**

- Implement an MLP-based classifier.
  Explore different architectures (number of layers, neurons per layer).
  - Train and test the model using both TF and TF-IDF.

4. Training and Evaluation (4 marks):

a. Train each algorithm on the training set.

b. Evaluate the performance of each algorithm on the testing set using accuracy, TPR, FPR as the primary metrics.

c. Compare the impact of using TF and TF-IDF on each algorithm's performance.

5. Visualization and Analysis (2 marks):

a. Visualize the performance metrics (e.g., accuracy) for each algorithm using appropriate plots (e.g., bar chart).

b. Discuss any observed trends or differences in performance.

6. Discussion (3 marks):

a. Compare and analyze the results obtained by the three algorithms.

b. Discuss the impact of using TF vs. TF-IDF on classification performance.

c. Provide insights into the strengths and limitations of each algorithm in the context of sentiment analysis.

Submission Guidelines:

- Submit the Python code for each algorithm implementation.
- Include a README file with instructions on how to run your code and any dependencies.
- Submit a report document (PDF) containing detailed explanations, visualizations, and comparative analysis.