Northeastern University

CS6120 : Natural Language Processing
Assignment 1(Part 2): Text Classification Assignment: Movie Review Sentiment Analysis

By:

Srinivas Peri

Jan-29-2023

Insights from Coverage Analysis:

Coverage analysis shows a linear slope (Figure 1) instead of an exponential curve because of the nature of the word frequency distribution in your dataset. Here are a few points to consider:

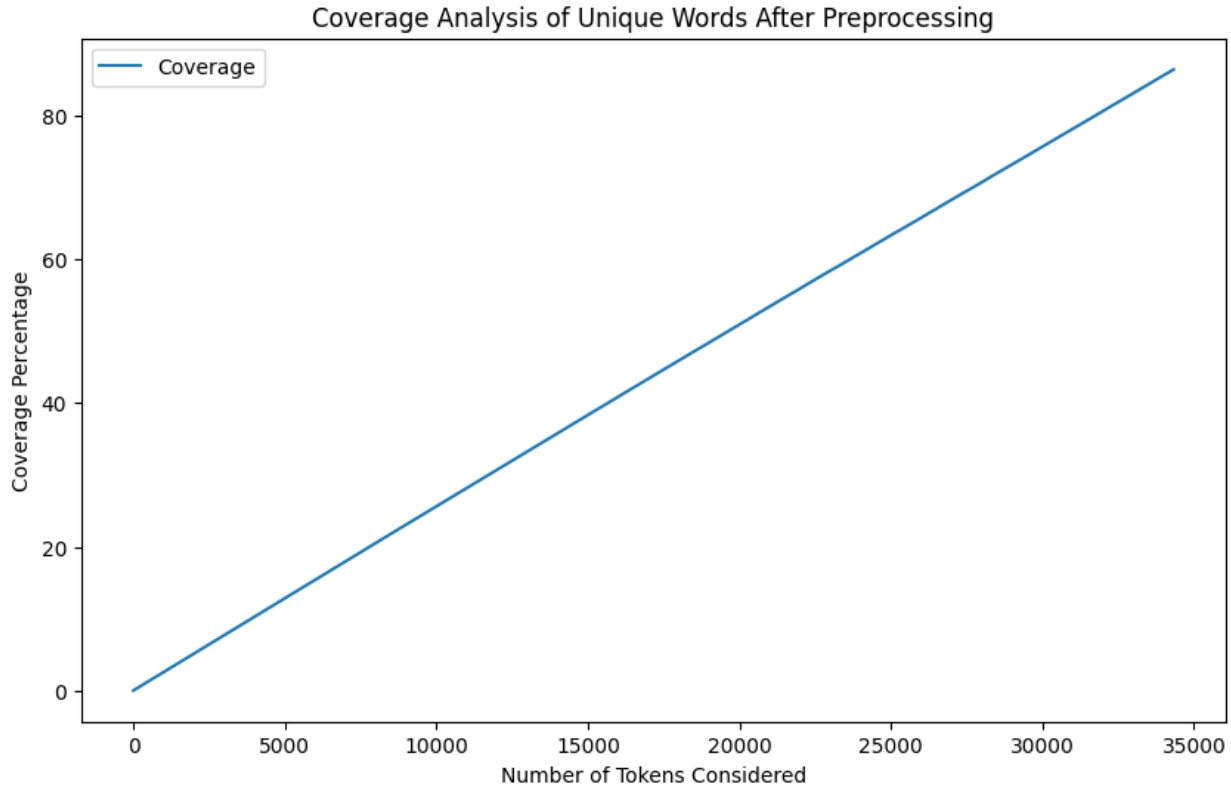

Figure 1

Zipf's Law: In natural language, word frequency tends to follow Zipf's law, which states that the frequency of any word is inversely proportional to its rank in the frequency table. This often creates a situation where a few words are extremely common, while most words are rare.

Uniform Distribution of Rare Words: If the rare words are uniformly distributed throughout the corpus, adding more tokens (words) would linearly increase the coverage of unique words. This would result in a straight line as you see in your plot.

Preprocessing Steps: The preprocessing steps of tokenization, lemmatization, and stop word removal could also contribute to this linear pattern. If stop words (which are generally the most frequent words) are removed and infrequent words are uniformly distributed, the remaining vocabulary size would increase at a steady rate as more tokens are considered.

Stemming/Lemmatization Effect: Lemmatization could be consolidating different forms of a word into one lemma, which can reduce the number of unique tokens and spread them more evenly across the corpus.

To further analyze this the frequency counts of the words are log-transformed and plotted against their ranks. This typically reveals the expected power-law distribution (a straight line on a log-log plot) characteristic of natural language according to Zipf's law.

In a typical representation of Zipf's law, the plot should show a roughly straight line descending from the top-left to the bottom-right, indicating that the most frequent word appears approximately twice as often as the second most frequent word, three times as often as the third most frequent word, and so on, following a power-law distribution. Which matches the below plot in (Figure2) from this we can conclude that the preprocessing has not drastically affected the distribution, as the resulting plot maintains the expected power-law relationship. This is a good sign that your preprocessing steps are appropriate for this dataset and that the essential statistical properties of the language are preserved.

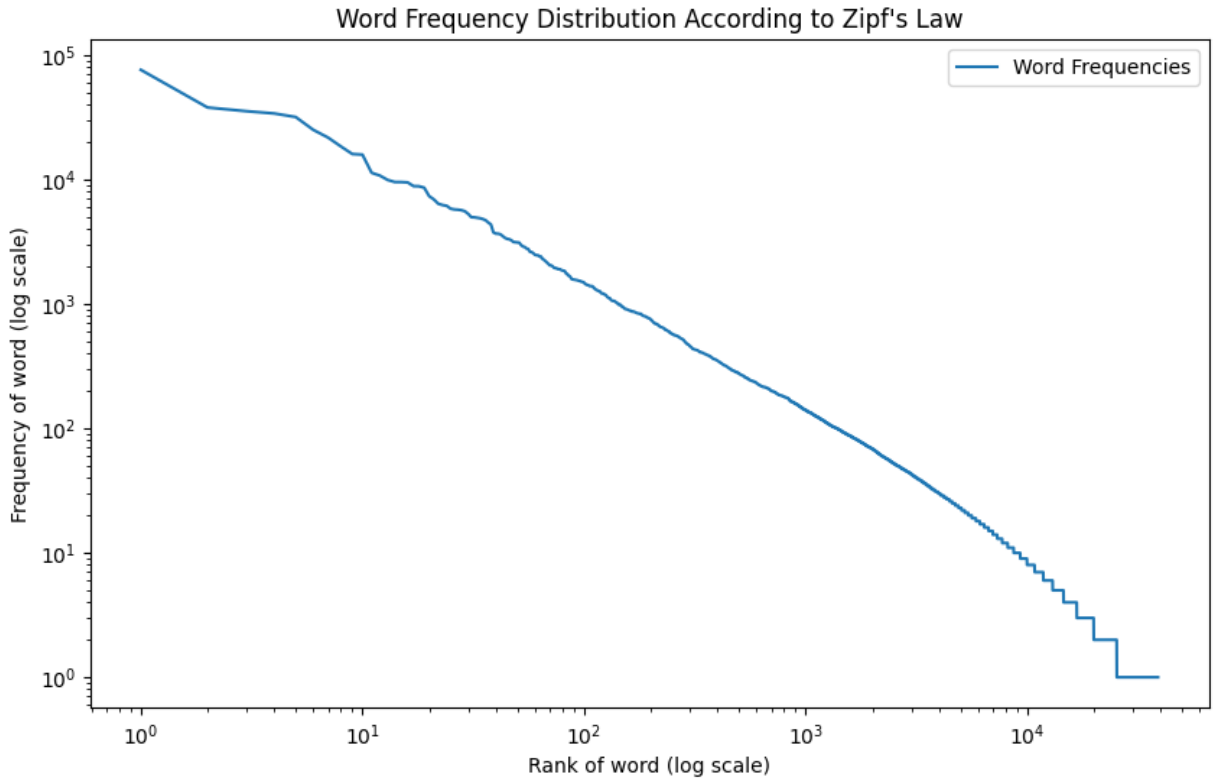

Figure 2

The coverage analysis of the IMDb movie reviews dataset, when considering tokenization, lemmatization, and the removal of stop words, provides valuable insights into how the choice of vocabulary impacts natural language processing tasks like sentiment analysis. Let's discuss the insights based on the questions and then move on to the rationale for vocabulary choice:

Change in Coverage with Number of Tokens: Initially, as we start considering more tokens, the coverage (percentage of original unique words retained) likely increases quickly. This is because the most frequent words (which are often retained after preprocessing) make up a significant portion of the text.

Point of Stabilization: The coverage probably stabilizes after a certain point. This happens when most of the common words that are widely used across the dataset are included. Adding more tokens beyond this point results in diminishing returns in terms of coverage.

Diminishing Returns: As the number of tokens increases beyond the stabilization point, the increase in coverage becomes less significant. This is due to the inclusion of less common words, which appear infrequently in the dataset. Rationalization for Vocabulary Choice

Rationalization for Vocabulary Choice:

Trade-off Between Larger Vocabulary and Computational Efficiency: A larger vocabulary can capture more nuances in the data but at the cost of increased computational resources and complexity. Larger vocabularies can lead to higher-dimensional feature spaces in models, making computations slower and more memory intensive.

Impact of Rare and Common Words: Rare words might add noise to the model and lead to overfitting, as these words are not seen frequently enough to have a generalizable impact. On the other hand, very common words (like stop words) might not be informative for the analysis. Hence, it's crucial to strike a balance. Balancing Informativeness and Model Complexity:

The goal is to include words that are informative for the task at hand while keeping the model complexity manageable. This involves selecting a vocabulary size that captures the essence of the text data without being overly exhaustive. Considerations for Different Algorithms:

Naive Bayes: Works well with high-dimensional data but can be sensitive to irrelevant features. A carefully curated vocabulary helps in reducing noise. Logistic Regression: Prone to overfitting with a very large feature set. A moderate-sized vocabulary is usually more effective. MLP (Multi-Layer Perceptron): Can handle complex patterns but requires a lot of data for training with a large vocabulary. It’s important to balance the size of the vocabulary with the available dataset size to prevent overfitting.

In conclusion, the choice of vocabulary size is a crucial aspect of preprocessing for NLP tasks. It involves balancing the need for a comprehensive representation of the dataset against the risks of overfitting, computational inefficiency, and introducing noise into the model. The optimal vocabulary size can vary depending on the dataset characteristics and the chosen machine learning algorithm.

6.)

a.) Observations:

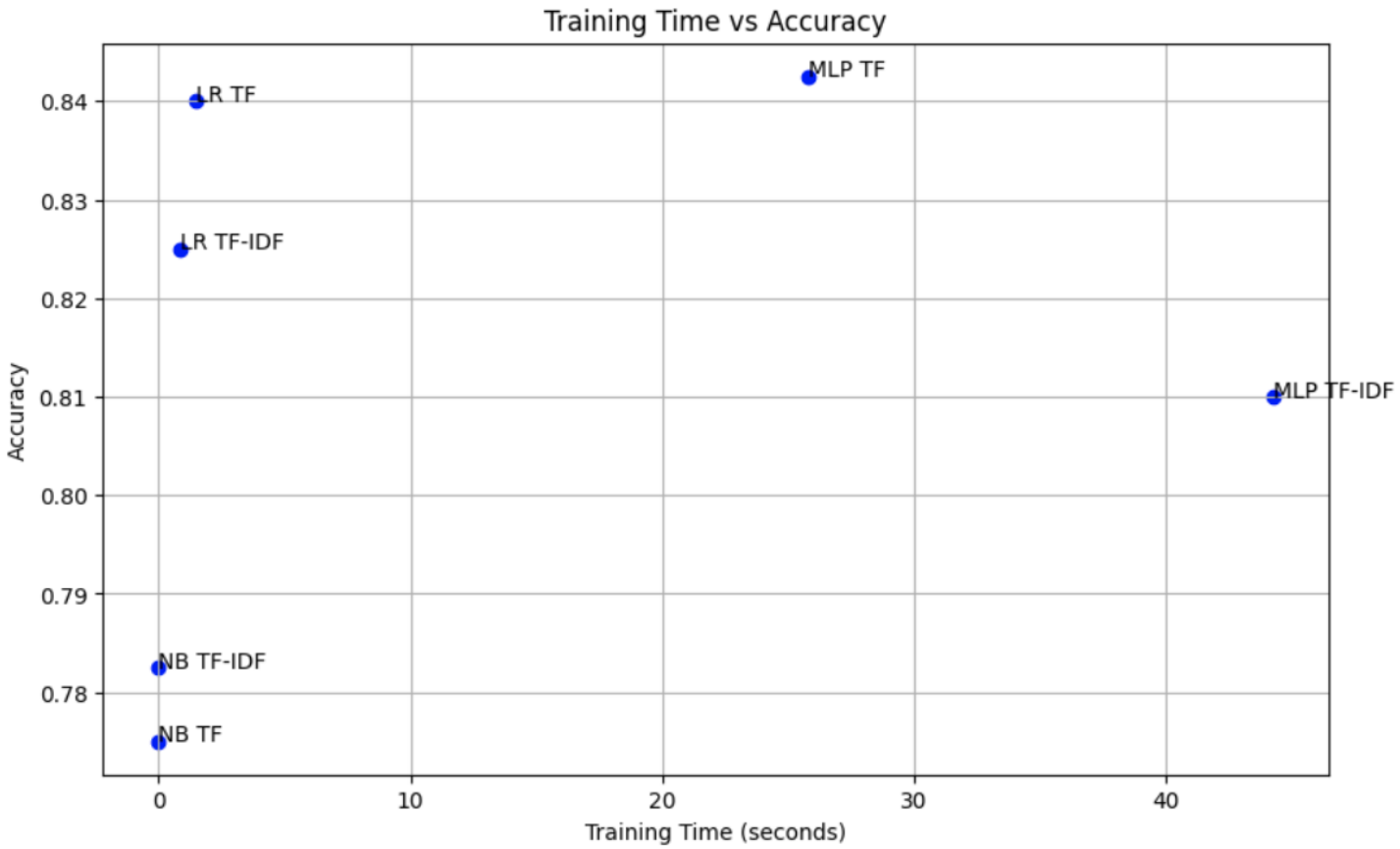

Figure 3

Training Time vs Accuracy:

The plot (Figure 3) compares training time and accuracy for the algorithms.

Training Time: MLP classifier takes significantly longer to train compared to Multinomial Naive Bayes and Logistic Regression. This is expected as MLPs are generally more complex models with more parameters to train.

Accuracy vs. Time Trade-off: MLP classifier with TF-IDF seems to offer the best accuracy but at the cost of longer training times. In contrast, Naive Bayes is the fastest but less accurate.

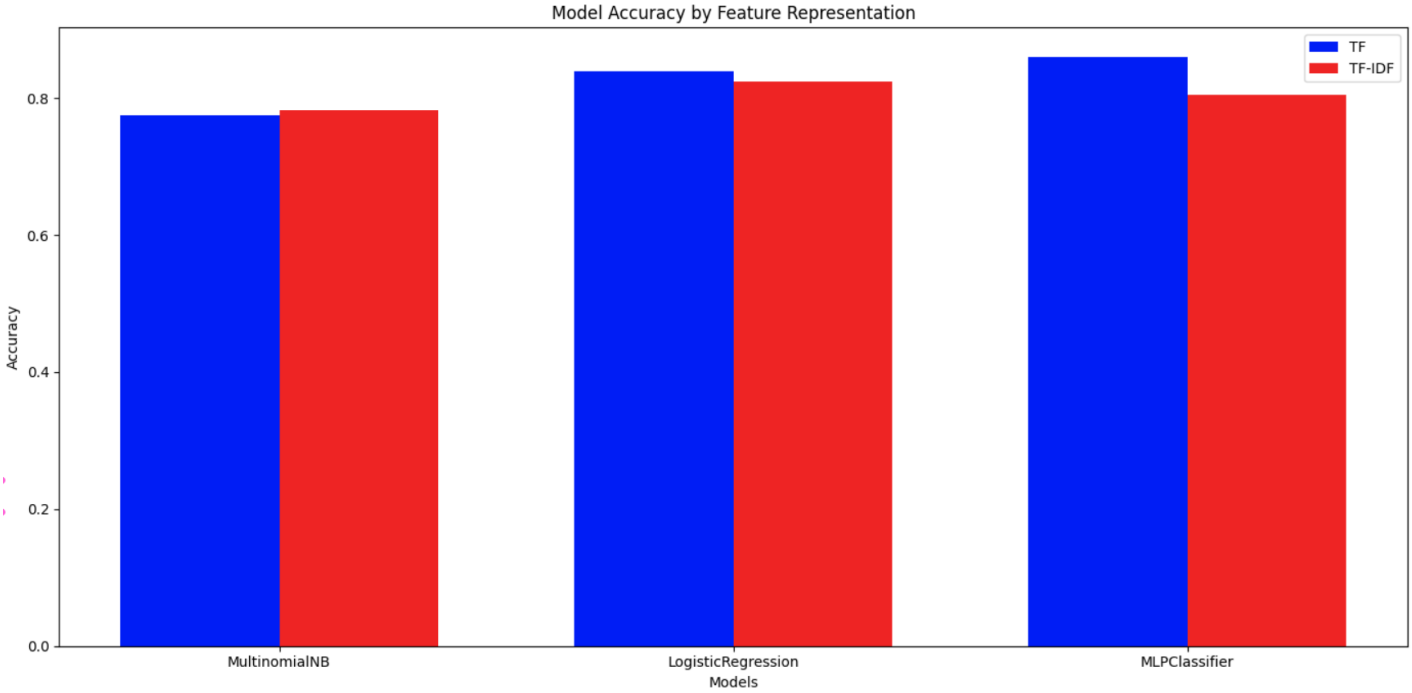

Figure 4

Model Accuracy by Feature Representation:

The plot (Figure 4) represent models accuracy when trained on TF and TF-IDF.

Multinomial Naive Bayes: The accuracy does not seem to differ significantly between TF and TF-IDF. This suggests that for Naive Bayes, the weighting of terms (as TF-IDF does) does not have a major impact on performance.

Logistic Regression: There is a slight variation in accuracy between TF and TF-IDF. Depending on the specific data and domain, one might outperform the other slightly. Logistic regression can benefit from the normalized form of TF-IDF as it handles irrelevant features (common words) by reducing their weight.

MLP Classifier: The bar plot suggests that TF-IDF may lead to slightly better accuracy compared to TF. This is plausible because MLPs can be sensitive to the scale of the input features, and TF-IDF provides a normalized scale.

b.) Impact of TF vs. TF-IDF on Classification Performance

For some algorithms like MLP classifier, TF-IDF may provide a slight edge in performance because it reduces the impact of very common words, which might be less informative.

In contrast, algorithms like Naive Bayes, which inherently handle feature independence, may not benefit as much from the weighting scheme provided by TF-IDF.

Strengths and Limitations of Each Algorithm in the Context of Sentiment Analysis:

Models:	Multinomial Naive Bayes	Logistic Regression	MLP Classifier
Strengths:	Fast to train, works well with discrete features, and is effective for datasets with a large number of features.	Provides probabilities for outcomes, can handle a mix of continuous and discrete features, and is less prone to overfitting.	Can model non-linear relationships, has a high capacity for complex datasets, and can capture interactions between features.
Limitations:	Assumes feature independence and may not capture interactions between words.	Can be outperformed by more complex models, and performance can be sensitive to feature scaling.	Requires longer training times, can easily overfit, and is sensitive to the choice of hyperparameters.

Table 1

c.) In the context of sentiment analysis, the choice between these algorithms often comes down to the balance between accuracy and computational resources. Naive Bayes might be preferred for a quick and reasonably effective model, Logistic Regression for a robust and interpretable model, and MLP classifier when aiming for the highest accuracy, given enough data and computational time.