Data Preprocessing and Featurization for NLP Classification

Kashyap Raparthi

May 1, 2024

1 Introduction

In this report, we describe the data preprocessing and featurization steps adopted for the BBC articles dataset for Natural Language Processing classification tasks.

2 Data Preprocessing

The preprocessing steps included:

- Reading each document from the BBC_articles folder to extract article ID, category, and text.
- **Lowercasing**: Lowercasing the text ensures that words are treated uniformly. It prevents the model from treating the same word with different cases as different features, improving generalization.
- **Removing Punctuation Marks**: Removing punctuation marks helps clean the text and ensures that they do not interfere with the tokenization process. Punctuation marks do not usually carry meaningful information for classification tasks.
- **Tokenization**: Tokenization breaks the text into individual words or tokens. It is essential for the model to understand the structure of the text and identify meaningful units for analysis.
- **Removing Stopwords**: Stopwords are common words such as "and," "the," "is," etc., that occur frequently but do not carry significant meaning. Removing stopwords reduces the dimensionality of the data and focuses on the words that are more informative for classification.

3 Featurization

For featurization, we used the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer. TF-IDF is a numerical statistic that reflects how important a word is to a document in a collection. It is widely used for text feature extraction in NLP tasks.

TF-IDF is often a good choice as it tends to perform well in capturing the essence of the text for classification purposes. It considers both the frequency of words in a document and their importance in the entire corpus, which can be beneficial for distinguishing between different categories. Additionally, TF-IDF vectors are typically sparse, which can be advantageous for memory efficiency.

4 Code Implementation

The Python code was implemented using the pandas library for data manipulation, nltk for text preprocessing, and sklearn for the TF-IDF vectorization. Only the top 100 features were considered in order to perform vectorization.

5 Instructions

To run the code, ensure you have the following dependencies installed:

- pandas
- nltk
- scikit-learn
- punkt
- stopwords

You can install them using pip:

```
pip install pandas nltk scikit-learn
```

Ensure that the 'BBC_articles' folder containing the text files is in the same directory as the Python script. After running the script, you will get a CSV file named 'vectorized_dataset.csv' containing the vectorized dataset.