

Document Categorization using NLP

Project Overview

This project focuses on document classification using the 20 Newsgroups dataset, a collection of approximately 20,000 newsgroup documents partitioned across 20 different newsgroups. The project involves data pre-processing, text analysis, visualization, and classification to accurately categorize documents into their respective topics.

Key Components

1. Dataset Loading and Exploration

The dataset is loaded using scikit-learn’s fetch\_20newsgroups function. Both training and test subsets are loaded with headers, footers, and quotes removed to ensure cleaner and more relevant text data. Initial exploration includes reviewing sample articles and their corresponding categories to familiarize with the dataset structure.

2. Text Preprocessing

The preprocessing pipeline consists of:

* Lowercasing all text for uniformity
* Removing non-alphanumeric characters while retaining spaces
* Tokenizing text into individual words using NLTK
* Removing English stopwords to reduce noise and improve model focus

This pipeline is applied to both training and test datasets to maintain consistency.

3. Target Categories

The 20 newsgroups represent diverse topics, such as:

* Technology and computing (e.g., comp.graphics, comp.sys.mac.hardware)
* Recreation and hobbies (e.g., rec.autos, rec.sport.baseball)
* Science and research (e.g., sci.crypt, sci.med)
* Politics and religion (e.g., talk.politics.misc, soc.religion.christian)

4. Data Visualization

A word cloud visualization is created from the combined preprocessed text of the dataset. This highlights the most frequently occurring words, providing insight into common themes and important keywords across newsgroup discussions.

Technical Implementation

Libraries Used

* NLTK: For text tokenization and stopword removal
* scikit-learn: For dataset loading and model implementation
* Matplotlib: For plotting visualizations
* WordCloud: For generating word cloud visualizations

Models Used

* Logistic Regression: Applied as a linear classifier suitable for high-dimensional text data, leveraging probability estimates to assign document categories.
* Multinomial Naive Bayes: A probabilistic classifier that performs well with word frequency features, serving as a strong baseline for text classification tasks.

Model Training and Evaluation

The preprocessed texts were converted into numerical features (e.g., TF-IDF vectors) to train both Logistic Regression and Multinomial Naive Bayes models. Their performance was evaluated using metrics including accuracy, precision, recall, and F1-score to determine the best fit for document classification.

Potential Next Steps

1. Advanced Feature Engineering: Explore word embeddings (Word2Vec, GloVe) or transformer-based embeddings (BERT) for richer text representation.
2. Deep Learning Models: Implement neural networks such as LSTM or CNN for potentially improved classification accuracy.
3. Hyperparameter Tuning: Optimize model parameters using grid search or randomized search techniques.
4. Ensemble Methods: Combine multiple classifiers to enhance prediction robustness.
5. Deployment: Develop a user-friendly web application that allows real-time document classification.

Conclusion

This project establishes a robust pipeline for document classification on a real-world text dataset. Effective preprocessing and exploratory visualization enable cleaner data representation, while Logistic Regression and Multinomial Naive Bayes models provide strong baseline classification results. Future enhancements will focus on richer feature extraction and more sophisticated modeling techniques to further improve classification accuracy and usability.