CS 410 Project Report

etz3@illinois.edu

1. Overview of code

My code performs sentiment analysis on IMDb movie reviews. It starts by cleaning and preprocessing the text data, using BeautifulSoup for HTML parsing (in case there are html tags in the text), regular expressions for text cleaning, and NLTK for tokenization, lemmatization, and stopword removal. The TF-IDF Vectorization converts the processed text into numerical features. Then, I used a Multinomial Naive Bayes classifier to train the dataset for sentiment prediction. Cross-validation is used to evaluate the model's accuracy and generalization. I also allow users to interactively input movie reviews for real-time sentiment predictions, showcasing practical applications in user-driven sentiment analysis. This tool is valuable for understanding sentiment patterns in movie reviews and can potentially be extended for broader text analysis tasks.

2. Documentation of software implementation

The clean_text(text) function uses BeautifulSoup for HTML parsing, regular expressions for text cleaning, and NLTK for converting text to lowercase, removing special characters, numbers, and stopwords. It returns the cleaned and preprocessed text.

The tokenize_and_lemmatize(text) function tokenizes the text using NLTK's word_tokenize, lemmatizes tokens using NLTK's WordNetLemmatizer, and excludes stopwords. It returns the lemmatized text.

The next section performs TF-IDF vectorization on the cleaned data, and trains a Multinomial Naive Bayes classifier. It returns the trained classifier.

The cross_validation section takes TF-IDF features (X_tfidf) and labels (y) as input, uses a Multinomial Naive Bayes classifier for cross-validation, and prints cross-validation scores and mean accuracy.

The rest of the program which is essentially the main() function reads the IMDb dataset, initializes NLTK resources, and calls functions sequentially: train_sentiment_model, perform_cross_validation, and the user interaction loop using predict_sentiment. This is the part that allows for the user to input a sample movie review and my program will tell if it is good or bad.

3. How to run software

To use my program, first ensure Python and required libraries are installed. Download the IMDb dataset and the Python file. Make sure to change the file path for the IMDb dataset to wherever you saved the dataset (line 12 in the Python file). Then all you have to do is run the script, which automatically preprocesses the data, trains a sentiment analysis model, and performs cross-validation. After, you can interactively input movie reviews to receive real-time sentiment predictions. No external APIs are utilized. To install dependencies, you can use pip install beautifulsoup4 pandas scikit-learn nltk.

4. Contributions of each team member

I did the project on my own.