Sentiment Analysis in Amazon Customer Reviews – Oorja Dorkar

Introduction

Sentiment analysis, a branch of Natural Language Processing (NLP), focuses on extracting subjective information from documents to ascertain opinions regarding products, services, events, or ideas. This project leverages customer review data from Amazon.com to perform a supervised binary sentiment classification, distinguishing between positive and negative reviews. We analyze the impact of various data preprocessing techniques and compare the effectiveness of three machine learning models: Multinomial Naive Bayes (MultinomialNB), Logistic Regression (LogisticRegression), and Linear Support Vector Classification (LinearSVC). The results show that incorporating negation handling and n-grams significantly enhances model accuracy, with LinearSVC achieving the highest prediction accuracy.

Data

Data Source

The dataset, obtained from "Amazon product data" managed by Dr. Julian McAuley of UCSD, comprises 982,619 Kindle store reviews from May 1996 to July 2014, formatted in JSON.

Sentiment Labeling

Reviews are categorized as negative ("neg") for ratings of 1-3 and positive ("pos") for ratings of 4-5, leading to 829,277 positive and 153,342 negative reviews.

Undersampling

To address the imbalance where 84.4% of reviews are positive, we undersample positive reviews to equal the number of negative ones.

Preprocessing

- **HTML Parsing**: Convert HTML entities to text.
- **Negation Handling**: Modify contractions (e.g., changing "can't" to "cannot") to prepare for negation handling, where terms following negations are prefixed with "not_".
- **Tokenization and Normalization**: Tokenize words and convert to lowercase.
- **N-gram Modeling**: Utilize bigrams and trigrams to capture information conveyed by sequences of words.
- **Lemmatization**: Reduce words to their base forms.

The data is split into training, validation, and testing sets with ratios of 60%, 20%, and 20%, respectively.

Feature Extraction

Using the Bag of Words approach, we tokenize the text and transform it into numerical feature vectors. This includes counting token occurrences and applying TF-IDF weighting to diminish the importance of tokens appearing frequently across documents.

Model Comparison

We evaluate the three models with basic preprocessing, addition of negation handling, and inclusion of n-grams:

Preprocessing Added	Number of Features	MultinomialNB	LogisticRegression	LinearSVC
Basic preprocessing	56,558	0.8329	0.8453	0.8485
Negation handling	71,853	0.8262	0.8519	0.8562
Bigrams and Trigrams	2,027,753	0.8584	0.8675	0.8731

Feature Selection

Further tuning is performed to optimize feature count, revealing that LinearSVC consistently outperforms the others:

Model	Best Features	Validation Accuracy	Testing Accuracy
MultinomialNB	1,000,000	0.8580	0.8585
LogisticRegression	500,000	0.8697	0.8682
LinearSVC	1,700,000	0.8746	0.8730

Discussion

Notably, attempts to remove stopwords resulted in decreased accuracy, possibly due to the generic nature of stopwords in the NLTK package. Considering the dataset's size and the use of TF-IDF, removing stopwords might be unnecessary. Although we performed coarse feature tuning, more detailed adjustments could potentially enhance model performance. Further exploration could include tuning penalty parameters or employing more complex models like Long Short-Term Memory (LSTM) networks.

This report underscores the potential of specific NLP techniques in improving sentiment analysis, particularly through sophisticated preprocessing and strategic model selection. Future studies could refine these approaches for even greater accuracy and applicability.