LING 406 Spring 2024 – Term Project Report

Nivedita Rethnakar

Abstract

This project implements a sentiment analysis system using a Naive Bayes classifier to categorize movie reviews as positive or negative. Through preprocessing, feature engineering, and model training, it aims to accurately capture the sentiment expressed in textual data, offering insights into the effectiveness of machine learning approaches in sentiment analysis tasks.

1 Introduction

Sentiment analysis has emerged as a crucial component in today's digital landscape, playing a pivotal role in understanding and harnessing the vast amount of textual data generated daily. In an era where social media, online reviews, and customer feedback are ubiquitous, sentiment analysis offers invaluable insights into public opinion, customer satisfaction, and market trends. Businesses leverage sentiment analysis to gauge customer sentiment towards their products and services, enabling them to make data-driven decisions, improve customer experience, and tailor marketing strategies effectively. Additionally, sentiment analysis plays a vital role in brand monitoring, crisis management, and reputation management, allowing organizations to proactively address issues and maintain a positive brand image.

Beyond the commercial realm, sentiment analysis finds applications in diverse fields such as politics, healthcare, and social sciences. In politics, sentiment analysis helps analyze public sentiment towards political candidates, policies, and societal issues, aiding policymakers in understanding public perception and shaping their strategies accordingly. In healthcare, sentiment analysis of patient reviews and feedback provides valuable insights into healthcare service quality, patient satisfaction levels, and areas for improvement. Furthermore, sentiment analysis contributes to sentiment-based recommendation systems, personalized content delivery, and sentiment-aware decision-making across various domains, making it a crucial tool in today's data-driven society.

2 Problem Definition

Sentiment analysis, within the realm of Computational Linguistics, is the process of automatically identifying and categorizing sentiments or opinions expressed in text as either positive or negative. In this context, the task is to implement a sentiment analyzer that can accurately classify input text, particularly movie reviews, into positive or negative categories based on the sentiment expressed in the text.

The input to the system consists of textual data, specifically movie reviews, which can be either positive or negative in sentiment. The output of the system is the classification of these reviews as either positive or negative. Since the system is trained using a dataset comprising 1000 negative reviews and 1000 positive reviews, it follows a supervised machine learning approach. This means that the system learns from labeled data during training and then applies this learning to classify new, unseen data during testing.

The task involves two main components: natural language processing (NLP) for understanding the sentiments expressed in the training data and machine learning techniques for predicting sentiments in the testing data. The system needs to leverage NLP to extract meaningful features from the text, such as sentiment-bearing words or phrases, and then use machine learning models, such as those available in NLTK, scikit-learn, or deep learning frameworks, to make accurate predictions about the sentiment of the input text.

One of the specific applications of sentiment analysis in this project is movie review classification. The system processes movie reviews, identifies the sentiments conveyed in each review (positive or negative), and then evaluates the accuracy of its classifications. Instead of outputting individual predictions for each review, the system calculates and presents the mean performance metrics, such as accuracy, precision, recall, and F1 score, for each machine learning model used in the sentiment analysis task.

3 Previous Work

In my exploration of this project, the paper "Sentiment Analysis of Movie Review Comments" by Yessenov, Kuat, and Sasa Misailovic from Massachusetts Institute of Technology, Spring 2009, provided a comprehensive exploration of sentiment analysis within the context of movie review comments. It delved into various aspects such as problem definition, where sentiments in textual data needed to be classified as positive or negative, and discussed techniques for feature extraction, focusing on identifying sentiment-bearing words and phrases. The paper also touched upon classifier implementation strategies, examining how machine learning algorithms can be trained and applied to categorize sentiments in textual data accurately.

On the practical implementation side, the online article "How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK)" by Shaumik Daityari from DigitalOcean was immensely helpful. It offered a step-by-step guide on leveraging the NLTK library in Python for sentiment analysis tasks. The article covered essential topics such as data preprocessing, feature extraction using techniques like bag-of-words and TF-IDF, and building and training sentiment analysis models using machine learning algorithms available in NLTK. Additionally, it provided insights into handling textual data, including cleaning and tokenization, which are crucial steps in preparing data for sentiment analysis tasks. Overall, these resources complemented each other well, providing theoretical insights and practical guidance for implementing sentiment analysis solutions effectively.

Drawing from my previous work, particularly in Assignment 3, I integrated k-fold cross-validation techniques into this sentiment analysis project, enhancing model evaluation and performance assessment. Furthermore, reflecting on a past project in CS 222's Software Design Lab, where my team and I developed a sentiment predictor trained on political tweets, I gained insights into the impact of implicit biases within datasets on data analysis outcomes. This experience underscored the importance of dataset selection and preprocessing in mitigating biases and ensuring robust sentiment analysis results.

4 Approach

For the baseline approach, I employed a Naive Bayes machine learning model, a popular choice for text classification tasks due to its simplicity and effectiveness. The dataset consisted of 1000 negative reviews and 1000 positive reviews of movies, serving as the training and testing data for the sentiment analysis task.

Regarding data preprocessing, I initially performed basic steps such as lower-casing all text and tokenizing words based on whitespace. This basic preprocessing aimed to standardize the text data and prepare it for feature extraction using a bag-of-words representation.

In terms of text representation, I utilized the bag-of-words model, which represents each document (movie review) as a vector of word frequencies. This representation disregards word order but captures important information about the presence of specific words in each review.

Moving on to feature sets, I experimented with several approaches to enhance the classifier's performance. Firstly, I incorporated decision tree and Bernoulli models alongside the Naive Bayes model to explore different algorithmic approaches. Additionally, I implemented more advanced preprocessing techniques in the improved classifier.

The improved preprocessing steps included:

1. Lemmatization using the NLTK Lemmatizer: This process reduces words to their base form (e.g., "running" to "run"), helping to standardize vocabulary and reduce feature dimensionality.

2. Removal of empty and punctuation-only tokens: These tokens didn't contribute meaningful information to sentiment analysis and were therefore removed.

3. Experimentation with stop word removal: While common in text preprocessing, I found that removing stop words didn't significantly improve performance, possibly due to the extensive list of English stop words potentially altering sentiment in reviews.

In addition to traditional preprocessing, I explored incorporating additional feature sets:

1. Part-of-speech (POS) tags: I experimented with using POS tags to capture syntactic information in the text, but this didn't yield substantial improvements in classification accuracy.

2. Emotion-based features using the EmoLex database: This involved associating words in the reviews with emotions (e.g., positive, negative, joy, fear) from the EmoLex database. I then removed words with emotions contrary to the review's main sentiment, which notably improved classification accuracy, especially when filtering out negative words from positive reviews.

To evaluate model performance, I employed k-fold cross-validation with k set to 10 for both the baseline and improved classifiers. This technique ensured robustness in model evaluation by training and testing the models on different subsets of the dataset multiple times.

Overall, this comprehensive approach involved iterative experimentation with various models, preprocessing techniques, and feature sets, leading to a refined sentiment analysis system with improved accuracy and robustness.

5 Results and Discussion

The evaluation results of the sentiment analysis classifiers, including Naive Bayes, Decision Tree, and Bernoulli models, reveal insights into their performance metrics such as accuracy and precision. The baseline Naive Bayes classifier achieved an accuracy of approximately 82.73% and a precision of about 85.83%. In contrast, the improved Naive Bayes classifier showed enhanced performance with an accuracy of around 85.52% and a precision of approximately 87.28%. Notably, the Decision Tree model exhibited an accuracy of approximately 75.76% and a precision of about 73.66%, while the Bernoulli model achieved an accuracy of around 83.67% but a lower precision of about 71.51%.

The consistent superiority of the Naive Bayes classifier across both baseline and improved versions underscores its effectiveness for this sentiment analysis task. This aligns with established knowledge in machine learning, where Naive Bayes is well-suited for binary classification tasks due to its simplicity and probabilistic nature. Even when incorporating more complex features like part-of-speech (POS) tags, Naive Bayes maintained its lead, showcasing its robustness in handling textual data with varied feature sets.

The Bernoulli model, a variant of Naive Bayes, performed relatively well but displayed lower precision compared to Naive Bayes. This could be attributed to the Bernoulli model's assumption of binary features, which may not fully capture the nuanced sentiment expressions present in the reviews. On the other hand, the Decision Tree model, while versatile, exhibited lower accuracy and precision, indicating challenges in accurately capturing the underlying patterns in the sentiment data compared to the probabilistic approach of Naive Bayes.

The feature representation used by each model varied slightly based on the preprocessing steps and feature sets employed. However, the core feature representation relied on a bag-of-words model with additional features such as lemmatization, token removal, and experimentation with POS tags and emotion-based features. The training time for each model was influenced by factors such as feature dimensionality, algorithm complexity, and dataset size. Generally, Naive Bayes models trained faster compared to decision tree models due to their probabilistic nature and simpler computations.

Although recall and F1-score metrics were not implemented due to technical issues, the analysis focused on standard metrics like precision and accuracy. These results reinforce the efficacy of Naive Bayes as the preferred model for sentiment analysis in this context, highlighting the importance of choosing an appropriate machine learning algorithm that aligns with the task's characteristics and dataset complexities.

6 Future Work and Conclusions

Engaging with this sentiment analysis project and delving into the intricacies of natural language processing (NLP) through lectures underscored the formidable challenges inherent in understanding textual data. Even within the scope of a binary classification task, the complexities introduced by linguistic nuances such as negation scope and sarcasm significantly elevate the difficulty level. It became evident that extracting meaningful insights from text goes beyond mere word counting; it necessitates a deep understanding of context, tone, and subtle linguistic cues. Given more time or enhanced proficiency in NLTK, addressing these challenges would be a natural progression to enhance classifier performance. Specifically, detecting sarcasm, a prevalent linguistic device in movie reviews, especially negative ones, presents a compelling yet challenging avenue for improvement.

Reflecting on the project's implementation, a notable area for refinement lies in the representation of text features. While utilizing dictionaries and numpy arrays served its purpose, transitioning to a pandas dataframe structure akin to Assignment 3 would offer enhanced convenience and flexibility. This shift would facilitate the implementation of incremental learning or leave-one-out strategies, allowing for more dynamic model adjustments and evaluation methods. Furthermore, incorporating more advanced feature engineering techniques, such as sentiment lexicons or semantic analysis, could deepen the model's understanding of sentiment nuances and improve its predictive capabilities.

The improved system, leveraging Naive Bayes with enhanced preprocessing and feature sets, demonstrated superior performance in accuracy and precision compared to other models. The decision to choose Naive Bayes was based on its consistent performance across varied feature sets and its suitability for binary classification tasks. While there wasn't a direct correlation between model performance and training time, faster training models like Naive Bayes tended to perform well due to their efficient computations and probabilistic nature. However, this relationship varied based on the complexity of features and the dataset's characteristics.

In conclusion, the project highlighted the importance of iterative experimentation with preprocessing techniques, feature sets, and model selection for robust sentiment analysis. Future work will focus on refining feature representations, incorporating advanced NLP techniques, and addressing nuanced linguistic phenomena to further enhance classifier performance.

References

[1] Yessenov, Kuat, and Sasa Misailovic. “Sentiment Analysis of Movie Review Comments” Massachusetts Institute of Technology, Spring 2009.

[2] Daityari, Shaumik. “How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK).” DigitalOcean, 28 Jan. 2021, www.digitalocean.com/community/tutorials/how-to-perform-sentimentanalysis-in-python-3-using-the-natural-language-toolkit-nltk.