Implementation and Evaluation of a Naive Bayes Classifier

Report Summary:

This report explores the implementation and evaluation of a Naive Bayes classifier for sentiment analysis on a movie review dataset. The data set was obtained from (https://github.com/dennybritz/cnn-text-classification-tf/tree/master/data/rt-polaritydata). The primary objectives were to split the data into training, development, and test sets, implement and train the classifier from scratch, evaluate its performance on the test set, analyze examples where the classifier was confident or uncertain, and identify the most useful features for each class.

Data Preprocessing:

To initiate the process, the movie review dataset was loaded, containing both negative and positive reviews. In order to prepare the data for classification, tokenization was applied, which involves breaking down the texts into smaller chunks, typically words in this context. Each review in the dataset was tokenized, splitting them into individual words or tokens. This step is crucial as it converts raw text into a format that can be processed by machine learning algorithms. Additionally, all words were converted into lowercase to standardize the text. This preprocessing step ensures uniformity in the representation of words. Through these preprocessing techniques, we transformed the raw textual data into a structured format suitable for subsequent analysis and modeling. These fundamental preprocessing steps serve as the initial groundwork for building a Naive Bayes classifier to classify movie reviews based on sentiment. Stopwords were also removed and by removing stopwords, the dimensionality of the feature space was reduced, and focus put on words that are more indicative of sentiment.

Vocabulary Building:

A vocabulary of 3000 words was built from the training data to represent the most frequent words in the dataset. This vocabulary serves as the feature space for the Naive Bayes classifier, enabling it to learn from the presence or absence of these words in the reviews.

Data Splitting:

The dataset was split into training (70%), development (15%), and test (15%) sets. This ensures that the classifier is trained on a sufficient amount of data, tuned on the development set, and evaluated on a separate test set to evaluate its generalization performance. Splitting the data into three sets helps prevent overfitting and provides a more accurate assessment of the model's performance.

Naive Bayes Classifier Implementation:

The Naive Bayes classifier was implemented from scratch, adhering to the principles of the algorithm. Features were extracted based on word presence, likelihoods and prior probabilities were computed, and predictions were made using Bayes' theorem. Debugging was facilitated using examples from lecture slides. Implementing the classifier from scratch provides a deeper understanding of its inner workings and allows for more flexibility in customization.

Model Training and Evaluation:

The classifier was trained on the training data and tuned on the development set by adjusting hyperparameters. Classifier performance was evaluated on the development set to assess its accuracy and fine-tune parameters for optimal performance. Training and evaluating the model on separate datasets allow us to validate its performance and make necessary adjustments to improve its accuracy.

Test Set Evaluation:

The best model, trained on the concatenated training and development sets, was evaluated on the test set to provide a final assessment of its performance. This step ensures that the model's effectiveness is validated on unseen data. Evaluating the model on a separate test set helps ensure that it generalizes well to new, unseen data and provides a more reliable estimate of its performance in real-world scenarios. The accuracy achieved on the development set was 73.97%. This metric indicates how well the classifier performs on a separate subset of data used for tuning hyperparameters and assessing model performance during development. The accuracy obtained on the test set was 76.03%. This metric reflects the classifier's performance on unseen data, providing an estimate of its generalization ability in real-world scenarios.

Confident and Uncertain Examples Analysis:

Examples from the test set where the classifier exhibited high confidence or uncertainty in its predictions were analyzed. Insights were drawn to understand the reasons behind the classifier's confidence or uncertainty, shedding light on its behavior and performance. Analyzing confident and uncertain examples provides valuable insights into the strengths and weaknesses of the classifier and can help identify areas for improvement. For instance, reviews such as "warm water under a red bridge is a quirky and poignant japanese film that explores the fascinating connections between women, water, nature, and sexuality." and " As the movie traces Mr. Brown's athletic exploits, it is impossible not to be awed by the power and grace of one of the greatest natural sportsmen of modern times." were confidently classified as positive with high confidence scores because they contain strong presence of positive words like quicky, awed, grace and poignant and these words leave no doubt about the positive sentiment. In contrast, there were instances where the classifier showed uncertainty in its predictions. Reviews such as "my oh my, is this an invigorating, electric movie." and "skip work to see it at the first opportunity." received lower confidence scores, indicating uncertainty in sentiment classification. Phrases like “my oh my” and “skip work” introduce ambiguity.

Most Useful Features Identification:

By analyzing the most informative features learned by the classifier, we gained insights into the key factors influencing sentiment classification. These features provide valuable information about the discriminative power of words in determining the sentiment of movie reviews. Identifying the most useful features helps understand which words contribute the most to the classification decision and can provide insights into the sentiment of the reviews. Features such as "riveting," "wonderfully", "gem”, “heartwarming" for the positive class and "unfunny", "mediocre", "poorly", "pointless", "badly" were highly indicative of sentiment polarity, with significant weights in distinguishing between positive and negative reviews.

Conclusion:

Overall, the implementation and evaluation of the Naive Bayes classifier yielded valuable insights into sentiment analysis on movie reviews. Through careful preprocessing, model training, and evaluation, we were able to develop an effective classifier and analyze its performance comprehensively. Further experimentation and analysis could lead to enhancements in classifier performance and provide deeper insights into sentiment analysis tasks.

Future Work:

Future work could involve exploring advanced techniques such as lemmatization, feature selection, and model optimization to further improve classifier performance. Additionally, exploring other machine learning algorithms and ensembling techniques could provide further insights and potentially enhance classification accuracy.