HW 2 - Report
CSCI 544 - Applied NLP

Nav Sanya Anand
6878418392

The provided code orchestrates a multi-step process for sentiment analysis using Amazon review data on office products. It begins with data reading and filtering, focusing on the review bodies and star ratings. To ensure balanced representation across different ratings, a balanced dataset is crafted by sampling equal numbers of reviews for each star rating category. Notably, the script employs ternary labels to classify sentiments as positive, negative, or neutral based on star ratings, with neutral reviews being assigned a distinct label. The data preprocessing stage involves extensive cleaning, including HTML tag removal, contraction expansion, special character removal, and lemmatization. Additionally, neutral reviews are excluded in the binary label creation process, resulting in a refined dataset for binary sentiment classification.

In the subsequent sections, the script proceeds to split the datasets into training and testing sets for both ternary and binary sentiment analysis. Leveraging the Word2Vec embeddings, it explores semantic relationships between words by calculating similarities between word vectors. Initially, pretrained Word2Vec embeddings are employed to demonstrate semantic relationships like gendered analogies and lexical similarities. Subsequently, custom Word2Vec models are trained using the tokenized reviews, enabling the exploration of semantic relationships tailored to the dataset. These steps collectively form a robust pipeline for sentiment analysis, encompassing data preprocessing, label generation, dataset splitting, and word embedding exploration, facilitating a comprehensive understanding of the sentiment expressed in the Amazon reviews.

Continuing the analysis, the script implements various machine learning models trained on different feature representations, including TF-IDF, pretrained Word2Vec embeddings, and custom-trained Word2Vec embeddings. For TF-IDF, the script employs a Perceptron classifier and a Linear Support Vector Machine (SVM) classifier, achieving notable accuracy scores on the binary sentiment classification task. Leveraging pretrained Word2Vec embeddings, the Perceptron and SVM classifiers attain competitive accuracies, showcasing the effectiveness of utilizing pre-existing semantic embeddings for sentiment analysis. Furthermore, the script explores the performance of models trained on custom-trained Word2Vec embeddings, where both the Perceptron and SVM classifiers exhibit commendable accuracy rates, demonstrating the efficacy of capturing semantic relationships within the domain-specific context of the Amazon office product reviews. Overall, the script offers a comprehensive evaluation of sentiment analysis models across diverse feature representations, shedding light on the relative strengths and limitations of each approach in discerning sentiment from textual data.

The script extends its analysis to include feedforward neural networks (FFNNs) for sentiment classification, employing both binary and tertiary sentiment labels. Utilizing the average method for feature representation, the FFNNs are trained on Word2Vec embeddings, both custom-trained and pretrained, to capture semantic nuances within the textual data. The binary sentiment classification tasks demonstrate the adaptability of FFNNs, achieving competitive accuracies across different feature

representations. By converting Word2Vec features into PyTorch tensors and implementing a multilayer perceptron architecture, the models are trained iteratively over epochs, optimizing their weights and biases to minimize cross-entropy loss. The evaluation phase showcases the efficacy of FFNNs in discerning sentiment, as evidenced by the achieved accuracies on the respective test sets, affirming the utility of neural networks in sentiment analysis tasks.

Moving beyond binary classification, the script extends its analysis to tertiary sentiment classification, accommodating nuanced sentiment categorization across positive, negative, and neutral sentiments. With a similar training and evaluation setup, the FFNNs adapt to the complexity of the ternary sentiment classification task, leveraging Word2Vec embeddings to capture the semantic richness of the reviews. Through the iterative training process, the FFNNs optimize their parameters to effectively classify sentiments, showcasing the versatility of neural network architectures in handling multi-class sentiment analysis tasks. The achieved accuracies on the test sets underscore the robustness of the FFNNs in discerning nuanced sentiment expressions, underscoring their potential in real-world sentiment analysis applications across diverse domains and datasets.

The script extends its analysis to incorporate the concatenation method for feature representation, particularly focusing on Word2Vec embeddings and pretrained embeddings. By concatenating embeddings of the first 10 words of each review, the model encapsulates a more comprehensive context for sentiment analysis. The utilization of this method facilitates the creation of feature vectors that capture richer semantic information, enhancing the discriminative power of the model. Through the concatenation approach, the script generates new feature sets for both binary and tertiary sentiment classification tasks, enabling the subsequent training and evaluation of feedforward neural networks (FFNNs) to discern sentiment nuances within the text data.

With the concatenation method employed, the FFNNs are trained iteratively over epochs to optimize their parameters and minimize cross-entropy loss. Leveraging PyTorch tensors and the multilayer perceptron architecture, the models adapt to the complexities of binary and tertiary sentiment classification tasks. By integrating Word2Vec embeddings and pretrained embeddings, the FFNNs showcase their adaptability in discerning sentiment expressions across diverse datasets. Through the evaluation phase, the accuracies achieved on the test sets underscore the efficacy of the concatenation method and the robustness of FFNNs in capturing nuanced sentiment expressions. This analysis underscores the potential of FFNNs and advanced feature representation techniques in advancing sentiment analysis tasks across various domains and datasets.

The script extends its analysis to include Convolutional Neural Networks (CNNs) for sentiment analysis tasks. CNNs are well-suited for processing sequential data like text due to their ability to capture local patterns and hierarchical representations. In this context, the CNN architecture is adapted to handle Word2Vec embeddings and pretrained embeddings for both binary and tertiary sentiment classification tasks. By leveraging PyTorch's capabilities, the script constructs a CNN model with customizable parameters such as the number of hidden nodes, dropout rate, and output classes.

During training, the script iterates over the dataset in mini-batches, computing the loss, and optimizing the model parameters using stochastic gradient descent (SGD). The loss is minimized over multiple epochs,

with the model gradually learning to distinguish between different sentiment classes. Through this iterative process, the CNNs adapt their weights to extract meaningful features from the input embeddings and improve classification accuracy. The evaluation phase assesses the model's performance on the test set, providing insights into its generalization capabilities and effectiveness in discerning sentiment expressions across various datasets. Overall, the incorporation of CNNs enriches the sentiment analysis pipeline, demonstrating the versatility of deep learning architectures in natural language understanding tasks.

Compiled Results

S.No.	**WORD2VEC TYPE**	**FUNCTION**	**ACCURACY**
1.	TF-IDF	PERCEPTRON	0.8164
2.	TF-IDF	SVM	0.8651
3.	PRE-TRAINED	PERCEPTRON	0.7555
4.	PRE-TRAINED	SVM	0.8313
5.	TRAINED	PERCEPTRON	0.7956
6.	TRAINED	SVM	0.8656
7.	PRE-TRAINED	FNN[AVG] - BINARY	0.7925
8.	PRE-TRAINED	FNN[AVG] - TERTIARY	0.6156
9.	TRAINED	FNN[AVG] - BINARY	0.8529
10.	TRAINED	FNN[AVG] - TERTIARY	0.6788
11.	PRE-TRAINED	FNN[CONCAT] - BINARY	0.7259
12.	PRE-TRAINED	FNN[CONCAT] - TERTIARY	0.5578
13.	TRAINED	FNN[CONCAT] - BINARY	0.7631
14.	TRAINED	FNN[CONCAT] - TERTIARY	0.6111
15.	PRE-TRAINED	CNN - BINARY	0.8424
16.	PRE-TRAINED	CNN - TERTIARY	0.8334
17.	TRAINED	CNN - BINARY	0.8816
18.	TRAINED	CNN - TERTIARY	0.8243