Sentiment Analysis Assignment Report

Introduction

Sentiment analysis is a crucial task in natural language processing (NLP) that involves determining the sentiment expressed in a piece of text, such as positive, negative, or neutral. In this assignment, we developed a text classification system for sentiment analysis using two different models: a traditional machine learning model (Logistic Regression) and a deep learning model (LSTM). We used the IMDB movie reviews dataset, which contains 50,000 reviews labeled as positive or negative.

Dataset Selection and Exploratory Data Analysis (EDA)

We selected the IMDB movie reviews dataset, which is widely used for sentiment analysis tasks. The dataset contains 50,000 reviews, evenly split between positive and negative sentiments.

Dataset Overview

- Number of Reviews: 50,000
- Sentiment Distribution: 25,000 positive and 25,000 negative reviews

Exploratory Data Analysis

We conducted an initial exploratory data analysis to understand the dataset better. This included:

- Class Distribution: We visualized the distribution of positive and negative reviews using a count plot, confirming that the dataset is balanced.
- Review Length: We analyzed the length of reviews by calculating the number of words in each review and plotted the distribution of review lengths for both positive and negative sentiments. This helped us understand the typical length of reviews and identify any outliers.

Preprocessing

Preprocessing is a critical step in preparing the text data for model training. We performed the following preprocessing steps:

- Handling Missing Values: The dataset did not contain any missing values, so no action was required.
- Tokenization: We tokenized the text into individual words.
- Removing Stopwords: We removed common stopwords to reduce noise in the data.
- Word Embeddings: We used Word2Vec embeddings to convert words into dense vectors, capturing semantic relationships between words.

Justification for Preprocessing Choices

- Tokenization: Breaking text into words is essential for further processing.
- Stopwords Removal: Removing stopwords helps in reducing the dimensionality of the data and focusing on meaningful words.
- Word2Vec Embeddings: Word2Vec captures the context of words in a continuous vector space, which is beneficial for both traditional and deep learning models.

Model Design and Implementation

We implemented two models for sentiment classification: Logistic Regression and LSTM.

Logistic Regression

Logistic Regression is a traditional machine learning model that is simple yet effective for binary classification tasks. We used the TF-IDF vectorizer to convert text into numerical features before training the model.

LSTM (Long Short-Term Memory)

LSTM is a type of recurrent neural network (RNN) that is well-suited for sequence data like text. We used Word2Vec embeddings as input to the LSTM model.

Training and Hyperparameter Tuning

We trained both models and fine-tuned hyperparameters to optimize performance. For Logistic Regression, we experimented with different regularization strengths. For LSTM, we varied the learning rate, batch size, and number of epochs.

Experiment Tables

Logistic Regression

Regularization Strength	***Accuracy***	***Precision***	***Recall***	***F1-Score***
0.01	0.85	0.86	0.84	0.85
0.1	0.86	0.87	0.85	0.86
1.0	0.85	0.86	0.84	0.85

LSTM

Learning Rate	***Batch Size***	***Epochs***	***Accuracy***	***Precision***	***Recall***	***F1-Score***
0.001	32	10	0.88	0.89	0.87	0.88
0.0001	64	20	0.89	0.90	0.88	0.89
0.00001	32	20	0.90	0.91	0.89	0.90

Model Evaluation

We evaluated the models using accuracy, precision, recall, and F1-score. The confusion matrix was also used to visualize the performance.

Performance Metrics

- Logistic Regression: Achieved an accuracy of 86% with a regularization strength of 0.1.
- LSTM: Achieved an accuracy of 90% with a learning rate of 0.0001, batch size of 32, and 20 epochs.

Confusion Matrix

- The confusion matrix for both models showed that the LSTM model had fewer misclassifications compared to Logistic Regression.

Discussion and Key Findings

- The LSTM model outperformed Logistic Regression, achieving higher accuracy, precision, recall, and F1-score. This is likely due to LSTM's ability to capture sequential dependencies in text data.
- Fine-tuning hyperparameters, especially for LSTM, significantly improved model performance.
- The use of Word2Vec embeddings contributed to the LSTM's superior performance by capturing semantic relationships between words.

Conclusion

In this assignment, we developed and compared two sentiment analysis models: Logistic Regression and LSTM. The LSTM model demonstrated better performance, highlighting the advantages of deep learning models for text classification tasks. Future work could explore other deep learning architectures, such as GRU or BERT, and further optimize hyperparameters for even better performance.

Team Contributions

Team Member	***Role***
Prince Ndanyuzwe	Conducted EDA and implemented the Logistic Regression model.
Cynthia Nekesa	Preprocessed the data and implemented the LSTM model.
Smart Israel	Evaluated the models and wrote the report.

Code Documentation

The code for this project is available on GitHub. The repository includes a README file with detailed instructions on how to execute the code together with the colab notebook.

GitHub Repository Link: https://github.com/NdanyuzweP/Sentiment-Analysis-Assignment

Colab Notebook Link:
https://colab.research.google.com/drive/1TieizSNC46iVaucxP_I7evXkNBDm9UCY#scrollTo=gGOVfub97sQj

References

How can you evaluate sentiment analysis model performance? (2023, October 31). LinkedIn. https://www.linkedin.com/advice/1/how-can-you-evaluate-sentiment-analysis-model-ygfec

IMDB dataset of 50K movie reviews. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Pratibha Awasthi. (2023). Cleaning-and-Preparing-IMDB-Data. GitHub. https://github.com/PratibhaAwasthi/Cleaning-and-Preparing-IMDB-Data

Prince, & Cynthia. (2025, February 12). Google Colab. Retrieved February 12, 2025, from https://colab.research.google.com/drive/1TieizSNC46iVaucxP_I7evXkNBDm9UCY#scrollTo=gGOVfub97sQj

WiseCoder. (2024, July 10). IMDB Dataset of 50K Movie Reviews | Kaggle Dataset. YouTube. https://www.youtube.com/watch?v=PZXI_jIMJ8Y&ab_channel=WiseCoder