Text Classification Project Report

Overview

This project focuses on classifying text data into five distinct categories using various text embedding techniques and machine learning models. The dataset contains text samples and corresponding labels. The project encompasses data exploration, preprocessing, and evaluation of three different text embedding methods: Word2Vec, GloVe, and BERT. Each method's performance is assessed using Logistic Regression.

Introduction

The goal of this project is to build and evaluate text classification models using different embedding techniques. The methods evaluated include:

- **Word2Vec**: A word embedding model that represents words as vectors in a continuous vector space.
- **GloVe**: A pre-trained word embedding model that captures global statistical information.
- **BERT**: A contextualized word embedding model that provides embeddings based on the context of the words.

Dataset Source

The dataset used for this project can be accessed https://www.kaggle.com/datasets/sunilthite/text-document-classification-dataset/data.

Data Exploration

The dataset includes:

- **Text**: The content of the text sample.
- **Label**: The category of the text, with labels as follows:
    - 0: Politics
    - 1: Sport
    - 2: Technology
    - 3: Entertainment

- 4: Business

Key steps in data exploration included:

- **Data Overview**: Checking the structure and summary statistics.
- **Label Distribution**: Visualizing how the categories are distributed.
- **Text Characteristics**: Analyzing text lengths and word frequencies.

Text Preprocessing

Text preprocessing involved:

- **Text Cleaning**: Removing mentions, URLs, and non-alphanumeric characters.
- **Feature Engineering**: Calculating word counts and text lengths.

Text Embedding Techniques

Word2Vec

- **Training**: A Word2Vec model was trained on the dataset.
- **Vector Conversion**: Text samples were converted into average Word2Vec vectors.

GloVe

- **Pre-trained Model**: A pre-trained GloVe model was utilized.
- **Vector Conversion**: Text samples were converted into average GloVe vectors.

BERT

- **Pre-trained Model**: A pre-trained BERT model was used to obtain contextualized embeddings for each text sample.

Model Training and Evaluation

Logistic Regression models were trained for each embedding technique. The models were evaluated using:

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**

Results

The performance of each embedding technique is summarized below:

Model	Accuracy	Precision	Recall	F1 Score
Word2Vec	88.03%	88.06%	88.03%	87.92%
GloVe	95.77%	95.87%	95.77%	95.80%
BERT	98.36%	98.40%	98.36%	98.36%

Usage

1. **Install Dependencies**:

bash

Copy code

pip install numpy pandas plotly seaborn matplotlib gensim scikit-learn transformers torch

2. **Prepare Dataset**:

Ensure **data.csv** is located in the same directory as your script.

3. **Run the Script**:

Execute the script to perform data exploration, preprocessing, model training, and evaluation.

bash

Copy code

python script.py

Dependencies

- **Python 3.x**
- **Libraries**:
    - **numpy**
    - **pandas**
    - **plotly**

- **seaborn**
- **matplotlib**
- **gensim**
- **scikit-learn**
- **transformers**
- **torch**

Conclusion

This project successfully evaluated different text embedding techniques and their impact on text classification performance. BERT provided the highest accuracy and overall performance, showing its effectiveness in understanding and classifying text with nuanced meanings. GloVe offered a good balance between performance and efficiency, while Word2Vec served as a solid baseline. Future work could focus on model fine-tuning, exploring other advanced models, and adapting approaches to specific domains.