Q2: Natural Language Processing Project

Documenting the Process

1. Introduction:

- In this project, we aim to enhance text classification accuracy using various text embedding techniques. Text embedding techniques are powerful tools for converting text into numerical representations that machines can understand, facilitating the analysis and classification of text. This project focuses on comparing the performance of Word2Vec, GloVe, and BERT in sentiment analysis using a Twitter dataset.

2. Methodology:

- **Data Used:** We utilized a dataset containing 32,567 tweets classified by sentiment (negative, neutral, positive). The data was divided into 29,603 tweets for training and 2,964 for evaluation. This dataset was chosen because it encompasses a diverse array of texts that reflect various sentiments, making it ideal for testing the effectiveness of text embedding techniques.

3. Embedding Techniques:

- **Word2Vec**: A pre-trained Word2Vec model was used to convert texts into vectors. Word2Vec relies on training a simple neural network to learn word representations based on local context. The **"genism"** library was used to load and apply the model.
- **GloVe:** The GloVe model, which relies on global statistics from texts to represent words, was employed. GloVe provides strong representations of semantic relationships between words. The model was loaded from text files containing the vectors.
- **BERT:** The BERT model was used, which considers the full context of a sentence, allowing it to handle word polysemy effectively. The transformers library was utilized to load and apply the model.

Data Preparation: The data was prepared for each model through appropriate encoding and splitting. For BERT, the model-specific encoding was applied to ensure data compatibility with the model's requirements. BertTokenizer was used to encode the texts and convert them into input vectors.

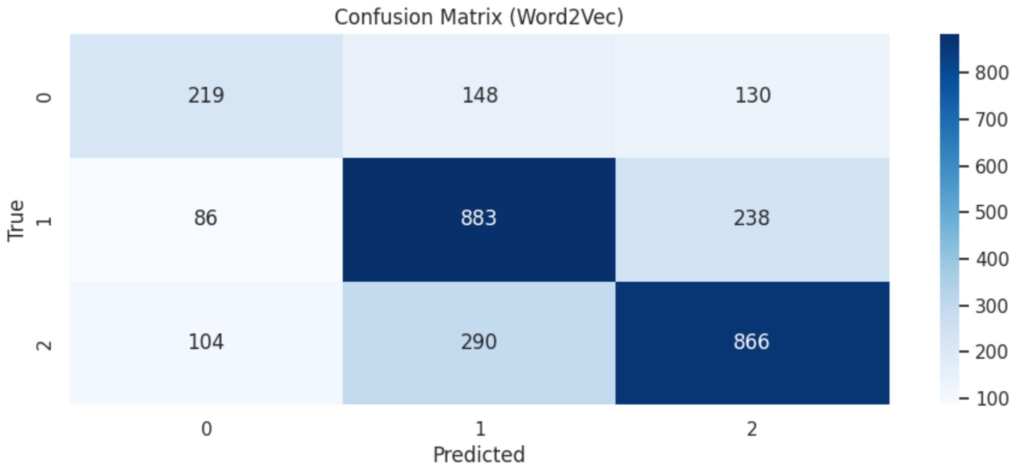
Results:

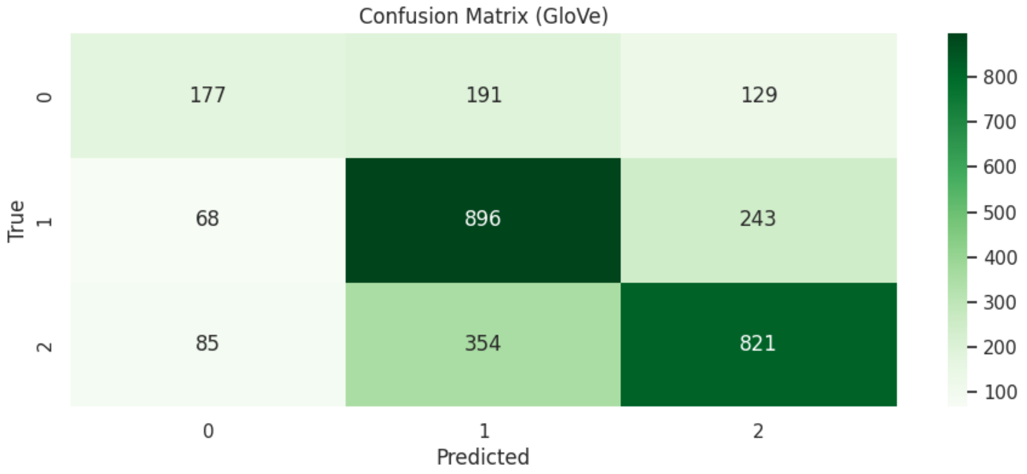
Model Performance:

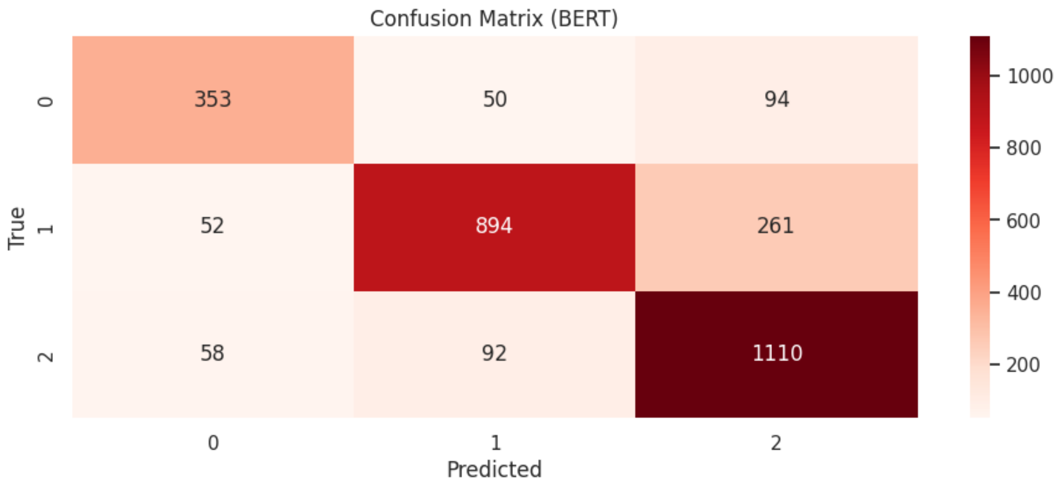
- **Word2Vec:** Achieved an accuracy of 66%, performing well in neutral and positive categories but underperforming in the negative category. Logistic regression was used to classify the resulting vectors.
- **GloVe:** Achieved an accuracy of 64%, demonstrating outstanding performance in the neutral category, indicating its effectiveness in representing semantic relationships. The same logistic regression model was employed to evaluate the performance.
- **BERT:** Achieved an accuracy of 80%, making it the best-performing model in this comparison. Despite its slow training speed and high computational resource requirements, BERT's superior contextual understanding capabilities contributed to its high accuracy. The model was trained and evaluated, confirming its effectiveness in handling complex text analysis.

Graphs and Tables:

Graphs illustrating the performance of different models were included, facilitating visual comparison of results. Confusion matrices can be used to clarify how models classify different categories.







Analysis:

Strengths and Weaknesses:

- **Word2Vec:**
  - **Strengths:** Fast and efficient in semantic relationships, suitable for applications requiring rapid processing**.**
  - **Weaknesses:** Limited in handling word polysemy as it relies solely on local context.
- **GloVe:**
  - **Strengths:** Based on global statistics, helping capture semantic relationships more effectively**.**
  - **Weaknesses:** Requires intensive preprocessing and may not perform well in handling word polysemy.
- **BERT:**
  - **Strengths:** Considers the full context of a sentence, making it capable of addressing word polysemy**.**
  - **Weaknesses:** Slow in training and application, requiring significant computational resources.

Performance Comparison: Based on the results, BERT may be preferred in applications requiring high accuracy in contextual understanding, while Word2Vec or GloVe can be used in applications needing faster processing.

Conclusion

Recommendations: We recommend using BERT in applications requiring high accuracy in contextual understanding, such as complex text analysis, while Word2Vec or GloVe can be utilized in applications requiring faster processing and fewer resources.

Final Conclusion: The results demonstrated that each embedding technique has its advantages and disadvantages, and the appropriate technique should be selected based on the specific application requirements.