Analysis of Pretrained Word Embedding Models for Detecting AI-Generated Text

Moroti Sonde(R11904541), Poojitha Kinthada(R11902142), Chandrika KanchetiR11908347
Sruthi Mandalapu(R11906160), Charishma Rathan Bala(R11908154), Gowtham Edumudi(R11912904)

Abstract - This research explores the potential of Natural Language Processing (NLP) techniques for detecting AI-generated text in the educational sector, particularly essays. We investigate the effectiveness of two pre-trained word embedding models, Word2Vec and GloVe, in conjunction with a Bi-LSTM model. The research uses a dataset of 28,827 essays, categorized as human-written or AI-generated. Text pre-processing involves stop-word and special character removal, followed by a train-test split. The pre-trained word embeddings convert textual data into vectors, fed into the Bi-LSTM model for training. Evaluation metrics like accuracy and precision assess the model's performance in classifying AI-generated and human-generated essays. A comparative analysis of Word2Vec and GloVe embeddings was conducted to determine their suitability for this task. At last, we discuss real-world implications and explore methods for reducing false positives in AI-generated text detection

Keywords - Data Preprocessing, Word Embeddings, Word2Vec, GloVe, Bi-LSTM Model

I Introduction

In recent years, the rapid integration of artificial intelligence (AI) into various aspects of life has led to great advancements as well as new challenges, particularly in the education sector. One of such challenge is the use of AI-generated text, which is becoming so advanced that it's hard to tell apart from text written by humans [1]. This has great implications for academic integrity, as AI tools become more accessible for tasks such as essay writing. To address this issue, our project focuses on the utilization of Natural Language Processing (NLP) techniques and deep learning models to detect AI-generated text effectively.

As of 2022, AI applications like the Generative Pre-Training Transformer (GPT-3) have become the focal point, producing text that mirrors human creativity and is often indistinguishable to the average person [2]. These applications are part of generative AI, which creates new content by generating multiple outputs and selecting the best match for the desired traits.OpenAI's ChatGPT and DALL-E are two popular generative AI tools that have had a great impact on popular opinions about AI [3]. These technologies highlight AI's rising significance in a variety of industries, including education, government, and entertainment. Despite their benefits, concerns about their misuse remain, such as in copying creative works, and helping with written assignment, showing the double-edged nature of AI technology advancement.

This project focuses on using Bidirectional Long Short-Term Memory (Bi-LSTM) models, which are effective at understanding sequences of data from both directions. We also studied how well two popular pretrained word embeddings, Word2Vec and Glove can help produce good result when trained with Bi-LSTM model. These embeddings transform text into numerical vector representations, capturing latent semantic features that are crucial for distinguishing between human and AI-authored text.

This research aims not only to evaluate the performance of these embedding models in identifying the nature of the text but also to explore the broader implications of AI in educational settings. By employing metrics such as accuracy and precision, we will quantitatively assess our model's success and discuss potential strategies to enhance its ability to reduce false positives, thereby contributing to maintaining the integrity of educational assessments.

1.1 Motivation

As AI technology continues to advance, there's growing concern about the authenticity of written content, with AI-generated texts becoming increasingly indistinguishable from human-written text. This issue poses significant challenges for educators and institutions dedicated to maintaining academic honesty and integrity.

This research seeks to address these challenges by using Natural Language Processing (NLP) techniques and Bidirectional Long Short-Term Memory (Bi-LSTM) models. The goal is to develop a system capable of identifying AI-generated text. By using pre-trained word embeddings such as Word2Vec and GloVe, this project explores their effectiveness in capturing the semantic meaning of text data.

1.2 Objectives

The objectives for this project are:

1. To assess the ability of Word2Vec and GloVe models to differentiate between human-written and AI-generated text in a classification task.
2. To develop a Bi-LSTM model using Word2Vec or GloVe embeddings for accurately identifying AI-generated text.

II Expermiental Setup

We implemented our approach using Python 3.10. To convert the text into vector representations, we used both the Glove and Word2vec embeddings. The embedding dimension used is the vocabulary size with input length of 880. The Bi-LSTM wrapper consists of an LSTM layer with 200units of neurons, a dropout rate of 0.5, and a recurrent dropout of 0.3. Lastly, the output layer consists of a sigmoid activation function.

For our experimental work, we used three different learning rates of 0.01, 0.001, and 0.0001 combined with each word embedding. All models are optimized using Adam optimizer and a loss function of binary cross-entropy, the training for each combination was done using a batch size of 64, early stopping of patience 5 and 30 epochs.

2.1 Implementation

Data Collection & Preprocessing: This work uses two open-source datasets publicly available on Kaggle [4]. In this work, the dataset consist of 28828 texts, 14414 AI-generated and 14414 human written text. Along with the AI model and the prompt used in generating the text. The initial step proceeds with data for text analysis tasks, likely in preparation for tasks like machine learning or natural language processing (NLP). The Natural Language Toolkit (NLTK), a popular library for NLP tasks in Python that offers a wide range of functionalities for text processing, including tokenization, stemming, lemmatization, and removal of special characters and digits was used in cleaning the data. During data pre-processing, typically content is employed with a breakdown of text data into smaller units like words (tokens). This is often done using a tokenizer like the Tokenizer class from keras.preprocessing.text. The dataset was then splitted into 70%,15%, and 15% for training, test and validation set respectively.

Word Embeddings: These are techniques used in NLP where words or phrases are mapped to vectors of real numbers. They help to capture semantic meanings, syntactic similarity, and relation with other words. After Data preprocessing the next step follows up with pre-trained word embeddings from GloVe/Word2Vec to represent words in the essay dataset. These models come with pre-trained word vectors for a vast vocabulary. Each word in the model's vocabulary has a corresponding vector representation. Word2vec serves as a bridge connecting words and real numbers. It converts words into unique numerical vectors. The key lies in how these vectors represent the associations between words. Words that share meanings or frequently co-occur have similar vectors in this numerical space. This enables the performance of multiple tasks based on word meanings and contexts. You can identify words with related meanings, compare the overall meanings of documents, or assist by recognizing word connections. In general, Word2vec enables the exploration of word relationships by converting them into a numerical environment where analysis and manipulation become feasible. Similar to Word2Vec, GloVe captures semantic relationships between words by analyzing their co-occurrence statistics in a large text corpus and each word in the data will be converted into a corresponding vector based on the GloVe model.

Bi-LSTM Model: This is a term used for a sequence model which contains two LSTMs. One for processing input tokens in the forward direction and the other for processing in the reverse direction.Then returns a probability vector as final output.

2.2 Architecture

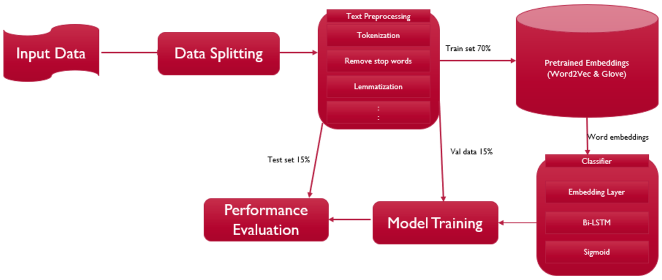

Figure 1: Architecture of the model

The process begins with text preprocessing, which includes cleaning the text data. This initial step typically eliminates stop words (common words such as "the" or "and") and special characters like punctuation marks. Following text preprocessing, the text data is segregated into three distinct sets: training data, validation data, and test data. The training data is utilized to train the machine learning model, while the validation data is employed to refine the model during training in order to prevent overfitting. The testing data is utilized to assess the model's performance on unseen data. The text data is subsequently converted into numerical representations using pretrained word embedding models like Word2vec or GloVe. This phase effectively transforms the text data into a format that the machine learning model can comprehend. Throughout the training process, the Bi-LSTM classifier learns to recognize patterns that differentiate human-authored essays from AI-generated essays based on the features extracted from the word vectors. The training procedure encompasses adjusting the weights of the Bi-LSTM's internal connections to enhance its classification accuracy. Once trained, the model is evaluated using the testing data. The model's performance is quantified using metrics such as accuracy (the percentage of accurately classified essays) and precision (the percentage of correctly identified AI-generated essays).

III Results

Figure 2 shows the accuracy of our glove embedding with Bi-LSTM model. We can observe that the model achieves the best accuracy of 99% when the learning rate is 0.001 while the precision is the same 98% for both 0.001 and 0.0001 learning rates. As seen in the confusion matrix shown in Figure 3, the model was able to correctly predict 2118 and 2143 text written by humans and AI respectively with false positives of 40 and 24. Furthermore, we can see from the accuracy and loss graph shown in figure 4,5, and 6 that the smaller the learning rate, the longer epochs it took to converge before early stopping is activated to prevent overfitting.

Figure 7 represents the accuracy and precision obtained using different learning rates for the word2vec model. As seen, the model trained with 0.01 and 0.0001 learning rate achieved the same performance of 98% for both accuracy and precision.

When compared to the glove embedding model(lr=0.0001), there are more false positives recorded in both the human and AI-generated text as seen in the confusion matrix of figure 8. Figure 9, 10, and 11 plots the visualization of the training and loss with respect to the number of epochs. Similar to glove embedding visualization, the smaller the learning rate, the longer it takes to converge before early stopping is activated. From figure 11, we can see that the training took over 27 epochs while others were less than that. This supports our initial findings that the smaller the learning rate, the longer it takes to converge before overfitting.

IV Discussion and Limitations

Across all the experiments, we can see that although the glove embedding with a learning rate of 0.001 outperforms others, they all have very close scores with little or no significant difference. Also, both Word2vec and Glove showed promising results in identifying AI-generated text which is crucial for academic integrity and honesty.

Bi-LSTMs, although powerful but are complex and can be difficult to interpret which may limit their practical use without substantial expert knowledge. Also, the model might not perform well enough when exposed to text written by children as it was not part of the training data hence might lead to lots of false positives.

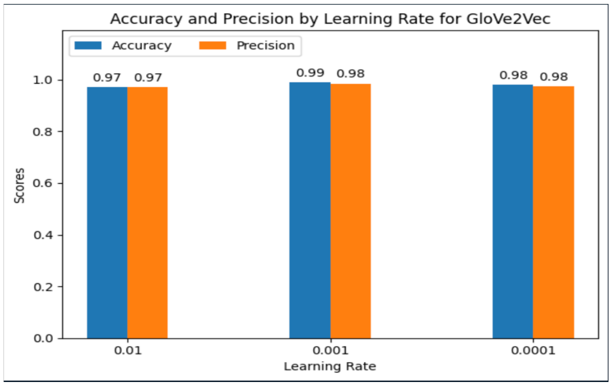

Figure 2: Glove Accuracy and Precision by Learning Rate

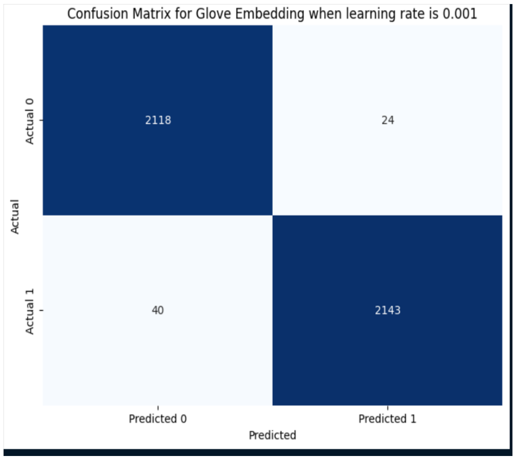

Figure 3: Glove Confusion matrix with learning rate 0.001

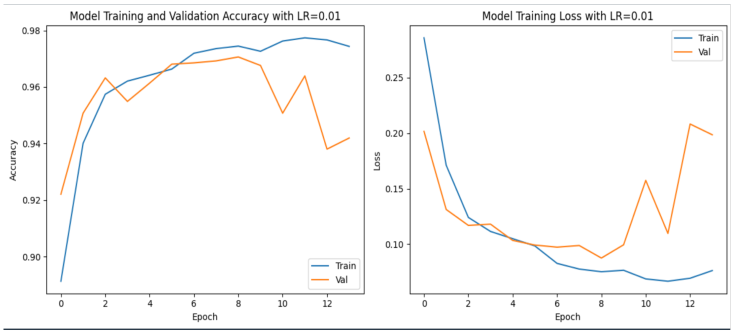

Figure 4: Glove training and validation accuracy + lr=0.01

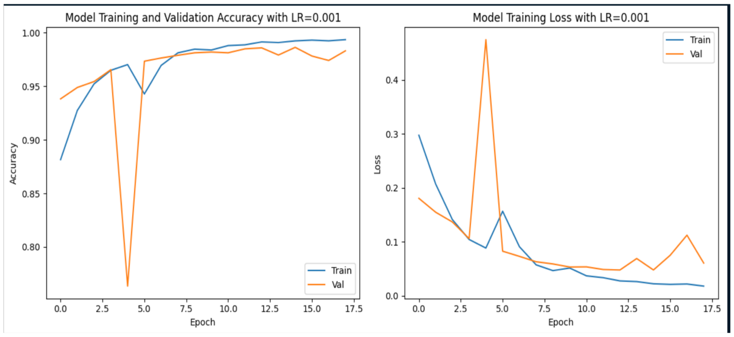

Figure 5: Glove training and validation accuracy + lr=0.001

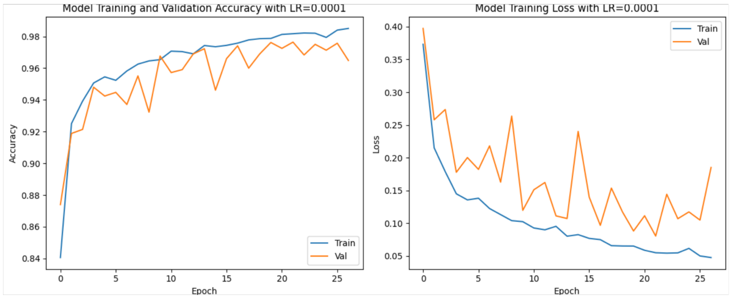

Figure 6: Glove training and validation accuracy + lr=0.0001

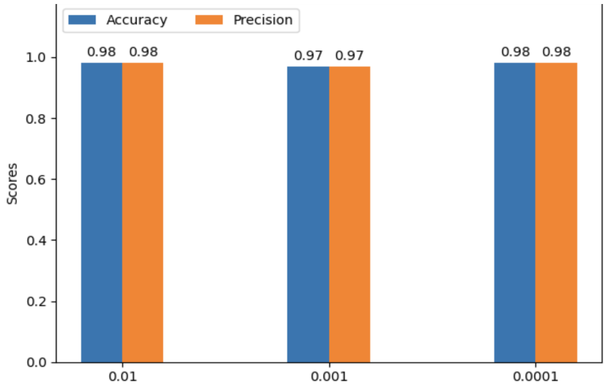

Figure 7: Word2Vec Accuracy and Precision by Learning Rate

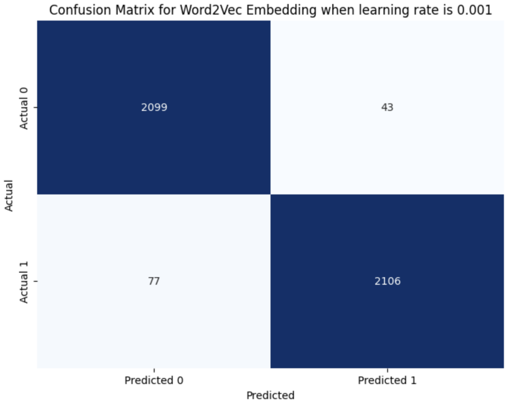

Figure 8: Word2Vec Confusion Matrix with learning rate 0.001

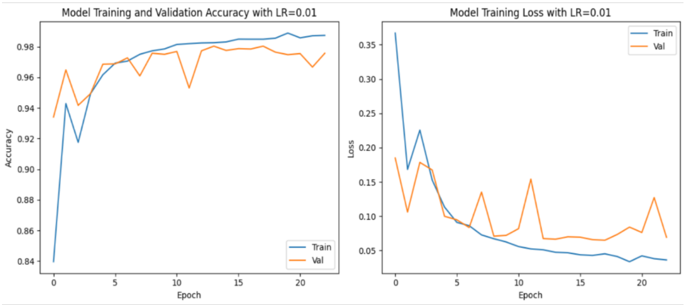

Figure 9: Word2Vec Training and Validation Accuracy(lr=0.01)

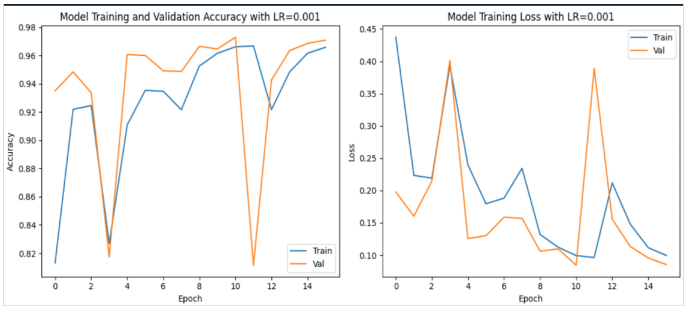

Figure 10: Word2Vec Training and Validation Accuracy(lr=0.001)

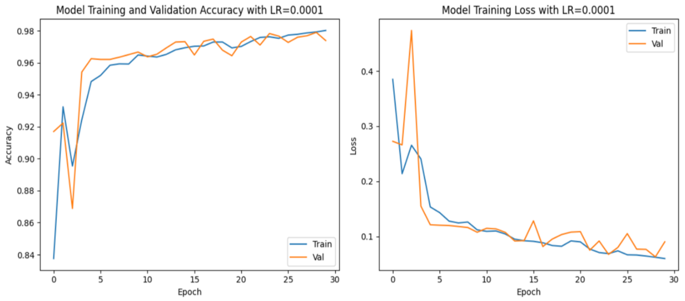

Figure 11: Word2Vec Training and Validation Accuracy(lr=0.0001)

V Conclusion and Future Work

The project shows that using Word2Vec and Glove embeddings with a Bi-LSTM model effectively identifies AI-generated and human-generated text. The models achieved notable accuracy and precision, demonstrating their utility in educational contexts for detecting AI-generated content.
For future expansion, there is a need to address potential ethical issues such as the impact of false positives on students, and also expand the dataset to include different samples, such as essays by children to test the robustness of the model.

VI Contributions

Charishma Rathan Bala, Gowtham Edumudi - Data Preprocessing (Cleaning and Splitting).
Sruthi Mandalapu, Poojitha Kinthada - Word Embeddings, Bi-LSTM Model Implementation.
Moroti Sonde, Chandrika Kancheti - Results and Analysis of Word Embeddings (Comparing the Accuracy and Precision).

References

[1] Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. "Classification of Human- and AI-Generated Texts: Investigating Features for ChatGPT". In: *Lecture Notes on Data Engineering and Communications Technologies*. Springer Nature Singapore, 2023, pp. 152–170. ISBN: 9789819979479. DOI: `10.1007/978-981-99-7947-9_12`. URL: `http://dx.doi.org/10.1007/978-981-99-7947-9_12`.

[2] Francesco Greco et al. "David versus Goliath: Can Machine Learning Detect LLM-Generated Text? A Case Study in the Detection of Phishing Emails". In: ().

[3] *Jules King, Perpetual Baffour, Scott Crossley, Ryan Holbrook, Maggie Demkin. (2023). LLM - Detect AI Generated Text. Kaggle.* `https://kaggle.com/competitions/llm-detect-ai-generated-text`.

[4] *Kaggle.* `https://www.kaggle.com/datasets/thedrcat/daigt-proper-train-dataset/`.