Team ID: 21
Team Name: RandomClassics

GenAI Content Detection - Workshop on Detecting AI-Generated Content

Abstract

With the growth of AI that can generate text, it's important to distinguish between text written by humans and text created by AI. This project aims to make a system that can easily tell if the text is written by a human or made by AI, using three models: XGBoost, ALBERT, and DistilBERT.

To get the text ready for the models, we cleaned and processed it using a few steps. First, we broke the text into smaller parts, like individual words, in a process called tokenization. Then, we simplified the words to their basic forms using lemmatization (for example, changing "running" to "run"). We also removed common, unimportant words like "the" and "is," called stopwords, and turned the words into numbers using a method called TF-IDF vectorization. This made it easier for the models to work with the data. [1]

Next, we fine-tuned the models to make them perform as well as possible. We used a tool called RandomizedSearchCV to try out different settings (called hyperparameters) for each model and find the best combination. [2]

To see how well the models were doing, we used a few key measurements. Models were evaluated using metrics like accuracy, precision, recall, F1 score, and confusion matrix.

We also created visuals to explain the results. ROC curves showed how well the model found AI-written texts without wrongly marking human-written texts as AI. Feature importance plots highlighted the key factors the models used to make decisions. We used a t-SNE plot to visually map how human-written and AI-generated texts are grouped and a training vs. validation loss graph to show how the models improved as they learned.

In the end, XGBoost performed better than both ALBERT and DistilBERT, achieving an impressive accuracy of 95%. DistilBERT and ALBERT, however, achieved accuracies of 50.46% and 50.34%, respectively.

Goal of the Project

The goal of our project is to build a model that can accurately classify text responses as either AI-generated or human-generated. We will train the model using a dataset of labeled examples, where each example is marked as AI-generated or human-generated. The model will learn the patterns and differences between how AI and humans generate text by analyzing linguistic features, word usage, sentence structure, and writing style.

Methodology

This project used two models, XGBoost and BERT: For XGBoost, the first step was data preprocessing. This involved combining all text entries, removing stopwords, and applying lemmatization and stemming to simplify the words. The text was then transformed into numerical features using TF-IDF vectorization. After preprocessing, the data was split into training, validation, and test sets. The model was trained using 100 estimators, and its performance was evaluated using metrics such as accuracy, precision, recall, F1 score, and a confusion matrix. Additionally, a graph of training loss vs. validation loss was plotted to track the model's performance during training. To improve accuracy, RandomizedSearchCV was used to fine-tune the model's hyperparameters, and the final model was tested on the test set.[3]

For ALBERT and DistilBERT, the same preprocessing steps were applied as for XGBoost, with an added step for tokenization, which is essential for preparing text for transformer models. ALBERT is a lighter version of BERT with faster training times, while DistilBERT is a smaller and more efficient version of BERT. Both models were trained for three epochs, and their performance was tested on the test set. Although these models are designed to be more efficient, they still needed fine-tuning to achieve the best results. [4][5]

Results

Metric	XGBoost	DistilBERT	ALBERT
Accuracy	0.95086	0.5046	0.5034
Precision	0.95091	0.5646	0.4929
Recall	0.95083	0.5083	0.5077
F1 Score	0.95086	0.3415	0.3585

Table 1. Performance comparison between three different model

Analysis

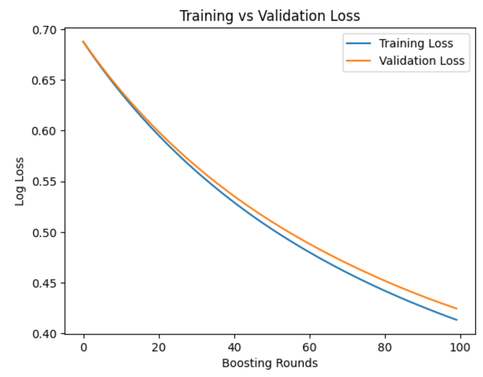

Figure 1. Training vs Validation Loss in Xgboost

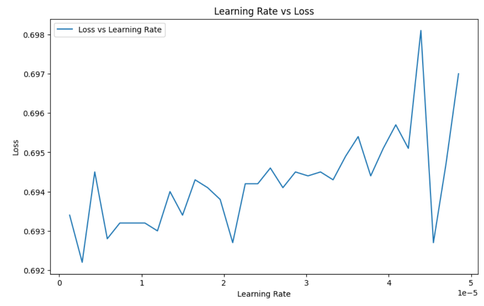

Figure 2. Learning Rate vs Loss in DistilBERT

Limitations and Future Work

Limitations: One problem with this project was the size of the dataset. A bigger and more varied dataset could have helped the models work better on different types of text. Another issue was that training BERT took a lot of time and computing power. This made it hard to test more training rounds or try different settings to improve its performance.

Future Work: In the future, we would try to optimize the AlBERT model to use the best features and get better results. We would also experiment with other ways of turning text into numbers for XGBoost, like Word2Vec or GloVe, which can better understand the meaning of words compared to TF-IDF

References

[1] R. Agerri and A. Garcia-Serrano, "Preprocessing text for text mining: Techniques and tools," *Journal of Computer Science and Technology*, vol. 32, pp. 1–18, 2017. DOI: 10.1007/s11390-017-1756-4.

[2] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl, "Random search for hyper-parameter optimization," in *Proceedings of the 14th International Conference on Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1–9. [Online]. Available: https://arxiv.org/abs/1206.2944.

[3] D. Dimitrov, F. Alam, M. Hasanain, *et al.*, "Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes," in *Proceedings of the 18th International Workshop on Semantic Evaluation*, ser. SemEval 2024, Mexico City, Mexico, Jun. 2024.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2020.