SCaLAR at SemEval-2024 Task 8: Unmasking the machine : Exploring the power of RoBERTa Ensemble for Detecting Machine Generated Text

Anand Kumar M
Department of IT
NITK, India

Abhin B
Artificial Intelligence
Department of IT
NITK, India

Sidhaarth Sredharan Murali
Artificial Intelligence
Department of IT
NITK, India

Abstract

SemEval SubtaskB, a shared task that is concerned with the detection of text generated by one out of the 5 different models - davinci, bloomz, chatGPT, cohere and dolly. This is an important task considering the boom of generative models in the current day scenario and their ability to draft mails, formal documents, write and qualify exams and many more which keep evolving every passing day. The purpose of classifying text as generated by which pretrained model helps in analyzing how each of the training data has affected the ability of the model in performing a certain given task. In the proposed approach, data augmentation was done in order to handle lengthier sentences and also labelling them with the same parent label. Upon the augmented data three RoBERTa models were trained on different segments of data which were then ensembled using a voting classifier based on their R2 score to achieve a higher accuracy than the individual models itself. The proposed model achieved an overall validation accuracy of 97.05% and testing accuracy of 76.25%.

1 Introduction

In the current day scenario, AI has noticed a major boom due the emergence of Large Language Models (LLMs) in the field of Natural Language Processing (NLP). These LLMs are capable of generating text with any given context that they have been trained on making them versatile to a lot of applications. LLMs have also showcased their unrivaled ability to code basic to complex programs. Many Large Language Models (LLMs) depend heavily on the data used for their training. Consequently, they may occasionally provide inaccurate information, especially in contexts where precision is crucial, such as sensitive or professional advice. Hence AI-generated text classification has become increasingly important due to the surge in the use of language models for content creation. Accurately identifying the source of a text, whether human-written or generated by a specific language model, is crucial for various applications, such as combating misinformation and plagiarism detection. Subtask B - Multi-Way Machine-Generated Text Classification shared task aims to not only detect text generated by these language models, but also specifically distinguish between outputs generated by different models. This in a real life scenario helps in determining the transparency and vulnerability of a model to attacks and reasoning as to why particular models perform in certain ways. Different contributions of the paper is as follows :-

- **Data Augmentation:** Data augmentation is a crucial task of increasing the volume of available data with specific manipulations which also helps build a more robust model able to tackle edge case scenarios. We propose a novel approach to handle long texts. We initially set a specific threshold to split them into smaller segments while preserving label information, ensuring efficient model training.

- **Ensemble Learning:** Ensemble learning as name suggests weaker models are brought together to achieve a better model with enhanced performance. We employ a weighted ensemble voting classifier that combines the predictions of multiple models trained on diverse validation sets, leading to improved generalizability and robustness.

We observe how effective and relevant Language models are in tackling Natural Language Processing tasks such as the current shared task when compared to other neural network based LSTM or other sequence models. Our final submission had a test accuracy of 76.25% and our standing was 18th position in the leader-board.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1135–1139

2 Background

Our work improves model generalizability in large-scale tasks by utilising insights from (Wang et al., 2024) and building on recent studies. Previously methods proposed by authors in (Ma et al., 2023) collected 500 scientific articles from 10 domains including biology, chemistry, IT and others and used chatGPT to paraphrase texts for each article. The authors extracted certain features such as perplexity, semantic document and six others to use classifiers on these extracted features. The authors used three classifiers: XGBoost, random forest, and multi-layer perceptron, to train and test models for detecting human-generated and AI-generated texts, as well as human-generated and AI-rephrased texts. They performed a 5-fold cross-validation and evaluated the models using accuracy and F1-score which majorly motivated our approach. Another work (Mindner et al., 2023) used similar techniques to the previous one while using school topics as their dataset. In their work (Abburi et al., 2023) the authors use ensemble neural model that generates probabilities from different pre-trained LLMs which are used as features to a TML classifier following it. Author in (Huimin et al., 2018) presents his work on text classification ensemble learning method based on multi-angle perturbation heterogeneous base classifiers and validates the effectiveness of the algorithm through experiments. In a similar work (Mohammed and Kora, 2022) the authors propose a new meta-learning ensemble method that fuses baseline deep learning models using 2-tiers of meta-classifiers.

Furthermore, our method for comprehending model decision-making in short text classification—particularly in identifying AI-generated content—is influenced by methods from works on short text classification. Authors in their work (Tang et al., 2022) use a sliding window to align the sentences with the labels and preserve the edge characteristics of the long text. Another work in the same field (Shorten et al., 2021) categorizes text data augmentations into symbolic and neural methods. Symbolic methods use rules or discrete data structures to form new examples, while neural methods use auxiliary neural networks to sample new data. Our research aims to advance the development of robust and adaptable machine learning models customised to particular tasks through this synthesis of diverse viewpoints. These viewpoints are then combined back again with the help of ensembling ensuring no loss of data.

3 Methodology

Our methodology majorly focuses on exploiting Pre-trained language models such as the RoBERTa model (Liu et al., 2019) and enhancing its performance through a much simpler traditional approach of ensemble learning. We worked on the M4 based dataset with our methodology (Wang et al., 2023) The ensemble model shows better performance compared to all 3 RoBERTa-base models which were trained on different segments of augmented data. It reduces over-fitting and increases interpretability for any given task.

3.1 RoBERTaForSequenceClassification

RoBERTa which stands for Robustly Optimized BERT Approach is a variant of the famous BERT model (Devlin et al., 2018) which was developed by Google in 2018. RoBERTa was later introduced by researchers at Facebook AI in 2019. It builds upon the architecture of BERT while bringing in few major changes. It uses Dynamic masking strategies and removes the Next Sentence Prediction (NSP) in its pre-training step. It is further pre-trained with larger amounts of data with larger mini-batch size. The novelty of RoBERTa lies in its ability to achieve state-of-the-art performance on various natural language understanding benchmarks by leveraging advancements in pre-training techniques and model architectures. RoBERTa continues to employ similar tokenizing technique as BERT with WordPiece Encoder (WPE). RoBERTa as a base model in itself gives out embedding for a given sentence or a word as it is only composed of Encoder architecture.

RoBERTaForSequenceClassifcation consists of a classification head on top of the base RoBERTa model. This classification head maps the backbone outputs to logits suitable for a classification task based on the number of labels provided.

3.2 Ensemble Learning

Ensemble learning is a machine learning technique that combines multiple individual models to obtain a model with enhanced performance which is more robust as well. It involves training several individual base models which are often referred to as experts on similar data and producing an aggregation out of those models based on their individual performances. The benefits of ensembling

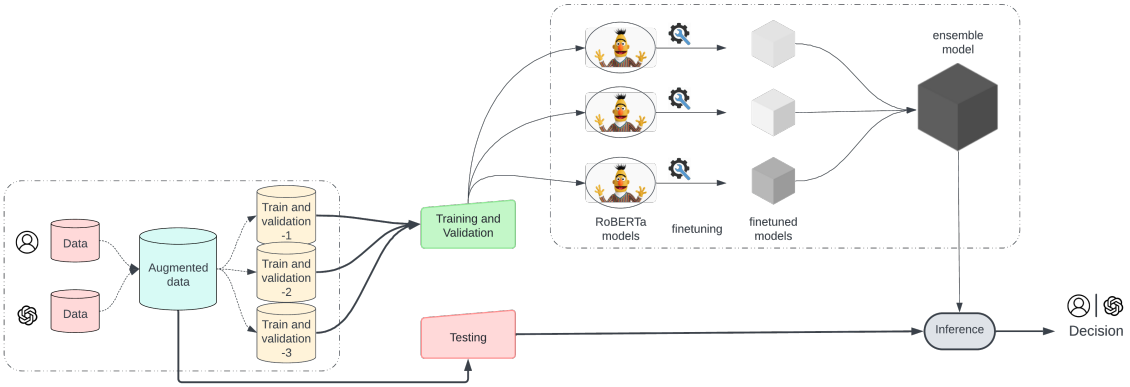

Figure 1: Proposed methodology for AI vs Human generated text Detection using weighted voting ensembling of RoBERTa classifiers

include improved generalization, more robustness compared to single models, and efficient as it compensates for the loss in performance of the poor learning algorithms. The common techniques in Ensemble learning including Bagging and Boosting. We specifically used the concept of Voting classifier which takes predictions from different models and has a specified weighting parameter based on which it gives out the final prediction. We implemented our own Voting classifier which scores the three RoBERTa models based on the R2-scores achieved by their predictions. R2-scores here are used as the weighting parameter for the prediction and thus we derive our final prediction out of the voting classifier.

4 System Overview

As mentioned, we first perform data augmentation. Before we get into the details of our experimental setup, we want to elaborate on different measures we took in order to augment our data. Data augmentation for training data was performed by carefully splitting the validation data while noting that there is no major imbalance in the class distribution. This was followed by training three different RoBERTa models on different combinations of training and validation dataset. We had 3 validation data splits namely *val1*, *val2* and *val3*. For *model1* we used *val2* and *val3* in training and *val1* for validating the *model1* and so on.

4.1 Data Augmentation

We implemented a data augmentation strategy to address instances in our dataset exceeding the token limit, ensuring no information loss while maintaining model compatibility. Given a dataset comprising 71,027 instances for training and 3,000 for validation, with some instances surpassing the 512-token limit, we devised a method to split these instances into k different segments. Utilizing the modulo operation, if an instance contains n tokens, $[n/512]$ determines the number of segments it will be divided into, while the remainder represents the number of tokens in the last augmented segment of the instance. This process yielded approximately 9985 additional instances for training and 188 for validation.

Subsequently, we merged the augmented training and validation sets to form a combined dataset of 81,012 training instances and 3188 validation instances. This validation dataset was then partitioned into three parts of which two-thirds are used for training alone with the complete training data and the remaining one-third is used for validation. Notably, each RoBERTa model was provided with a distinct subset of one-third of the validation data, thus adhering to a different k-fold validation scheme to enhance generalizability.

4.2 Implementation Details

The implementation of our method includes three vanilla RoBERTaForSequenceClassification models with 12 encoder layers with a classification head at the end were used. These models were trained on three different splits of two-thirds of validation data coupled with the training data. Each model was effectively trained on roughly 82000 samples with roughly 1060 validation instances. The voting classifier first takes in all three fine-tuned RoBERTa models and predicts on the complete validation set and analyzes the performance of each of the models based on their R2-score and constructs a weighted

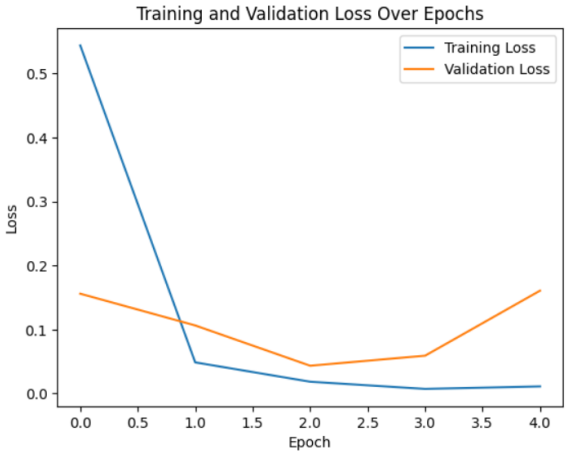

Figure 2: Training and validation loss observed over the RoBERTa model-1.

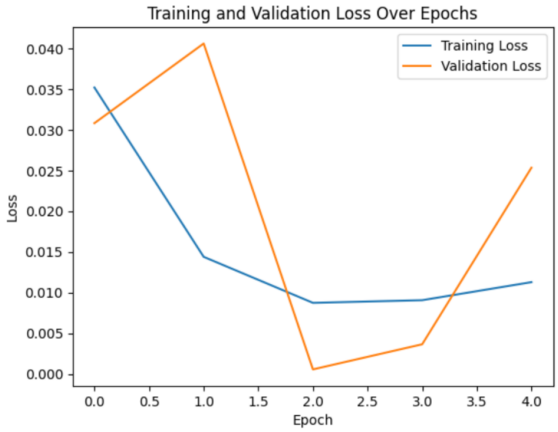

Figure 3: Training and validation loss observed over the RoBERTa model-2.

voting classifier which gives our final predictions. The R2 scores observed for each of the three models were 0.36, 0.29 and 0.35 which had roughly similar weight given to each of their predictions. The performance of each of the models were analyzed with the help of training and validation loss plots across training epochs.

5 Experimental Results

As a part of our experimental setup we used P100 GPU which is available through kaggle. Further we used the `RobertaForSequenceclassification` available through transformers library along with `RobertaTokenizerFast` for the modelling aspect. The learning rate used was a fixed one and we found it optimal at $1e-5$ along with $CrossEntropyLoss$. $AdamW$ optimizer was used with weight decay coefficient of $1e-2$ and $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Batch size of 20 was

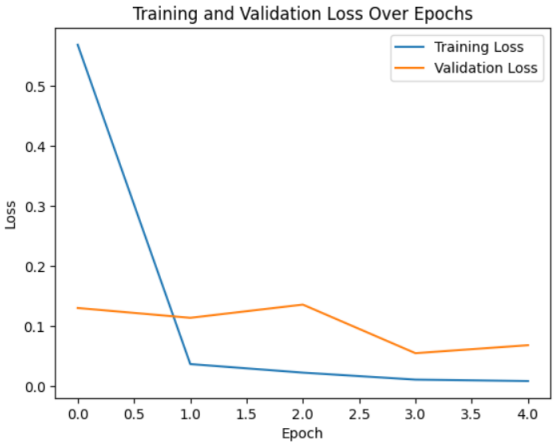

Figure 4: Training and validation loss observed over the RoBERTa model-3.

used for training and validation.

Our proposed methodology beat the baseline model which was a RoBERTa model with an average accuracy of 0.75. Our experimental results with respect to each of the RoBERTa model is displayed along with the improvement in performance with the use of R2-score based weighted voting classifier. In testing phase our model gave an accuracy of 76.25% which shows clear signs of over-fitting compared to 97.05% in validation accuracy.

Table 1: Proposed methodology performance comparison

Models	*Train Acc (%)*	*Val Acc (%)*
Baseline	75	75
RoBERTa-1	96.40	95.06
RoBERTa-2	93.64	92.10
RoBERTa-3	97.21	96.62
Voting Classifier (*proposed*)	**97.26**	**97.05**

6 Conclusion

In the proposed methodology, we beat the baseline RoBERTa model and further enhance the performance of the model using R2-score based Voting classifier. The model has performed well on the training data when compared to testing data which shows slight signs of over-fitting. In the light of ensemble learning for Pre-trained language models we see that the models are very sensitive to over-fitting hence should be used with caution. Techniques like early stopping and using data augmen-

tation. Further on embeddings from LLMs can be used to tackle this task more effectively.

References

Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. *arXiv preprint arXiv:2309.07755*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fan Huimin, Li Pengpeng, Zhao Yingze, and Li Danyang. 2018. An ensemble learning method for text classification based on heterogeneous classifiers. *International Journal of Advanced Network, Monitoring and Controls*, 3(1):130–134.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. Ai vs. human–differentiation analysis of scientific content generation. *arXiv*, 2301.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human-and ai-generated texts: Investigating features for chatgpt. In *International Conference on Artificial Intelligence in Education Technology*, pages 152–170. Springer.

Ammar Mohammed and Rania Kora. 2022. An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, 34(10):8825–8837.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8:1–34.

Changhao Tang, Kun Ma, Benkuan Cui, Ke Ji, and Ajith Abraham. 2022. Long text feature extraction network with data augmentation. *Applied Intelligence*, 52(15):17652–17667.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.