IUCL at SemEval Task 8: A Comparison of Traditional and Neural Models for Detecting Machine Authored Text

Srikar Kashyap Pulipaka, Shrirang Rajendra Mhalgi,
Joseph Edward Larson, Sandra Kübler
Indiana University
{spulipa,srmhalgi,joelarso,skuebler}@iu.edu

Abstract

Since Large Language Models have reached a stage where it is becoming more and more difficult to distinguish between human and machine written text, there is an increasing need for automated systems to distinguish between them. As part of Sem-Eval Task 8, Subtask A: Binary Human-Written vs. Machine-Generated Text Classification, we explore a variety of machine learning classifiers, from traditional statistical methods, such as Naïve Bayes and Decision Trees, to finetuned transformer models, such as RoBERTa and ALBERT. Our findings show that using a finetuned RoBERTa model with optimized hyperparameters yields the best accuracy. However, the improvement does not translate to the test set because of the differences in distribution in the development and test sets.

1 Introduction

Large Language Models (LLMs) are becoming more and more accessible, which has resulted in an increase in machine-generated content across a wide variety of domains, including education, technology, and science. With this increase in machine generated texts from LLMs, and with the increase in the quality of LLM created texts, concerns regarding but not limited to fake product review generation (Adelani et al., 2019) spam/phishing (Weiss, 2019) and fake news generation (Zellers et al., 2019; Brown et al., 2020; Uchendu et al., 2020) have arisen. Weiss (2019) demonstrated that humans can only detect such misuses of LLMs at chance level, which demonstrates the clear need for automated systems to detect machine generated content. In this paper, we describe the IUCL submission to SemEval task 8 (Wang et al., 2024); we focused mostly on comparing traditional and neural models. Our best system ranked 70th out of 137 submissions.

2 Related Work

In terms of impressionistic differences between human generated text and LLM generated text, it has been observed that LLMs tend to be more focused (i.e. less diversion from the subject at hand), more objective, and highly formal. Human texts, on the other hand, are overall more emotional, subjective, and less formal. In terms of linguistic difference, humans use fewer nouns and conjunctions, while employing more punctuation and adverbs. Dependency relations are also shown to be shorter. Lastly, human texts have higher type/token ratios in texts of the same length (Guo et al., 2023) Current LLM models include GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), CTRL (Keskar et al., 2019) and ChatGPT.

We will first begin by discussing statistical approaches to detecting machine-generated content, then using LLM technology itself to do so.

Solaiman et al. (2019) use a bag-of-words approach with TF-IDF feature vectors (both unigrams and bigrams) and a logistic regression model to differentiate between human-written web pages and text generated web pages from GPT2. They examine a different number of parameters of the LLM (117M, 345M, 762M and 1,542M) as well as different sampling methods (k-sampling, p-sampling and pure sampling). This is because an assumption that many researchers take is that language models sample from the head to generate natural looking text e.g. max sampling (Gu et al., 2017) and k-max sampling (Fan et al., 2018). Their findings are that the larger the LLM, the harder to detect how machine-like the generated text is and k samples are easier to detect than pure samples, probably due to the fact that k samples over-produce common words, which is easy to detect using statistical methods.

Gehrmann et al. (2019) use BERT and a group of statistical features: the probability of each word, ab-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1026–1031

solute rank of each word, and entropy of the distribution, and create a tool for users to see specifically what features are more likely to be machine generated over human generated. They clearly show that the model GPT-2 oversamples certain words; it is worth pointing out, however, that as LLMs grow more sophisticated, such methods may not work as well.

Solaiman et al. (2019) use finetuning on RoBERTa and find that it can detect text generated from GPT-2 with an accuracy of 95%. The RoBERTa detector has also been used in detecting fake news articles from several LLMs (Uchendu et al., 2020), Amazon product reviews (Adelani et al., 2019), and biomedical texts (Rodriguez et al., 2022).

3 Data

We used the M4 dataset (Wang et al., 2023) provided by the SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. We used the English data provided for Subtask A, the Monolingual (English) binary classification task.

The dataset for this subtask consists of 119,757 samples of human-written and machine generated text. There are an additional 5,000 samples as a development set. The test set consists of 34,272 samples.

About 53% of the samples in the training set are machine generated while the rest are human written. The machine generated text was produced by a range of models: ChatGPT and DaVinci by OpenAI, Dolly by Databricks, Cohere. The sources from which the human texts are taken are Reddit, WikiHow, ArXiV, Wikipedia and PeerRead. In contrast, the development set consists of an equal ratio of human and machine generated samples. The machine generated samples are entirely from the Bloomz model. The human sources are also equally distributed between WikiHow, Wikipedia, Reddit, ArXiV and PeerRead. In the test set, 52.5% of the texts are machine generated with GPT4, Cohere, ChatGPT (GPT3.5), Bloomz, Dolly, and DaVinci as sources. Note that this means optimizing a system on development data is difficult since the test data are much closer to the training data than the development data.

Further details about the data and the task are available at the overview of the shared task (Wang et al., 2024).

We present a comparison of a range of classifiers (see below). For those experiments, we use the development set of 5,000 samples for benchmarking and finetuning the model performance.

4 Methods and Features

4.1 Features

Ratio features We started with extraction of features from the dataset that cannot be controlled consciously by authors: stopword ratio and average sentence length. We used the NLTK stopwords[1] (Bird et al., 2009) to calculate the stopword ratio for the dataset. The left graph in Figure 1 shows the distribution for the sentences generated from different sources. The median stopword ratio for humans and different models are around 0.40. It is difficult to distinguish human text from machine text as the distributions of the texts generated by machines are similar to those of the human generated texts. We then computed the average sentence length generated by different sources, see the right graph in Figure 1. The average number of the sentences generated in each of the category is around 21. Again, there is little difference between machine and human generated texts.

Textual features We also used TF-IDF and word unigram features.

4.2 Statistical Learning Methods

We used the ratio features to train Multinomial Naïve Bayes, Random Forest, XGBoost, Logistic Regression and SVC models on the data. For the textual features, we trained SVC, Decision Tree, Logistic Regression and Random Forest classifier models. For all models, we used the scikit-learn implementations (Pedregosa et al., 2011).

We chose the Naïve Bayes classifier because of its simplicity and the ability to handle missing data values. Support Vector Classifier is better at handling high dimensional spaces and is robust to overfitting. Random Forest is an ensemble learning method which is robust to overfitting and provides feature importance ranking, helping to identify the most influential features. Logistic Regression and Multinomial Naïve Bayes classifiers are easy to interpret and are computationally efficient. XGBoost provides a gateway to handle data in a highly efficient and scalable manner. Because of time constraints, we did not perform any hyperparameter

[1] https://gist.github.com/sebleier/554280

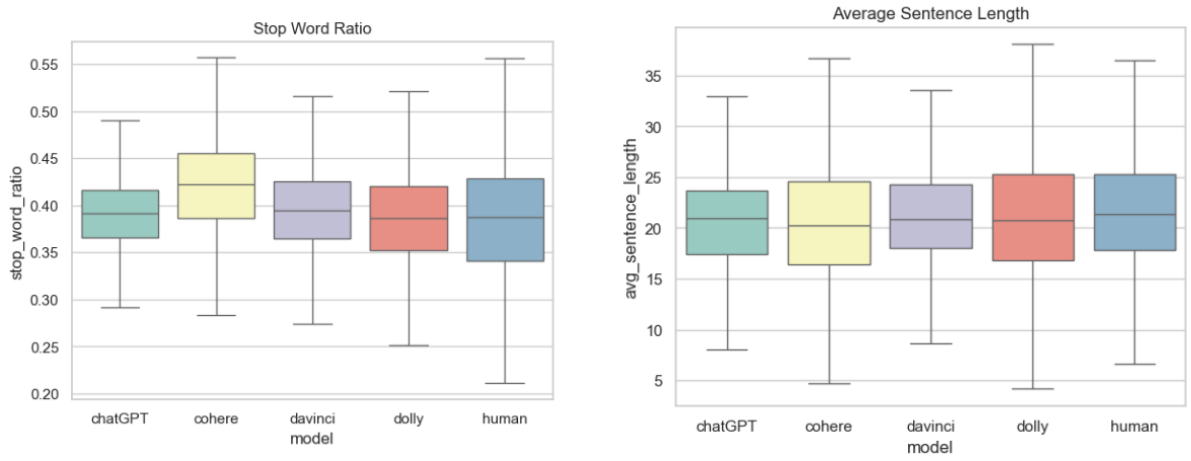

Figure 1: Stop word ratio (left) and average sentence ratio (right) generated by different models.

tuning, and used the default settings to train the models.

4.3 Deep Learning Methods

Fully connected DL model We used the same data preprocessing techniques described in Section 4.1 and trained a fully connected 2 layer neural network having 512 hidden units with ReLU activation. We used the Binary Crossentropy to calculate the loss and Adam optimizer to train our neural network on 100 epochs. We set the batch size to 2048, due to processing limitations and kept a learning rate of 0.001 with an early stopping mechanism in place.

Finetuned Language Models We also finetuned the following language models: BERT and its derivative models ROBERTa and ALBERT. We use the Hugging Face library (transformers) for this task.

BERT (Bidirectional Encoder Representations from Transformers) is a language model developed by Devlin et al. (2019). It is a bidirectional model that uses a transformer architecture. We use the BERT base model for our experiments.

RoBERTa is a variant of BERT developed by Liu et al. (2019). It is pre-trained on a larger corpus of texts. We use the RoBERTa base as well as large models for our experiments. The best performing model of our study is a RoBERTa base model. ALBERT is a smaller version of BERT developed by (Lan et al., 2020). The hyperparameters selected are shown in Table 1.

	RoBERTa	BERT	ALBERT
Learning Rate	5e-5	2e-5	2e-5
Batch Size	8	32	16
Nr. Epochs:	3	4	4
Grad. Acc. St.	4	2	2

Table 1: Hyperparameters for the neural models

5 Results

We will first discuss our results on the development data, then the official results of the shared task.

5.1 Results on the Development Set

The shared task provides a baseline accuracy of 74% using a RoBERTa model. Our aim is to investigate a range of models and features and incrementally improve models, starting out with traditional machine learning models and then moving on to deep learning models.

Table 2 shows the performance of the different combinations of models and features on the development set.

We first look at the statistical methods combined with the standard sparse features, bag of words, and TF-IDF weighted bag of words features. The results in the first block show that the TF-IDF weighted feature results in a lower accuracy than standard frequency counts (56.44% vs. 60.22%) for logistic regression. For this reason, we decided to concentrate on frequency counts. Among the different statistical classifiers, logistic regression reaches the highest results (60.22%), followed by XGBoost with 59.26%.

When we use the ratio features, i.e., stop word

Features	Model	Acc.
TF-IDF	Logistic Regression	56.44
words	Random Forest	58.86
	Naïve Bayes	50.54
	XGBoost	59.26
	Logistic Regression	60.22
Ratio features	Logistic Regression	67.14
BERT	Logistic Regression	63.48
	Fully connected NN	67.19
	Fully connected NN (optimized)	70.11
ALBERT	ALBERT	66.78
RoBERTa	RoBERTa BASELINE	*74.00*
	XLM-RoBERTa Large	77.67
	XLM-RoBERTa (10,000 training samples)	78.24
	XLM-RoBERTa Base Default	79.61
	XLM-RoBERTa Base (optimized)	**79.90**

Table 2: Model comparison with respect to features and accuracy for Dev Set

ratio and average sentence ratio, combined with logistic regression, we reach an accuracy of 67.14%, which is surprising in that this outperforms word features by almost 6% absolute, even though they did not show large differences in Figure 1.

Next, we investigate whether using BERT embeddings instead of sparse features improves results. When we use those features with logistic regression, results increase by 3% absolute to 63.48%, combining them with the fully connected neural network, we reach an accuracy of 70.11%, outperforming the ratio features, but not reaching the baseline provided by the shared task.

We then move on to use BERT and its variants. We start off with ALBERT, a smaller version of BERT. This model gives us an accuracy of 66.78%. This shows that we need a large scale model for good performance. We find that the XLM-RoBERTa model, a multilingual pre-trained model performs better than a RoBERTa model. An XLM-RoBERTa model with full data and default parameters gives us an accuracy of 79.61%. We add gradient accumulation to the finetuning process to speed up training and improve performance. We also reduce the batch size and adjust the learning rate, to get an incremental 0.3% improvement due to the hyperparameters. Optimizing hyperparameters tuning further increases accuracy to 79.90%. This is the best accuracy we have obtained in our experiments. When we compare those results to the XLM-RoBERTa large model with its higher number of parameters, accuracy drops to 77.67%, showing that simply increasing the number of parameters does not guarantee good performance.

A final experiment investigates the importance of the training set size. For this experiment, we reduce the training data to 10,000 samples. This model gives us an accuracy of 78.24%, showing that finetuning XLM-RoBERTa with even a small dataset reaches competitive results. Increasing the training set from 10,000 to about 120,000 results in an increase in accuracy of 1.66% absolute.

System	Score	Rank
Our submission	74.96	70
Baseline	88.46	–
safeai	96.88	1

Table 3: Official Results (accuracy).

5.2 Official Results

We generated our final predictions using the finetuned XLM-RoBERTa system. We show our results in comparison to the best system and the baseline in Table 3. Our submission had an accuracy of 74.96% on the test set and was ranked 70 out of 137 teams. The best ranking team had an accuracy of 96.88%. Note that while our system improved over the baseline for the development data, this is not the case for the test data. This is most likely a consequence of the different distributions between the development and test data.

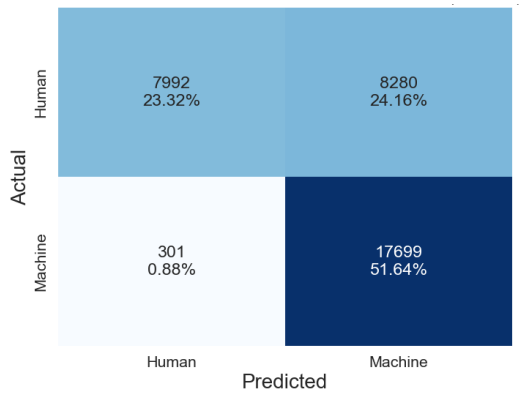

Figure 2: Confusion Matrix of the best model (Test Set)

5.3 Discussion

We had a closer look at the confusion matrix for the best performing model, the optimized XLM-RoBERTa model, on the test data, shown in Figure 2. We notice that the model has a tendency to incorrectly identify human samples as machine generated (false positives) in 8,280 cases, as opposed to just 301 cases of false negatives.

One of the limitations of our work is that we have not explored data processing and augmentation techniques that can help us improve the performance of the model.

6 Conclusion and Future Work

In this project, we have investigated the performance of various machine learning models. We found that our best performing model is a base XLM-RoBERTa model that is fine-tuned on the dataset. Using the smaller ALBERT or the large XLM-RoBERTa models resulted in decreases in accuracy. However, we also see that finetuning is very sensitive to underlying data characteristics, since the gains we saw on the development set did not translate to equivalent gains on the test set.

There is a significant scope for improvement in the performance of the models by working on further text preprocessing and feature engineering. Future work includes using ensemble methods that combines the finetuned models along with a model using ratio features.

Acknowledgements

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. arXiv.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020*, volume 33, pages 1877–1901, Vancouver, Canada.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, MN.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 889–898, Melbourne, Australia.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy.

Jiatao Gu, Kyunghyun Cho, and Victor O.K. Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978, Copenhagen, Denmark.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. In *arXiv*, 2301.07597.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. In *arXiv*, 1909.05858.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the Eighth International Conference on Learning Representations*.

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. In *arXiv*, 1907.11692.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. In *arXiv*, 1908.09203.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online.

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, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Max Weiss. 2019. Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. *Technology Science*, 2019121801.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA.