A Natural Approach for Synthetic Short-Form Text Analysis

Ruiting Shao[1], **Ryan Schwarz**[2], **Christopher Clifton**[2], **Edward J. Delp**[1]
[1]School of Electrical and Computer Engineering, [2]Department of Computer Science
Purdue University, West Lafayette, Indiana, USA
{shao72, schwarzr, clifton, ace}@purdue.edu

Abstract

Detecting synthetically generated text in the wild has become increasingly difficult with advances in Natural Language Generation techniques and the proliferation of freely available Large Language Models (LLMs). Social media and news sites can be flooded with synthetically generated misinformation via tweets and posts while authentic users can inadvertently spread this text via shares and retweets. Most modern natural language processing techniques designed to detect synthetically generated text focus primarily on long-form content, such as news articles, or incorporate stylometric characteristics and metadata during their analysis. Unfortunately, for short form text like tweets, this information is often unavailable, usually detached from its original source, displayed out of context, and is often too short or informal to yield significant information from stylometry. This paper proposes a method of detecting synthetically generated tweets via a Transformer architecture and incorporating unique style-based features. Additionally, we have created a new dataset consisting of human-generated and Large Language Model generated tweets for 4 topics and another dataset consisting of tweets paraphrased by 3 different paraphrase models.

Keywords: Natural Language Processing, Natural Language Generation, Synthetic Text Detection, Authorship Attribution, Large Language Model

1. Introduction

The impact of general misinformation and bot generated text has been witnessed on a large scale in the last decade. In 2014, Twitter was flooded with an army of bots tweeting about a small technology company, Cynk (Ferrara et al., 2016). This flurry of artificial posts created a large amount of chatter, which automatic trading scripts attempted to capitalize upon. This led to the stock price inflating by over 500%. When it was discovered the original social media posts were synthetic, the stock price responded by decreasing below its original value, trading was halted, and unfortunate investors were left to realize massive financial losses.

In 2016, both the U.S. presidential election and Brexit referendums were believed to have been partially influenced by twitter bots (Gorodnichenko et al., 2021).

Although currently most discovered bot activity incorporates manually written sentences rather than model-generated ones (Vargo et al., 2018), with the proliferation and availability of robust Large Language Models (LLMs), such as ChatGPT, it is highly likely that future bot activity will include some combination of synthetic and human-generated sentences.

In this paper, we propose a method of synthetic text detection on tweets via an ensemble of reasonable stylistic features incorporated with LLM-extracted ones. We also briefly explore the task of authorship attribution using LLM-generated content rather than traditional authors and the effects of our techniques on short-text samples. Finally, we analyze the potential threat to both detection and attribution from paraphrasing attacks (Sadasivan et al., 2023).

2. Related Work

2.1. Language Generation

Language generation can easily be structured as a product of conditional probabilities lending to its sequential nature (Radford et al., 2019).

$$P(x) = \prod_{i=1}^{n} P(s_i | s_1, s_2, ...s_{n-1})$$

where x represents a sample of generated text and s_i represents individual tokens at the i^{th} location. With the invention and popularity of self-attention architectures, such as the transformer (Vaswani et al., 2017), many language models have been created which can estimate these probabilities with sufficient prose and verbosity. While the transformer uses an encoder-decoder structure to understand language, popular models such as the Generative Pretrained Transformer (GPT) series from OpenAI (Radford et al., 2018, 2019; Brown et al., 2020; Wang and Komatsuzaki, 2021) and BERT (Devlin et al., 2018) make use of either the encoder or decoder for increase performance in certain tasks.

LREC-COLING 2024, pages 1034–1042
20-25 May, 2024.

2.2. Detection

Detecting short-form text generated by an LLM is a relatively unexplored and challenging task for many applications. Previous work on general synthetic text detection, such as GROVER, incorporate a transformer-based architecture as both a generator and detector on paragraph and article-length text sequences (Zellers et al., 2019). This is helpful in the context of determining the validity of a news article attempting to spread propaganda or long-form social media post containing misinformation but relies heavily on the stylometric features of the given text. The shorter the sequence of text becomes, the less importance the same stylometric features have on aiding in a classification (López-Escobedo et al., 2013).

Recently, the unique TweepFake dataset has been created specifically for the purpose of synthetic tweet detection and attribution to specific bot and human authors (Fagni et al., 2021). While high accuracy was achieved by BERT-based classifiers such as RoBERTa (Liu et al., 2019) the data chosen was largely comprised of authentic human accounts and bots impersonating those humans making the dataset more effective when analyzing detection in the context of a specific, well-known user. Other variations of BERT, such as BERTAA (Fabien et al., 2020), have also proven successful at the tangential task of authorship attribution on similar datasets.

In similar work, tweets across a twitter users timeline were collected and analyzed for potential synthetically generated samples (Kumarage et al., 2023). The authors analyzed timelines consisting of wholly synthetic or authentic tweets, it also placed an emphasis on determining a point in a timeline where tweets became synthetic, in the event of an account hijacking. Intuitively, it showed that the shorter the timeline is, the harder it was to accurately classify the synthetic text. This is likely due to less overall text leading to a lower amount of semantic information for the model to learn. The authors did however note, that for lower values of timeline transition point there was an increase in benefit from infusing external stylometric features compared to using word embedding and bag of word representations.

Incorporating metadata into detection methods can greatly improve a classifier's results (Hovy, 2016), however this data is not always available. A synthetically generated tweet can often be incorrectly attributed to a legitimate author, spread by legitimate users through retweets and shares, or displayed independently of twitter altogether via articles, news reports, and memes. These factors make it difficult to rely on external features derived from metadata in a realistic scenario.

Work such as (Sadasivan et al., 2023) further examines the difficulties of detecting Artificial Intelligence (AI) generated text and describes the problem in terms of a given classifiers Area Under Receiver Operator Characteristic (AUROC):

$$AUROC(D) \leq \frac{1}{2} + TV(M,H) - \frac{TV(M,H)^2}{2}$$

where D represents a synthetic text detector, TV references the total variation distance between two distributions, and M, H represent the distributions of machine-generated and human-generated samples, respectively. As the TV distance shrinks, i.e., two distributions become more similar, any classifier D will tend towards a random classifier. Shorter text, which has less unique characteristics between synthetic and human-generated text, inherently reduces TV distance and thus detector accuracy.

This difficulty even extends to watermarked text, such as the work presented in (Kirchenbauer et al., 2023). Watermarking techniques attempt to apply a machine detectable watermarked pattern to text generated by an LLM while hiding the pattern from the average human reader. Popular techniques involve dividing the vocabulary $|V|$ of an LLM evenly into a red and green list of tokens. In a hard watermarking scheme, only the green tokens will be considered during token generation. This has the effect of providing an easily detectable pattern, with the trade-off of sentence verbosity. The more popular soft watermarking scheme samples from the green tokens inverse to the entropy from a given prompt. The phrase "The quick brown fox", for example, has an incredibly low entropy with an almost deterministic completion of "jumps over the lazy dog". Therefore, even if one of the completion tokens is on the red list it will likely be chosen. Whereas a prompt with relatively high entropy is almost certain to generate a token from the green list.

In this case, detection simply involves comparing the number of tokens seen in a sample with the red list tokens. Because the lists are evenly divided, a human generated sample will utilize approximately 50% from the red list, where the watermarked model will utilize almost none. A simple p-score threshold provides a highly accurate detector of the applied watermark. Unfortunately, these types of schemes are susceptible to spoofing attacks. A malicious actor, with sufficient access to known watermarked text, can recreate the green list tokens with a high degree of accuracy (Sadasivan et al., 2023).

2.3. Attribution

Unlike detection, authorship attribution can be defined as a multinomial classification problem at-

tributing a given text to its corresponding author, generally from a list of many potential authors. The task of authorship attribution is a very old one, with a history which long predates computing (Stamatatos, 2009). However, many modern approaches to authorship attribution utilize transformer-based (Vaswani et al., 2017) models. These include BERTAA (Fabien et al., 2020), which builds upon BERT (Devlin et al., 2018) for the specific task of authorship attribution.

In this paper, we treated the generation models as "authors" for the author attribution task.

2.4. Paraphrasing

Paraphrasing can be defined as modifying a natural language input to contain different words while maintaining its semantic meaning to a human reader (Sadasivan et al., 2023). The process of paraphrasing generally has a shortening effect on the text, and in the case of authorship attribution, greatly increases distributional similarity between the classes.

In order to measure closeness and account for paraphrasing, there are various measures of distributional similarity we can use. Total Variation (TV) distance is a measure of the largest possible difference in probability between any event occurring in 2 distributions. Similarly, Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) measures the average difference between events in distributions by using one probability distribution to approximate another. However, some problems with the KL-divergence is its failure to satisfy the triangle inequality in some cases, making its use a a metric somewhat limited. Additionally KL-Divergence is asymmetric, meaning that $KL(P,Q)$ is not necessarily equivalent to $KL(Q,P)$. To solve this, we can use the square root of the Jensen-Shannon Divergence (Endres and Schindelin, 2003), also known as the Jensen-Shannon Distance (JSD). Similar to KL-Divergence, the Jensen-Shannon Divergence measures the average distance between probability distributions, however it is symmetrical and satisfies the triangle inequality. The Jensen-Shannon Divergence is formulated as:

$$JSD(P,Q) = D(P||\frac{P+Q}{2}) + D(Q||\frac{P+Q}{2})$$

This is a desirable metric because it vanishes when $P = Q$ and is symmetric, however this does not satisfy the triangle inequality, but its square root does. This f-divergence is closely related to the Kullback-Leibler divergence (Kullback and Leibler, 1951):

$$D_{KL}(P||Q) - \sigma P(x) log(\frac{P(x)}{Q(x)})$$

3. Datasets

To test our detection method on short-form text, we created a new dataset consisting of synthetically generated tweets from 3 popular LLMs: GPT2 (Radford et al., 2019), GPT3 (Brown et al., 2020), and GPT-J 6B (Wang and Komatsuzaki, 2021). Using the Twitter API, we extracted approximately $300k$ tweets across 4 categories including politics, science, climate, and covid. These tweets were selected from primarily verified Twitter accounts between 2016 and 2021. The categories were selected according to various hashtags and keywords as shown in Table 3.

We then used the corresponding APIs, provided by OpenAI, to fine-tune the 3 LLMs using randomly selected samples of $5,000$, $10,000$, and $15,000$ human-generated tweets from each of the 4 categories. These fine-tuned models were then used to generate $20,000$ synthetic tweets for each of our 4 categories. GPT2 and GPT3 could generate convincing tweets with no input prompt, while GPT-J 6B required the first 10 words of the GPT3-generated text as a prompt for completion. Both the human-generated and synthetically generated tweets contain English words and sentences as well as emojis, twitter links, and unique punctuation such as the twitter hashtag. [1]

We also evaluate our method on the TweepFake dataset (Fagni et al., 2021) which contains deepfake tweets which are generated based on Markov Chains, recurrent neural networks (RNN), long short-term memory networks (LSTM), GPT2 and other technologies.

For our paraphrased-related tasks, we created an additional dataset by paraphrasing the inputs of our initial tweet dataset while maintaining the original labels. The model used to paraphrase was similar to the model in Sadasivan et al. (2023), however the Pegasus-Paraphraser pretrained model was chosen rather than the Pegasus-Summarizer model, due to it not reducing the size of the original input as drastically. We also attempted to paraphrase data using the ChatGPT and T5 models to assess the effects of the paraphraser on detection and attribution.

4. Method

The block diagram of the proposed system is illustrated in Figure 1. The input tweet T_{org} will first be normalized and have noise removed, such as URLs and mentions, to get a cleaned tweet T_{clean}. At the same time, we will perform preliminary statistic calculations based on the original input tweet, such as word count per sentence, average word length and lexical richness. In addi-

[1] The dataset is available here

Category	Search Words	Size
Politics	#Trump, #DonaldTrump	30k
Science	#Science #Engineering #Physics #Biology #Chemistry	36k
Climate	#Climate #GlobalWarming #ClimateChange	54k
Covid	#Coronavirus #Covid #Covid-19	160k

Table 1: Tweet categories, including keywords, and scraped samples

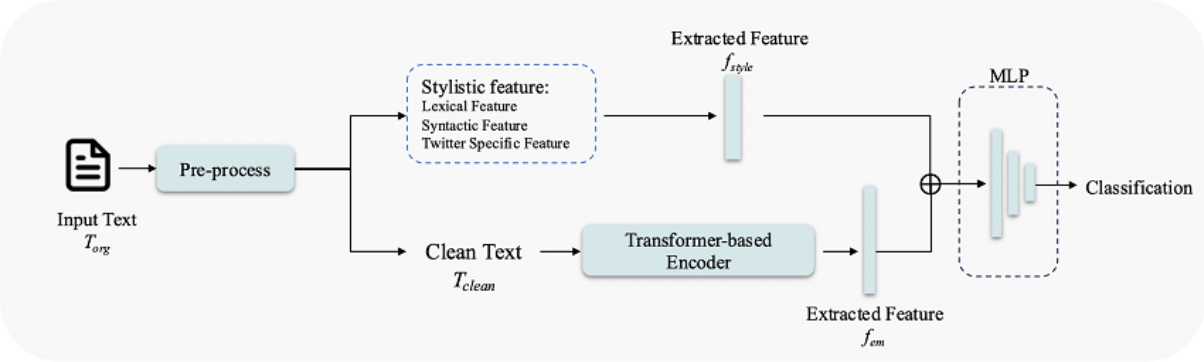

Figure 1: Block diagram of proposed detection method.

tion to this, we also perform lexicographic processing and syntax analysis on the input tweets. The most popular machine learning techniques, such as Bag-of-Words (BoW), N-gram, and TF-IDF help convert text into dictionary-based statistical features. Since we want to extract intrinsic stylistic features of the synthetic tweets, which should be content independent, we choose to use character-level N-gram and part-of-speech (PoS) tag N-gram as part of our self-defined stylistic features. Here we introduce the PoS tag N-gram as a rule-based matching feature. It helps us to extract and classify different writing patterns in phrases. Besides this, we are also curious about the emojis shown in tweets. According to Emojipedia's statistics (Broni, 2022), by 2022, over 22.4% of all tweets now contain at least one emoji, while over half of the comments on Instagram include emojis. Using emojis is an easy and concise way to express emotion and convey meaning. So, it should also be a feature to detect synthetic tweets. Thus, the extracted stylistic features f_{style} can be categorized into 3 types: twitter specific feature, lexical feature, and syntactic feature. Table 2 gives detailed information of the self-defined stylistic feature. Emoji richness refers to the ratio of unique emojis over the total number of emojis used in the text. Vocabulary richness refers to 3 well-defined scores: type-token ratio (TTR) (Chotlos, 1944), mean segmental type-token ratio (MSTTR) (Johnson, 1944), and moving average type-token ratio (MATTR) (Covington and McFall, 2010). Additionally, we use 3 well-defined scores for readability: Flesch reading ease (FRE) formula (Flesch, 1979), Gunning fog index (GFI) (Gunning), and Dale-Chall readability (DCR) score (Dale and Chall, 1948).

Then a contextualized feature f_{em} will be extracted from the cleaned tweet T_{clean} by a transformer-based encoder. Here we choose to use RoBERTa for two reasons, 1) it is a powerful and effective language model which achieves a good performance in variety of NLP sub-tasks, and 2) it use Byte-Pair Encoding (BPE) (Gage, 1994) for text encoding which enables the encoding of any rare words in the vocabulary with appropriate sub-word tokens without introducing any "unknown" tokens. This is important for the twitter posts since they may contain some non-dictionary phrases or abbreviation.

The stylistic feature f_{style} and contextualized feature f_{em} will be concatenated together to form a new feature vector and fed into a Multi-layer Perceptron (MLP) for human-generated and synthetic tweets detection.

During the training time, we will first build and memorize character-level and PoS tag N-grams dictionaries based on the training dataset. And use them to calculate the corresponding feature vectors in training and testing phrases.

5. Experiments

5.1. Synthetic Analysis

We first conduct an experiment to determine how to preprocess the emojis in the text using 3 approaches: remove emojis directly, encode emojis directly and use emoji description instead.

Feature Type	Description	Examples
Twitter specific feature	Statistical features based on Twitter-specific features	Total emoji count, unique emoji count, emoji repeated times, emoji frequency, emoji richness, email count, hashtag count, mention count, hashtag frequency, mention frequency
Lexical feature	Stylistic features based on characters and words	Word length, word count, sentence count, character count, word frequency, digits counts, upper case word count, vocabulary richness, character level N-gram, contractions count, readability
Syntactic feature	Stylistic features based on the organization of sentences	Stop words count, stop word frequency, Special punctuation frequency, proper noun count, noun count, Part-of-Speech (PoS) tag N-gram

Table 2: List of extracted self-defined style features.

We tested the three different pre-processing approaches with RoBERTa on our dataset and found that encoding emojis directly or using emoji description instead can achieve 1.4% accuracy over removing all emojis entirely. The accuracy for encoding emojis directly and using emoji description instead were about the same, with 0.1% difference. Therefore, we directly encoded the emojis for the experiments.

To demonstrate the advantage of our proposed method, we conducted experiments on the synthetic tweets dataset we designed and the TweepFake dataset for comparison. In all experiments, the training, validation, and testing datasets are split in a 6:2:2 ratio. Since in our proposed method we use RoBERTa to extract learned feature, we take the RoBERTa model with a sequence classifier as the baseline for comparison. All experimental results are compared to this baseline to show the effectiveness. Our proposed method is implemented in PyTorch and trained using the Adaptive Moment Estimation (Adam) optimizer with a learning rate of $5e^{-4}$ and weight decay of 0.001. To prevent over-fitting in the training phase, we use two strategies: 1) an early stop execution will take place if 5 successive epochs stop improving on validation loss, and 2) label smoothing is implemented in cross-entropy loss function (Szegedy et al., 2016).

Table 3 shows the results for synthetic tweets detection with different stylistic features on the TweepFake dataset. In this experiment, we isolate the stylistic features into 3 different portions, preliminary statistical features, character-level N-gram and PoS N-gram to check the effectiveness of these features. Here we use RoBERTa fine-tuned on TweepFake dataset as a baseline. Experimental results show that stylistic feature can help to improve the performance of synthetic tweet detection.

Table 4 shows the results for synthetic tweets detection on our generated Synthetic Tweets Dataset. Here the stylistic features are the combination of preliminary statistical features, character-level N-gram and PoS N-gram, i.e., the full stylistic features described in Table 2. Figure 2 and Figure 3 present more detailed evaluations on different topics and different generative models. This indicates that the proposed method will generally improve the performance regardless of the generative models and content.

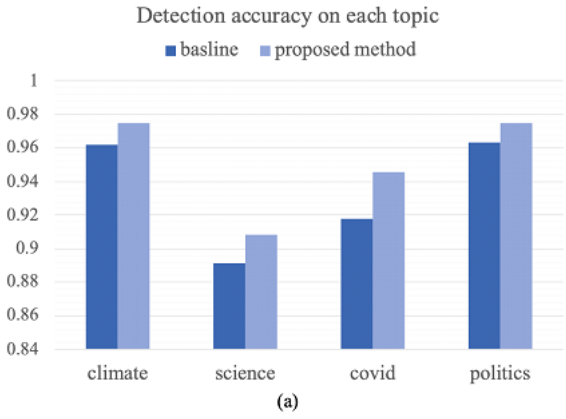

Figure 2: Detection accuracy on different topics.

We also conduct a preliminary experiment for generative model attribution using our method and our generated synthetic tweets dataset. The goal is to determine which generative model is used to create the synthetic tweets. In Table 5, we compared the accuracy score of our method to the baseline. The results indicate that the proposed method shows some improvement in human-generated and GPT3 attribution identifica-

Models	Accuracy	Precision	Recall	F1
RoBERTa (baseline)	0.88398	0.87488	0.90313	0.88878
RoBERTa + prelim	0.92205	0.92293	**0.92564**	0.92428
RoBERTa + prelim + char	0.92257	0.92383	**0.92564**	0.92473
RoBERTa + prelim + char + PoS	**0.92461**	**0.93209**	0.91842	**0.92521**

Table 3: The performance of the proposed method testing on the TweepFake dataset for synthetic tweets detection.

Models	Accuracy	Precision	Recall	F1
RoBERTa (baseline)	0.93432	0.91678	0.95535	0.93567
RoBERTa + stylistic feature	**0.95136**	**0.94155**	**0.96248**	**0.95190**

Table 4: The performance of the proposed method testing on the Generated Synthetic Tweets Dataset for synthetic tweets detection.

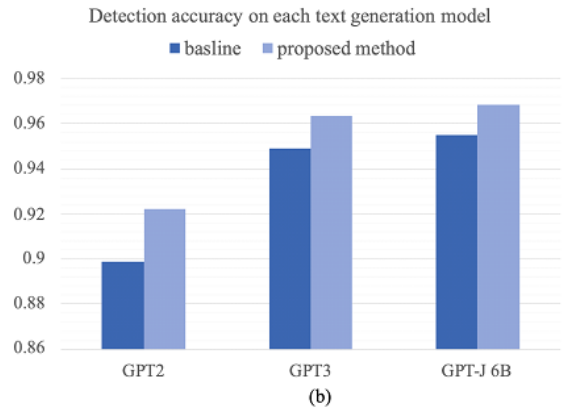

Figure 3: Detection accuracy on different generative models.

tion, but performs lightly worse in GPT2 and GPT-J 6B cases. Overall, the proposed method achieves higher balance accuracy than the baseline.

5.2. Paraphrasing Attacks

In the context of paraphrasing attacks, the unmodified dataset has a relatively high divergence distance between all 4 classes, with the closest being between class 2 (GPT2) and class 3 (GPT3), as shown in Table 6. After the dataset has been paraphrased, the divergence is much lower, meaning that the class distributions are more similar to one another. Even class 0 (human) and class 3 (GPT 3), which had the farthest divergence distance in the unmodified dataset, have far more similar distributions in the paraphrased set, as shown in Table 7

After the creation of the paraphrased dataset, we examine authorship attribution for both datasets and analyze the results against their distributional similarities to find that the two are correlated. Similar to Fabien et al. (2020) we use some traditional techniques, such as Logistic Regression (LR) and a Term Frequency Inverse Document Frequency (TFIDF) representation as well as BERTAA for comparison.

Expectantly, the reduced distributional distance between the unmodified and paraphrased text has had a negative impact on the classifier's ability to detect and attribute the synthetic text. Table 8 shows the result for synthetic tweets detection on the paraphrased dataset, where we treat both the paraphrased and generated tweets as synthetic. The results shown in the "Paraphrased" line are obtained by pre-training on the unmodified tweets and testing on the paraphrased dataset. In this way, the proposed method can still detect inherent features of the synthetic short text after a paraphrasing attack. And the discrepancy in accuracy is due to the increased difficulty of classifying paraphrased samples for human generated tweets, which drops from 96.86% to 88.94%. And one substantial impact introduced by paraphrasing attacks is the increase in difficulty for the model to attribute synthetic tweets to their generation models. The generation model attribution accuracy on this dataset drops from 96.52% to 55.75%.

However, rather than detecting a random sample based on its author or synthetic label, which may or may not be paraphrased, we can instead infer detection using a classifier to detect samples which are machine-paraphrased. Operating under the assumption that a non-malicious twitter user would not machine-paraphrase their tweets, we can show success in detecting which tweets are malicious by detecting paraphrased text.

To achieve this goal, we rephrased our task by treating various paraphrasers as different authors to conduct detection and attribution task. We first conduct an experiment on the generated paraphrased dataset to detect if a given sample is paraphrased or not, regardless of which paraphrase model was responsible for the paraphrasing. The proposed method performs astonishingly

Models	RoBERTa (baseline)	RoBERTa + stylistic feature
Human	0.9599	**0.9668**
GPT2	**0.8865**	0.8835
GPT3	0.9042	**0.9123**
GPT-J 6B	**0.9739**	0.9718
Avg.	0.9336	**0.9419**

Table 5: The balanced accuracy of the proposed method on generated Synthetic Tweets Dataset for generative attribution identification.

JSD	Class 0	Class 1	Class 2	Class 3
Class 0	0	-	-	-
Class 1	0.0568	0	-	-
Class 2	0.0666	0.0221	0	-
Class 3	0.0751	0.0325	0.0198	0

Table 6: Jensen-Shannon Distance on the unmodified text categories

JSD	Class 0	Class 1	Class 2	Class 3
Class 0	0	-	-	-
Class 1	0.0167	0	-	-
Class 2	0.0271	0.0157	0	-
Class 3	0.0384	0.0271	0.0165	0

Table 7: Jensen-Shannon Distance on the paraphrased text categories

well in the binary classification case. Comparing the human-generated tweets to each of the paraphrased texts individually yielded a near-perfect accuracy, as shown in Table 9. Additionally, after rephrasing the task to that of authorship attribution where each of the 3 paraphrasers and non-paraphrasing samples are treated as authors, our model achieves an accuracy of 91.71%.

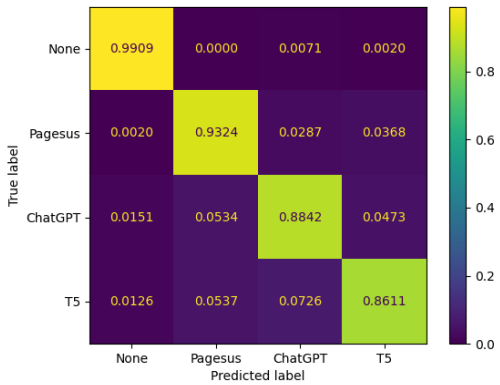

Figure 4: The confusion matrix of the paraphraser attribution task.

	LR	LR+TFIDF	BERTAA	Proposed
Unmodified	0.337	0.798	0.944	**0.977**
Paraphrased	0.356	0.589	0.807	**0.941**

Table 8: Balanced accuracy for detecting synthetic tweets on unmodified tweets and paraphrased tweets. Here the unmodified tweets are generated based on human, GPT2, GPT3, GPT-J 6B generation models.

None	PEGASUS	ChatGPT	T5	Avg.
0.9767	0.9990	0.9668	0.9874	0.9824

Table 9: Balanced accuracy for detecting paraphrasing on the paraphrased dataset. The proposed model is trained on the entire dataset and detecting accuracy of each paraphrase model is calculated. Here none stands for tweets without any paraphrasing.

6. Conclusion and Discussion

In this paper, we created a dataset of synthetic tweets for 4 different topics utilizing 3 popular LLMs (GPT2, GPT3, and GPT-J 6B). We also introduced an method, which exploits the efficiency of certain stylistic features combined with popular LLM models. We validated this method by pretraining the model on a public dataset (TweepFake) and our generated datasets. The pre-trained models perform well on both the synthetic text detection and generative model attribution tasks. We also created another paraphrased dataset based on a subset of the one mentioned above using 3 different paraphrasing models. The experiment results indicate that the proposed model can detect inherent features of the synthetic text but eliminate the gaps between different generative models under paraphrasing attacks. The proposed method also works well on paraphrasing detection and paraphrase model attribution task.

Future work in this area will explore improved feature integration in a zero-shot setting, in order to detect synthetic tweets generated by unknown LLMs. Additionally, further experimentation will be conducted regarding the generative model attribution task, to evaluate how malicious activity such as watermark spoofing and paraphrasing at-

tacks effect classifier accuracy and to improve defenses against these activities. As LLMs expand and open-source tools continue to be built which utilizing them, the tasks of detecting and attributing synthetically generated text will continue to be important.

7. Acknowledgements

This material is partially based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government. Address all correspondence to Edward J. Delp, ace@purdue.edu.

8. References

Keith Broni. 2022. Global emoji use reaches new heights. https://blog.emojipedia.org/global-emoji-use-reaches-new-heights/. Accessed on Jun 17th, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Proceedings of the Advances in Neural Information Processing Systems*, 33:1877–1901.

John W Chotlos. 1944. Iv. a statistical and comparative analysis of individual written language samples. *Psychological Monographs*, 56(2):75.

Michael A Covington and Joe D McFall. 2010. Cutting the gordian knot: The moving-average type–token ratio (mattr). *Journal of quantitative linguistics*, 17(2):94–100.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

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*.

D.M. Endres and J.E. Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860.

Maël Fabien, Esaú Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. Bertaa: Bert fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing*, pages 127–137.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS One*, 16(5):e0251415.

Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Communications of the ACM*, 59(7):96–104.

Rudolf Flesch. 1979. *How to write plain English: A book for lawyers and consumers*, volume 76026225. Harper & Row New York.

Philip Gage. 1994. A new algorithm for data compression. *the C Users Journal*, 12(2):23–38.

Yuriy Gorodnichenko, Tho Pham, and Oleksandr Talavera. 2021. Social media, sentiment and public opinions: Evidence from# brexit and# uselection. *European Economic Review*, 136:103772.

Robert Gunning. The technique of clear writing. *(No Title)*.

Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews–an adversarial study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 351–356.

Wendell Johnson. 1944. Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *the Annals of Mathematical Statistics*, 22(1):79–86.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.

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*.

Fernanda López-Escobedo, Carlos-Francisco Méndez-Cruz, Gerardo Sierra, and Julián Solórzano-Soto. 2013. Analysis of stylometric variables in long and short texts. *Procedia-Social and Behavioral Sciences*, 95:604–611.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *American Society for Information Science and Technology*, 60(3):538–556.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Chris J Vargo, Lei Guo, and Michelle A Amazeen. 2018. The agenda-setting power of fake news: A big data analysis of the online media landscape from 2014 to 2016. *New Media & Society*, 20(5):2028–2049.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Proceedings of the Advances in Neural Information Processing Systems*, 30.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax. Accessed on Jun 18th, 2023.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Proceedings of the Advances in Neural Information Processing Systems*, 32.