Through the Looking Glass: Learning to Attribute Synthetic Text Generated by Language Models

Shaoor Munir[†] Brishna Batool[†] Zubair Shafiq[‡] Padmini Srinivasan[*] Fareed Zaffar[†]

[†]Lahore University of Management Sciences, Pakistan
shaoor.munir@lums.edu.pk
[‡]University of California at Davis, USA
zubair@ucdavis.edu
[*]University of Iowa, USA
padmini-srinivasan@uiowa.edu

Abstract

Given the potential misuse of recent advances in synthetic text generation by language models (LMs), it is important to have the capacity to attribute authorship of synthetic text. While stylometric organic (i.e., human written) authorship attribution has been quite successful, it is unclear whether similar approaches can be used to attribute a synthetic text to its source LM. We address this question with the key insight that synthetic texts carry subtle distinguishing marks inherited from their source LM and that these marks can be leveraged by machine learning (ML) algorithms for attribution. We propose and test several ML-based attribution methods. Our best attributor built using a fine-tuned version of XLNet (XLNet-FT) consistently achieves excellent accuracy scores (91% to near perfect 98%) in terms of attributing the parent pre-trained LM behind a synthetic text. Our experiments show promising results across a range of experiments where the synthetic text may be generated using pre-trained LMs, fine-tuned LMs, or by varying text generation parameters.

1 Introduction

Recent advancements in natural language processing have enabled *synthetic* text generation that is often of comparable quality to the organic text (Ippolito et al., 2020; Radford et al., 2019; Zellers et al., 2019; Gehrmann et al., 2019). This capability has the potential to be misused by malicious actors to launch misinformation, spam, and phishing campaigns (Solaiman et al., 2019; Brown et al., 2020). To prevent potential misuse, prior research has shown considerable success in building machine learning (ML) algorithms that detect (Zellers et al., 2019) or assist humans in detecting (Gehrmann et al., 2019) synthetic text.

While prior research has shown promise in distinguishing between synthetic and organic text, very little has been done on attributing the authorship of the language model (LM) generating the synthetic text (Pan et al., 2020). It is important to be able to track the provenance of synthetic text to the source LM. This can be useful in identifying perpetrators of potential misuse and the unauthorized use of an LM (e.g., in case it is stolen through sophisticated model inversion attacks (Fredrikson et al., 2015) or outright security breaches).

It is particularly challenging to attribute the authorship of the synthetic texts because of the variety and number of available LMs and configurations. While there are only a handful of public pre-trained LMs, it is common to further fine-tune them before using them to generate synthetic text (Devlin et al., 2019; Sanh et al., 2020). Fine-tuning can significantly impact the characteristics of the generated text (Howard and Ruder, 2018; Cruz and Cheng, 2019). Moreover, variations in the sampling parameters used while generating synthetic text whether from pre-trained or fine-tuned LMs can further impact text characteristics (Zellers et al., 2019).

In this paper, we design and evaluate ML-based techniques for attributing the LM and configuration used to generate a synthetic text. We do this in the context of four problem scenarios, each representing a variation of a threat posed by an adversary or malicious user. The scenarios vary in terms of what information the LM attribution system has about the adversary's strategy for generating fake text.

Methodologically, our key insight for attributing the LM used by the adversary is that differences between LM architecture (i.e., layers, parameters), training (i.e., pre-training and fine-tuning), and generation techniques (i.e., sampling parameters) will leave their subtle mark on the generated synthetic texts. The success of our attributors at identifying the LM and configuration used relies on the presence of these subtle distinguishing marks and on the ability to exploit them effectively. As our re-

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1811–1822
April 19 - 23, 2021.

sults indicate, this success holds especially in terms of attributing pre-trained models used to generate text even under varying conditions.

In summary, our key contributions are:

- We evaluate a variety of attribution techniques on their ability to attribute the LM and configuration used to generate text. These include attributors making use of stylometric features as well as static and dynamic embeddings.
- We evaluate these attributors on a corpus of 350,000 synthetic texts that we generated in a controlled manner using combinations of LMs, sampling parameters, and fine-tuning.
- Our best attributor built on top of a fine-tuned version of XLNet (XLNet-FT) performs excellently at identifying pre-trained LM used to generate coherent synthetic texts. Accuracy ranges between 91% and close to perfect 98%. This performance holds for various experiments where we use fine-tuning and different sampling parameters. However, the performance is mediocre when attributing the fine-tuned LM used to generate the text.

Paper Organization: The rest of the paper is organized as follows. Section 2 presents the different threat models based on the adversary's strategy for generating synthetic text and assumptions made by the attributor. We then describe our data and attribution methods in Section 3. Experimental results are in Section 4. Section 5 contextualizes our work with respect to prior literature. Section 6 concludes the paper with an outlook on future work.

2 Threat Model

This section describes different threat models that we consider in this paper. The adversary's goal is to generate synthetic text using language models (LMs). The attributor's goal is to attribute the synthetic text to the source LM used by the adversary. All of the threat models operate under the closed world scenario, where the attributor is assumed to know the universe of LMs. The threat models differ based on the adversary's LM training (i.e., pre-training or fine-tuning) and sampling strategies.

2.1 Attributing pre-trained LMs

In the first scenario, the adversary uses a pre-trained LM to generate synthetic text. The attributor trains a classifier to attribute the synthetic text to the source pre-trained LM. We assume a closed-world scenario where both the adversary and attributor have access to the set of off-the-shelf pre-trained LMs such as GPT-2.[1]

More formally, the scenario can be described as: *Given n pre-trained LMs $PM_1, PM_2, ..., PM_n$, the goal is to train a n-class attributor to attribute test instances to the correct source pre-trained LM. In this scenario, the adversary generates texts using PM_k where $1 \leq k \leq n$ and the attributor's goal is to predict label PM_k for the generated texts.*

2.2 Attributing fine-tuned LMs to parent pre-trained LMs

In this scenario, the adversary fine-tunes a pre-trained LM to generate synthetic text. The attributor trains a classifier to attribute the synthetic text to the source pre-trained LM. The main difference from the first scenario is that the attributor is unaware of the fine-tuning used by the adversary before generating text. Note that the goal of the attributor is to detect the source pre-trained LM rather than the fine-tuned LM that is used to generate synthetic text.

More formally, the scenario can be described as: *Given n pre-trained LMs $PM_1, PM_2, ..., PM_n$, and a LM FM_k, generated by fine-tuning PM_K where $1 \leq k \leq n$, the goal is to train a n-class attributor to attribute test instances to the correct source pre-trained LM. In this scenario, the adversary generates text using fine-tuned LM FM_k and the attributor's goal is to predict label PM_k for generated text.*

2.3 Attributing pre-trained or fine-tuned LMs with different sampling parameters

In this scenario, the attributor trains a classifier to attribute the synthetic text generated by the adversary using a pre-trained or fine-tuned LM. The main difference from the first scenario is that the adversary potentially uses different sampling parameters for text generation than those used by the attributor to train the classifier.

More formally, the scenario can be described as: *Given n pre-trained or fine-tuned LMs $M_1, M_2, ..., M_n$, the goal is to train a n-class attributor to attribute test instances to the correct source model. As per this scenario the adversary*

[1] This assumption holds for the rest of the paper, unless stated otherwise.

generates texts using model M_k, $1 \leq k \leq n$, with sampling parameters S_k that are unknown to the attributor, and the attributor's goal is to predict label M_k for the generated text.

2.4 Attributing fine-tuned variants of a pre-trained LM

In this scenario, the adversary fine-tunes a pre-trained LM to generate synthetic text. The attributor trains a classifier to attribute the synthetic text to the source fine-tuned LM. The main difference as compared to the second scenario is that the attributor is aware of the fine-tuning used by the adversary. Note that there are multiple fine-tuned variants of the same parent pre-trained LM.

More formally, the scenario can be described as: *Given n fine-tuned LMs $FM_1, FM_2, ..., FM_n$, the goal is to train a n-class attributor to attribute test instances to the correct fine-tuned LM. As per this scenario, the adversary generates text using a fine-tuned LM FM_k and the attributor's goal is to predict label FM_k for the generated text.*

3 Data & Methods

In this section, we present details about (1) the text generating language models (LMs) and their configurations, and (2) about the attributors studied. To address our research goals, we need a dataset of synthetic texts generated by various pre-trained and fine-tuned LMs under different configurations. Publicly available datasets are unsuitable because there can be high variability in the conditions under which text was generated [2]. It is crucial for us to be able to control the underlying conditions such as: the architecture of LM, prompt used for text generation, sampling parameters, and the data size and topics used for fine-tuning. Details about this generated dataset are also provided in this section.

3.1 Text Generation: LMs, parameters, and configurations

We used four pre-trained LMs: OpenAI GPT (Radford et al., 2018), OpenAI GPT2 (Radford et al., 2019), XLNet (Yang et al., 2019), and BART (Lewis et al., 2020). BART and XLNet are both based on the BERT architecture, which makes use of the bidirectional context of input text to develop a deep understanding of language. XLNet improves on BERT with a form of generalized auto-regressive pre-training using permutation modeling. It outperforms BERT on several classification tasks (Yang et al., 2019). BART combines the bidirectional encoder used by BERT with an auto-regressive decoder used by GPT, which, through a noising and text reconstruction pre-training task, achieves good performance in both language understanding and language generation tasks. In other words, both BART and XLNet augment their training strategies to make up for the lack of language generation capabilities in BERT. GPT and GPT2 are architecturally identical LMs with GPT2 trained on 10 times the data used for training original GPT LM. These use a more traditional generative pre-training approach, looking only at the context coming before a part of the text and not after (Radford et al., 2019). All four pre-trained LMs are publicly available.

3.1.1 Text generation parameters

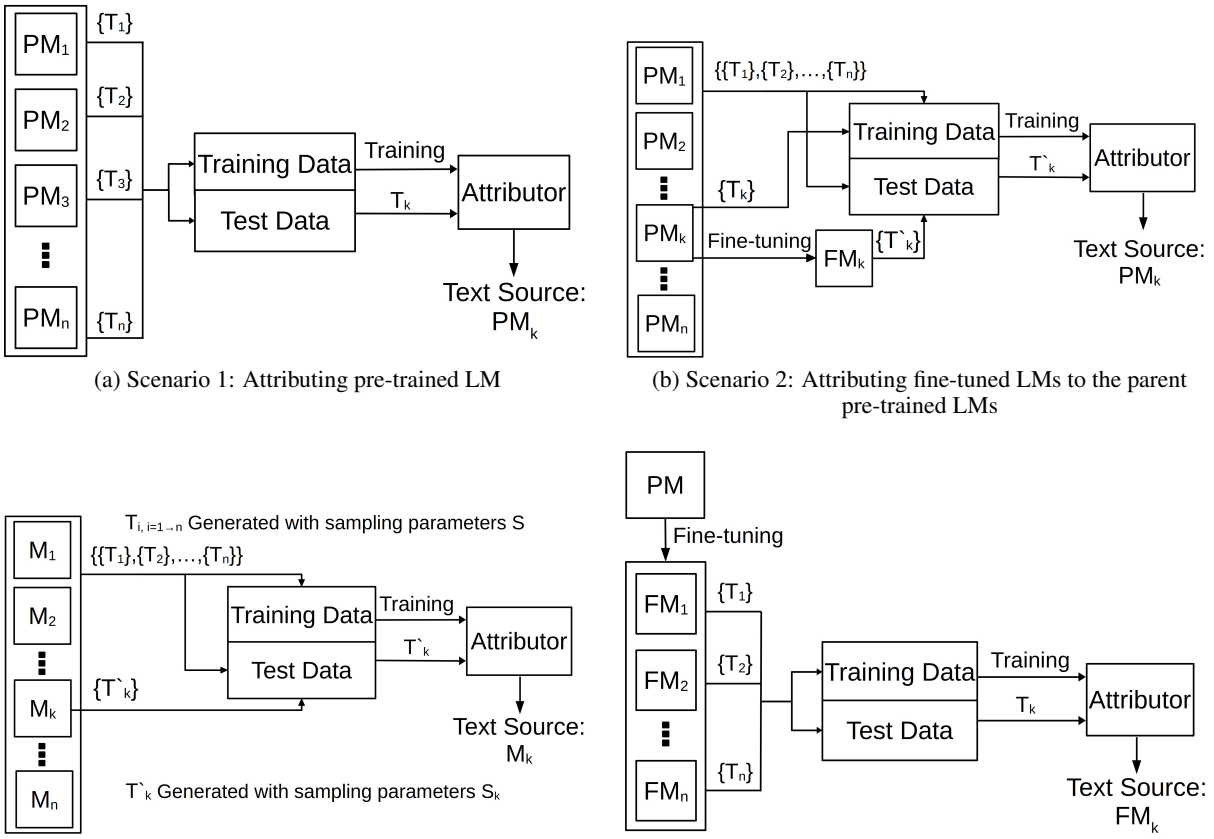
Three key parameters when generating texts are: *p*, *k*, and *temperature*. The range of values tested are given in Table 1, with default values emphasized in boldface. Note that one chooses to use either *p-value* or *k-value* sampling since they have the same goal - controlling the number of words taken into consideration while sampling text from an LM.

With *top-k* sampling, the LM randomly chooses one from the top *k* words. With *top-p* sampling, it chooses from the set of words whose cumulative probability exceeds *p*. Both Zellers et al. (2019) and Holtzman et al. (2020) conclude that synthetic text matches organic text closely when the *p-value* is kept in range [0.9, 1.0]. Higher values lead to repetitions as the length of text increases. Thus, we choose the lower limit of *p* from the range [0.9, 1.0].

For *top-k* sampling, we use a range of values both higher and lower than 40, which is used as the default for text generation by Radford et al. (2019) in their breakthrough GPT2 paper. Between a choice of *top-p* or *top-k* sampling, we chose *top-p* (p = 0.9) as default due to its lower dependency on vocabulary size and extensive use in previous research on GPT2 (Radford et al., 2019; Zellers et al., 2019; Ippolito et al., 2020).

Temperature controls the likelihood of low probability words appearing in the final pool of words used for random selection (Holtzman et al., 2020). Higher temperatures produce text containing highly unusual words that are normally not favored by *top-k* or *top-p* sampling. At the other end, Holtzman et al. (2020) note that temperatures below 1 reduce

[2] https://www.kaggle.com/abhishek/gpt2-output-data



(a) Scenario 1: Attributing pre-trained LM

(b) Scenario 2: Attributing fine-tuned LMs to the parent pre-trained LMs

(c) Scenario 3: Attributing LM with different sampling parameters

(d) Scenario 4: Attributing fine-tuned variants of a pre-trained LM

Figure 1: Illustration of different threat models studied in this work

word diversity but at the cost of increasing word repetition. To avoid this we set temperature as 1 in experiments where evaluation of its effect on synthetic text is not of concern.

#	Parameter	Values
1	Architecture	GPT, **GPT2**, XLNet, BART
2	Text Length	Short, **Medium**, Long
3	Fine-tuning Topic	**r/changemyview**, r/technology, r/relationships, r/conspiracy
4	p-value	**0.9**, 0.92, 0.94, 0.96, 0.98, 1.0
5	Temperature	0.1, 0.5, **1**, 1.5, 2
6	k-value	1, 20, 40, 80, 160

Table 1: The parameters explored (defaults in bold)

3.1.2 Data for fine-tuning

For scenarios where we need synthetic text generated using fine-tuned LMs, we limit text generation to the GPT2 LM. GPT2 has been shown to have state of the art performance in language generation tasks (Radford et al., 2019; Klein and Nabi, 2019). Data from four Reddit communities was used for fine-tuning LMs: *r/relationships*, *r/technology*, *r/changemyview*, and *r/conspiracy*. These subreddits were chosen based on qualitative differences between their content. r/technology contains technical jargon, while r/relationships focuses on personal pronouns and adopts a critical approach towards writing. r/changemyview has confrontational content with members attempting to challenge and disprove each other's views, while r/conspiracy focuses on hyperbolic statements. In essence, each subreddit is considered a different *topic* area. Table 2 shows the number of posts and comments scraped from each subreddit.

3.1.3 Dataset details

We generate text of three different lengths: *short* (up to 40 words), *medium* (between 40 and 100 words), and *long* (above 100 words). In experiments where length is not the focus, we use *medium* as the default. Each synthetic text is generated using a randomly selected subreddit submission as a prompt. We start by sampling words equal to the length of the prompt from the LM. We trim the generated text to follow standard sentence structure such as start capitalization and end punctuation, after which text is sorted into one of three length

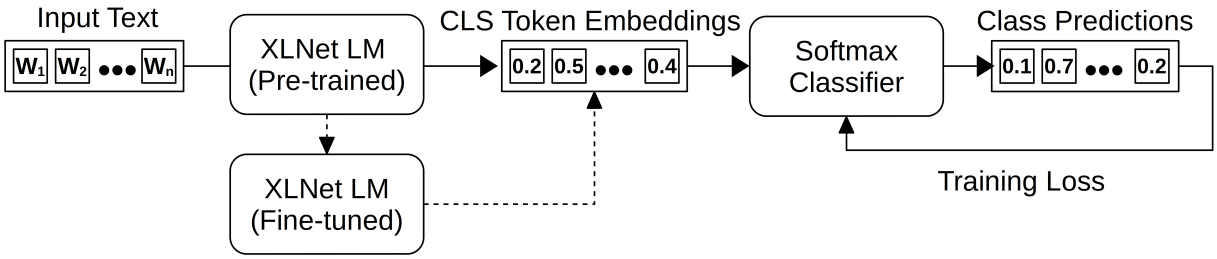


Figure 2: Attributor training on XLNet embeddings. The dashed lines are part of the fine-tuned pathway.

categories.

We generated 10,000 synthetic texts for each target class in our experiments. For example, when evaluating the performance of attributors against fine-tuned LMs, we generated 10,000 samples for each of four GPT2 LMs fine-tuned on one of the four Reddit topics mentioned previously. In total, 35 distinct sets of synthetic documents, each with 10,000 examples, were used for a total of 350,000 unique synthetic documents[3]. We build training and test datasets that are balanced in classes for each scenario because while there is growing evidence that synthetic text is appearing in the wild, there is little to no information about the relative impact of the source LMs. Thus, any split beyond an even split across classes has little justification.

Subreddit	Posts	Comments	Total
r/changemyview	136,775	321,527	458,302
r/relationships	200,047	167,219	367,266
r/technology	174,431	143,199	317,630
r/conspiracy	99,302	161,993	261,295

Table 2: Data scraped per each subreddit.

3.2 Attributors

We test six attributors in their ability to identify the source LMs. The first attributor is a decision tree classifier with Writeprints (Abbasi and Chen, 2008) feature set, second is a CNN with GloVe (Pennington et al., 2014) embeddings as the feature set. The next four attributors are softmax classifiers, with the first two built on top of pre-trained XLNet and GPT2, and the other two on top of XLNet and GPT2 fine-tuned on training data used in the corresponding scenario.

3.2.1 Decision tree with Writeprints features

The Writeprints features have been used extensively and successfully for authorship attribution. (Abbasi and Chen, 2008; Mahmood et al., 2020) When combined with SVMs and decision trees these have shown good performance in attribution tasks (Abbasi and Chen, 2008; Pearl and Steyvers, 2012). Due to ease of interpretability of features, we implement a decision tree classifier with Writeprints features to test our intuition that stylistic, rather than topical, differences contribute towards the attribution of synthetic text.

3.2.2 CNN with GloVe embeddings

Pre-trained GloVe embeddings have been shown to outperform word frequency and count-based embeddings for sentence and sequence classification tasks (Pennington et al., 2014; Le-Hong and Le, 2018). Also, the use of GloVe with CNNs has shown good results in classification tasks like newsgroup identification (Gupta et al., 2018).

3.2.3 Attributors from LM embeddings

Embeddings generated through LMs like BERT, XLNet, and GPT2 have been shown to capture language semantics and context much better than static embeddings generated through GloVe and other word count or frequency-based embeddings generators (Sun et al., 2020; Howard and Ruder, 2018). Because of their extensive pre-training, these LMs can capture long-term dependencies and incorporate contextual and hierarchical relations between words better than pre-computed static embeddings.

LMs such as XLNet make use of a special *[CLS]* token to get pooled output representing a complete text sequence. We use the final network layer embeddings of this token for attribution. Specifically, we train a softmax output layer that takes as input XLNet's *[CLS]* token embeddings and generates probabilities for each decision class in the experiment setup. For GPT2 we use a parallel strategy with pooled output from the complete final layer of the model for a particular input text. Again this output is connected to a softmax output layer which, similar to our strategy with XLNet, is trained to

[3]We will make this dataset available for research upon publication of our paper

generate class predictions based on the input embeddings.

In addition to using the pre-trained versions of XLNet and GPT2, we also evaluate attributors built from fine-tuned versions of these LMs. Note that here fine-tuning is on training data used to train all attributors in the corresponding experiment. Figure 2 illustrates these strategies with XLNet as an example. The sequence with dashed lines represents the fine-tuned versions.

4 Results

We present attribution accuracy results in the same order of scenarios described earlier in Section 2.[4]

4.1 Attributing pre-trained LMs

Table 3 presents the accuracy results for short (up to 40 words), medium (between 40 and 100 words), and long (more than 100 words) synthetic text generated using pre-trained GPT, GPT2, XLNet, and BART language models (LMs). Decision tree and the two XLNet versions achieve accuracy between 82 and, near perfect 98%, across the three types of texts. In comparison, CNN and GPT2 attributors lag behind.

While both XLNet attributors score higher than decision tree, XLNet-FT has the best performance which when compared with the next best XLNet-PT ranges from 3% to 7%. Note that apart from the pre-trained GPT2 attributor, all show marked improvement in accuracy scores with an increase in text length. Similar results showing direct proportionality of classifier performance with text length were also observed by Ippolito et al. (2020) in experiments detecting synthetic text.

Prior work has shown that uni-directional LMs are more suited for language generation due to generative pre-training (Lewis et al., 2020) where the LM learns to predict the next word based on the previous context. Bidirectional LMs like BERT and XLNet excel at classification as they make use of masked modeling and next sentence prediction tasks to improve understanding of necessary language attributes (Devlin et al., 2019; Yang et al., 2019). Our results are consistent in that XLNet performance is better than GPT2.

Interestingly, the decision tree with Writeprints outperforms GPT2 based attributors in all three text lengths. Our investigation into specific Writeprints features emphasized by the decision tree (**see appendix A.1**) reveals a greater emphasis on stylistic features. This gives further credence to our intuition that variations between texts generated by different LMs are more stylistic than topical in nature. Our results suggest that GPT2 based attributors are not adept at capturing such stylistic differences.

4.2 Attributing fine-tuned LMs to the parent pre-trained LMs

Our goal in this scenario is to attribute the synthetic text generated using a fine-tuned variant (using an unknown dataset) of a pre-trained LM. Note that the attributor is unaware of fine-tuning. We limit fine-tuned text generation in this experiment to just GPT2 for reasons described in Section 3.

The first row in Table 4 reports the accuracy results. We note that XLNet-FT again performs the best with XLNet-PT in the second place. CNN has the weakest results. Interestingly, Writeprints continues to do fairly well – once again emphasizing the role of style in identifying source LM. Comparing these results with Table 3 (for medium length texts), we see all accuracies drop slightly as expected when the adversary chooses to fine-tune the LM that is unknown to the attributor.

We run a second variation of the same experiment – one where the attributor has partial knowledge of the adversary's strategy. Specifically, the fact that the adversary is using a fine-tuned LM to generate text is known but the dataset used for fine-tuning remains unknown. In response, we pick some dataset (here r/relationships) to add fine-tuned LM generated texts to our training data. Note that the adversary uses r/changemyview. The second row in Table 4 reports similar results as the first row. Thus, it seems that this additional knowledge does not help improve attribution accuracy.

In sum, the accuracy for XLNet-FT across all experiments thus far is above 90%. This indicates that even when the adversary fine-tunes the LM for text generation, the parent pre-trained LM is still identifiable. This result confirms our intuition that as fine-tuning is known to leave the majority of layers unchanged, the text generated retains characteristics of the parent pre-trained LM, making accurate attribution possible.

[4] We measured performance using F1 score as well. However, since there were no remarkable differences, we only report accuracy results to be concise.

Synthetic Text Length	DT (Writeprints)	CNN (GloVe)	XLNet-PT	XLNet-FT	GPT2-PT	GPT2-FT
Short (Upto 40 Words)	82	68	85	**91**	72	74
Medium (40 to 100 words)	86	73	90	**96**	71	72
Long (Above 100 words)	93	83	95	**98**	72	72

Table 3: Accuracy percentages for attributing source pre-trained LMs. Datasets contain synthetic texts of different lengths generated using pre-trained BERT, GPT, GPT2 and XLNet.

Training data includes	Test data includes	DT (Writeprints)	CNN (GloVe)	XLNet-PT	XLNet-FT	GPT2-PT	GPT2-FT
GPT2 (PT)	GPT2 (FT-r/changemyview)	78	64	86	**93**	70	71
GPT2 (FT-r/relationships)	GPT2 (FT-r/changemyview)	77	67	86	**91**	70	70

Table 4: Accuracy percentages in attributing source pre-trained LM when adversary generates synthetic text using a fine-tuned LM. In addition to GPT2 variants mentioned in columns 1 and 2, training and testing data also includes classes representing XLNet, BART, and GPT.

4.3 Attributing LM with different sampling parameters

Here we consider the scenario where both the attributor and the adversary use the same LM but they differ in parameter choices when generating texts. We run this experiment assuming the adversary uses GPT2 fine-tuned on r/changemyview. The attributor is aware of this but not the sampling parameters. Selecting parameter values that are quite different from each other we see from Table 5 that there is virtually no performance drop for XLNet-FT. That is, our best performing attributor is resilient to these differences. Temperature sampling shows weaker results all around. This is not a concern as discussed later in this section.

We next explore the parameter differences angle further to get a sense of what would happen if the adversary chose a parameter value other than the ones explored in Table 5. The different values for k, p, and temperature are as listed in Table 1. We remind the reader that one uses either $top-k$ or $top-p$ sampling to control the number of words under consideration during text generation. We use $top-p$ sampling as the default strategy. When varying k or p, the temperature is fixed at the default value. When varying temperature, p is kept at the default value.

The results in Table 6 show that it is challenging to tell apart synthetic texts generated by different values of k and p. Given their strong similarities we expect to see results as in Table 5 when the adversary picks other parameter values. With temperature variations we get accuracy above 80%, indicating marked differences between texts generated at different temperatures. However, taking a closer look at the text reveals a serious problem: temperature > 1 produces erratic and confusing text. This problem becomes more acute as the temperature approaches its upper limit. This is consistent with the observation by Holtzman et al. (2020) showing that temperatures above 1 produce incoherent and confusing text. This reduces viability in a setting where the synthetic text is to be used as a suitable replacement for organic text.

We conclude from Tables 5 and 6 that our attributors should be resilient even when the adversary chooses parameter values for text generation beyond the ones explicitly tested here.

4.4 Attributing fine-tuned variants of a pre-trained LM

Finally, we explore the scenario where the adversary is using different fine-tuned LMs with the same parent pre-trained LM. The attributor is aware of this fine-tuning and is attempting to tell apart these fine-tuned LMs. Table 7 presents the accuracy results when synthetic text is generated by fine-tuning GPT2 on 4 different subreddits (r/changemyview, r/technology, r/relationships, r/conspiracy). XLNet-FT again achieved the best accuracy, however, this time it is less than 60%. Curiously, CNN which was the least successful in earlier experiments performed almost identically to XLNet-FT. GPT2 performed just slightly better than a random attributor (1/4, i.e., 25%). Overall, variations between texts generated by different fine-tuned variants of the same pre-trained LM are not

Training data includes	Test data includes	DT (Writeprints)	CNN (GloVe)	XLNet-PT	XLNet-FT	GPT2-PT	GPT2-FT
GPT2 (k=20)	GPT2 (k=160)	76	71	88	**95**	71	71
GPT2 (p=0.9)	GPT2 (p=1.0)	80	70	88	**96**	70	70
GPT2 (t=0.1)	GPT2 (t=1.0)	60	67	70	**77**	71	72

Table 5: Accuracy percentages for attributing source LM when adversary generates synthetic text using different sampling parameters. In addition to GPT2 variants mentioned in columns 1 and 2, training and testing data also includes classes representing XLNet, BART, and GPT.

Attributor	k-value	p-value	Temperature
DT (Writeprints)	**53**	**43**	82
CNN (GloVe)	47	26	79
XLNet-PT	42	22	81
XLNet-FT	45	25	**86**
GPT2-PT	28	17	44
GPT2-FT	28	18	44

Table 6: Accuracy percentages for identifying texts generated by GPT2 LM fine-tuned on r/changemyview with varying sampling parameters. Parameter values tested are as reported in Table 1.

Attributor	Accuracy
DT (Writeprints)	52
CNN (GloVe)	56
XLNet-PT	53
XLNet-FT	**57**
GPT2-PT	29
GPT2-FT	29

Table 7: Accuracy percentage in attributing fine-tuned GPT2 LMs. Dataset contains texts generated by four GPT2 LMs fine-tuned on each subreddit in Table 2.

pronounced enough to be leveraged by the attribution techniques we consider. Our preliminary analysis shows that there is some correlation between the attributor's mistakes and the vocabulary similarity of the corresponding subreddits. However, further research is needed to probe the causes of this lackluster performance and devise ways to improve the attribution of text produced by fine-tuned LMs.

5 Related Work

We contextualize our work with respect to prior literature on detection and attribution of organic and synthetic text.

5.1 Synthetic text detection

There has been a lot of recent interest in developing ML approaches to distinguish between organic and synthetic text. GLTR (Giant Language Model Test Room) leveraged the statistical tendency of LMs to produce words with higher probability of occurrence to help users differentiate between synthetic and organic text (Gehrmann et al., 2019). Grover used a purpose-built LM to train a classifier for synthetic and organic text (Zellers et al., 2019). Bakhtin et al. (2019) proposed energy based models for differentiating between synthetic and organic text. Our work takes this line of work a step further by trying to attribute synthetic text to the source LM.

5.2 Synthetic text attribution

Pan et al. (2020) proposed a dynamic embedding based approach to attribute synthetic text generated by a pre-trained LM as part of their broader investigation of sensitive information exposed by LMs. We significantly build on this work from both the methodological and application perspectives. Differently from this work, we use stylometric as well as static and dynamic embeddings. We also consider more realistic threat models where the synthetic text is generated by either pre-trained or fine-tuned LMs and using different sampling parameters.

5.3 Organic text attribution

There is a rich body of literature on authorship attribution of organic text using stylometric features. We discuss a few classic papers here. Mosteller and Wallace (1964) used word frequency analysis for authorship attribution. Abbasi and Chen (2008) proposed a ML-based approach for authorship attribution using an exhaustive stylometric feature set called Writeprints. While there is impressive progress in stylometric organic text attribution (e.g., Narayanan et al., 2012; Ruder et al., 2016), these approaches do not work as effectively for synthetic text attribution. As our evaluation showed, Writeprints were significantly outperformed by other approaches for synthetic text attribution. This is because LMs are trained on large text corpora

from different authors thus there are no clear-cut stylometric differences in synthetic text generated by different LMs.

5.4 Synthetic image attribution

Recent advances in Generative Adversarial Networks (GANs) have led to impressive results in synthetic image generation (Bao et al., 2017; Taigman et al., 2017; Ma et al., 2017). For example, Chen et al. (2020) proposed image models similar to pre-trained LMs to learn an unsupervised representation of images for various downstream tasks. Most related to our work, Yu et al. (2019) proposed an ML approach to attribute synthetic images generated by GANs with different architectures and parameters. At the most basic level, the problem of synthetic image attribution differs from synthetic text attribution because images are smooth and local where words in a text document may be correlated even if they are far apart (Sharir et al., 2020). For instance, Yu et al. (2019) showed that their ML classifier could use only part of the synthetic image for attribution. In contrast, we observed a large drop in accuracy when we make use of only part of input synthetic text.

6 Conclusion

In this paper, we presented an ML approach to attribute authorship of synthetic text to its source LM. Our results showed that an attributor based on fine-tuned XLNet embeddings outperformed other approaches based on stylometric features as well as static and dynamic embeddings. Our results also showed there is significant room for improvement in distinguishing between synthetic text generated by different fine-tuned variants of an LM. Further research is also needed for effective attribution of synthetic text generated by more diverse fine-tuned LMs in both closed-world and-open world settings. Finally, future research on synthetic text attribution should also consider more sophisticated LMs (e.g., GPT-3 with 175 billion parameters (Brown et al., 2020) and Google's trillion parameter LM (Fedus et al., 2021)) when they are publicly released.

References

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26(2).

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or Fake? Learning to Discriminate Machine from Human Generated Text. *CoRR*, abs/1906.03351.

J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. 2017. Cvae-gan: Fine-grained image generation through asymmetric training. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2764–2773.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR.

Jan Christian Blaise Cruz and Charibeth Cheng. 2019. Evaluating language model finetuning techniques for low-resource languages. *arXiv preprint arXiv:1907.00409*.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv 2101.03961*.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA. Association for Computing Machinery.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. In *57th Annual Meeting of the Association for Computational Linguistics (ACL): Demo Track*.

Varun Gupta, Akhilesh Kumar, and Aashish Bhardwaj. 2018. Newsgroup classification using cnn and glove embeddings. *International Journal of Applied Research on Information Technology and Computing*, 9:135.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Tassilo Klein and Moin Nabi. 2019. Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds. *arXiv*, 1911.02365.

Phuong Le-Hong and Anh-Cuong Le. 2018. A comparative study of neural network models for sentence classification. In *5th NAFOSTED Conference on Information and Computer Science (NICS)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2017. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, volume 30, pages 406–416. Curran Associates, Inc.

Asad Mahmood, Zubair Shafiq, and Padmini Srinivasan. 2020. A Girl Has A Name: Detecting Authorship Obfuscation. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations (ICLR)*.

Frederick Mosteller and David L. Wallace. 1964. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley series in behavioral science. Addison-Wesley.

Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. 2012. On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*.

X. Pan, M. Zhang, S. Ji, and M. Yang. 2020. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331.

Lisa Pearl and Mark Steyvers. 2012. Detecting authorship deception: A supervised machine learning approach using author writeprints. *Literary and Linguistic Computing*, 27:183–196.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

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

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv 1609.06686*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The Cost of Training NLP Models: A Concise Overview. *arXiv 2004.08900*.

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.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to Fine-Tune BERT for Text Classification? *arXiv 1905.05583*.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2017. Unsupervised cross-domain image generation. In *International Conference on Learning Representations (ICLR)*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural*

Information Processing Systems, volume 32, pages 5753–5763. Curran Associates, Inc.

N. Yu, L. Davis, and M. Fritz. 2019. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7555–7565.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32, pages 9054–9065. Curran Associates, Inc.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *IEEE International Conference on Computer Vision (ICCV)*.

A Appendix

A.1 Analysis of importance given by Decision Tree to Writeprints

Synthetic texts generated by pre-trained LMs are distinguishable to a high degree. This holds even when the adversary decides to use a fine-tuned LM or varies text generation parameters like p-value, k-value, or temperature. Thus, these texts carry exploitable model fingerprints.

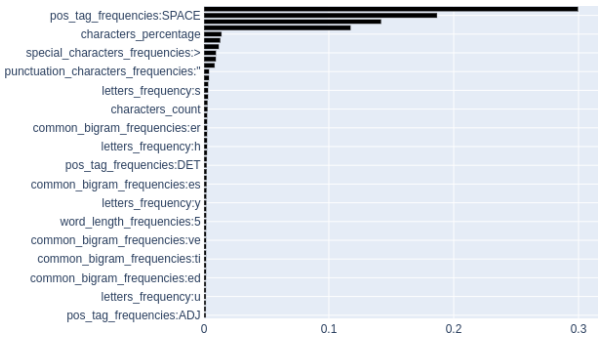
Most challenging is the ability to tell apart texts generated by different fine-tuned versions of the same pre-trained model. XLNet fine-tuned on the training data yields excellent results; except for attributing fine-tuned models accuracies are almost entirely above 90%. Interestingly, decision trees fare quite well offering the advantage of interpretability of decisions. Decision Tree based classifier focusing only on stylistic differences achieves an accuracy of higher than 80% in all three configurations. Investigating the importance given by classifier to different Writeprints show stylistic features like spaces, percentage of characters, and special characters being given the highest importance.

Figure 3 shows a comparison of importance given by a decision tree based attributor to features before and after eliminating whitespace as a feature. Running the experiment again after eliminating the highest rated feature (frequency of white space) results in minimal drop in performance (within a range of 1%- 2%) and shows continued focus on more stylistic language features as key indicators of differences between these texts. This confirms our intuition that different pre-trained versions of language models have different writing styles which are discernible through text classification techniques. Moreover, our experiments showed that fine-tuned LMs retained characteristics from their parent pre-trained LM, allowing an attributor trained entirely on pre-trained text to successfully attribute fine-tuned LM text with above 90% accuracy even in a worst case scenario.

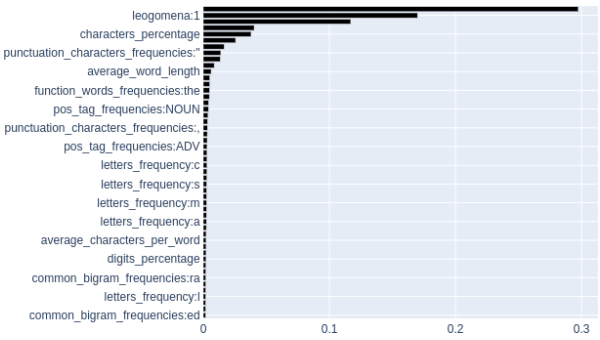
Comparing importance maps from a decision tree attributor trained on Writeprints from pre-trained and fine-tuned GPT2 LMs shows interesting results. From figure 4, it is apparent that top two most important features are common among pre-trained and fine-tuned variants, with a number of other similarities in features given relatively less importance. It shows that there are certain stylistic characteristics that are passed down from a pre-trained LM to its fine-tuned variant.

A.2 Details of pre-trained language models used

In our experiments we have made use of four publicly available pre-trained language models: XL-Net, BART, GPT, GPT2. Details about those are given in Table 8. Comparing sizes, XLNet is trained on largest dataset with over 142 GB of documents. Although no explicit size is mentioned for GPT2, it is said to be trained on 10 times more data than GPT.



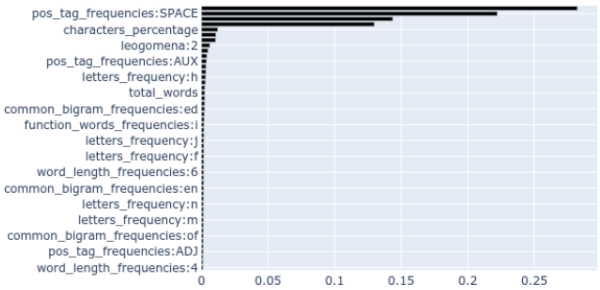
(a) Before removing whitespace



(b) After removing whitespace

Figure 3: Comparison of feature importance with and without whitespace

(a) Importance given by attributor trained on pre-trained text data to different Writeprints features



(b) Importance given by attributor trained on fine-tuned text data to different Writeprints features

Figure 4: Comparison of feature importance for pre-trained and fine-tuned variants of GPT2 LM

Language Model	**Dataset**	**Words**	**Size**	**Number of Documents**
OpenAI GPT	BooksCorpus (Zhu et al., 2015)	∼985M	Not Available	Not Available
OpenAI GPT2	WebText (Radford et al., 2018)	Not Available	40GB	∼8 million documents
BART	WikiText-103 (Merity et al., 2017)	∼103M Words	181MB	28,475 articles
XLNet	BookCorpus + English Wikipedia + CommonCrawl + Giga5 + ClueWeb 2012-B	∼32.8B subword pieces	142GB	Not Available

Table 8: Breakdown of pre-trained LMs used