ECE594 Final Project Report: Automatic Detection of Machine-Generated Text

Abstract

Modern large-scale generative language models (GLMs) are capable of generating multi-paragraph texts that are virtually indistinguishable from human-written text. The task of automatic machine-generated text detection has garnered much research and social interests. However, the current approaches are heavily dependent on the decoding strategy used to generate the text: the classifier that works for one decoding strategy does not work at all for the other. Hence, in this project, we explore to use Fisher Information Matrix to create hidden representations for the textual input and utilize representation learning, one-class techniques, outlier detection algorithms and novelty detection algorithms to achieve a more generalizable detector for machine-generated texts. As a result, our method achieve a gain of 0.18 in AUC for the cross decoding strategy performance while still maintaining a acceptable performance for same decoding strategy performance.

1 Introduction

With the advent of large-scale generative language models (GLMs), such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), the line between human-written and machine-generated text has become more and more blurry. At a surface level, humans can no longer distinguish between human-written and machine-generated text: for longer text, human may still detect machine-generated text through subtleties such as logical fallacies, topic drafting and certain domain-related knowledge, however, for shorter text with less than 32 tokens, humans struggles to even successfully distinguish machine-generated text half of the time (Ippolito et al., 2020). Moreover, harnessing the power of these powerful GLMs no longer requires knowledge in them nor the computing sources, numerous online API provides access to a selection GLMs [1].

The powerful language, coupling with the ease of use, has made GLMs dangerous and detrimental when used under malicious intent. Deceptive text aided by the GLMs can quickly produce and disseminate untruthful and hateful information to cause social harm, set political agenda, and even influences elections. For private companies and individuals, text riddled with fake and harmful information can dismantle the user trust in cyber-spaces, such as online communities and marketplaces, via fake post and fake reviews. Therefore, it is of the interest of both public and private individuals for the machine-generated texts to be detected.

Traditionally, the problem of machine-generated text detection is straightforwardly framed as a binary classification problem: given a piece of text, the task is classify the text into machine-generated class and human-written class. Past work has explored the use of simple statistic method to evaluate generation probability of each token or the probability of the entire sentence according to a given GLM (Gehrmann et al., 2019); based on these probabilities, these methods rank the sentences and perform binary classification using a predefined threshold. These methods are essentially unsupervised since they do not require any training other than the pre-trained GLM itself. Other works have tried to explicitly fine-tune a language model (LM)-based binary classifier, such as BERT (Devlin et al., 2019), on human-written and machine-generated text. These methods result in classifiers that achieve impressive performance when testing on machine-generated texts that are generated using the same generation algorithm (in-domain), yet they fail almost completely when testing on texts generated from another generation algorithm (cross-domain). Specifically, Devlin et al. (2019) found that the performance of these binary classifiers are highly dependent on the *decoding strategy*,

[1] https://transformer.huggingface.co/

the generation algorithm that determines which token is selected the next from the provability distribution over all tokens given the previously decoded token in a sequence; and the classifier trained with texts generated by one decoding strategy will not generalize to texts generated by another decoding strategy.

This project aims address this generalization problem by re-frame the machine-generated text detection task from a binary classification problem to a one-class classification problem. Specifically, instead of creating a model that classifies texts into human-written and machine-generated class, we create a model that classifies texts into human-written and non-human-written class. In order to achieve the machine-generated text detection in this one-class setting, we attempt to improve the past methods in two aspects: (1) *representation* and (2) *detection* method.

To acquire better hidden representation of the textual data, we leverage Fisher Information Matrix (FIM). FIM has been shown to be able to create textual representations of corpora that capture the transferability between natural language understanding (NLU) tasks (Vu et al., 2020); and compare to the last hidden layer output from LMs, FIM computed from a set of selective modules of a LM offers a more multifaceted, comprehensive view of the input textual data. Moreover, we utilize metric learning and representation learning method such as the triplet network (Dong and Shen, 2018) and Deep SVDD (Ruff et al., 2018) in attempt to transform the FIM representations into a space where the human-written texts are congregated together and more segregated from all the other sources of machine-generated texts. In terms of detection method, we explore the uses of *outlier detection* and *novelty detection* methods to classify the transformed FIM representations into human-written and non-human-written classes. Outlier detection, also known as, anomaly detection, aims to capture the "normalcy" represented by the majority of the data and thereby detect the abnormal data patterns that deviate from the normal data (Chandola et al., 2009). Novelty detection aims to discern if test data are from the same distribution as the data available during training and classify test data that are different from training data into a new "novel" class (Pimentel et al., 2014). Outlier detection and novelty detection are two very related domains with only a subtle difference: in outlier detection, the training data is assumed to be contaminated with abnormal data, albeit in small quantity, yet in novelty detection, the training data is assumed to be only from the normal class. Either way these methods are applicable to our problem by considering the human-written texts as the normal class and the non-human-written text, i.e., machine-generated texts from various decoding strategies, as the abnormal or novel class.

Using textual data from human-written texts and machine-generated texts of two decoding strategies, we experiment our FIM-based textual data representation with various methods from outlier detection and novelty detection in both in-domain and cross-domain setting. We show that creating a machine-generated text detector that is generalizable among decoding strategies is a difficult problem: even with more comprehensive representation and detection methods that directly aim to improve the generalizability, the barrier of machine-generated text identifying features between different decoding strategies is greater than previously perceived. Although through our representation and detection method, we managed to improve Area Under the Receiver Operating Characteristic Curve (AUC) by around 18%, the overall AUC is still too low for the method to be reliably used in practice.

2 Related Work

Machine-Generated Text Detection The problem of machine-generated text detection has gained more attention since the advent of powerful pre-trained generative language models. These Transformer-based large-scale models are capable of generating convincing multi-paragraph texts in multiple domains, including creative writing [2] and academic paper writing (Beltagy et al., 2019), and etc. Some work (Gehrmann et al., 2019) is focusing on using the a GLM to evaluate the probability of a given text or the individual tokens under the assumption that if higher the probability more likely the text is generated by the GLM. However, these works are directly dependent on the GLM was used to generated the texts: if the texts were generated by the same GLM used for the evaluation, then the method would have a higher chance of success, otherwise, these methods would not work. Other methods (Devlin et al., 2019) fine-tune another LM to classify the texts into human-written and

[2] https://www.gwern.net/GPT-3

machine-generated class by performing sequence classifications. However, as shown by Devlin et al. (2019), these methods are also dependent on the decoding strategy and GLM that was used to generated the texts. Hence, in this project, we aim to breach this dependency on decoding strategy by utilizing a different textual data representation and various detection methods.

Textual Representation via Fisher Information Matrix Fisher Information Matrix (FIM) is a parameterization-independent metric on statistical models (Osawa et al., 2020). Intuitively, FIM provides a measure of importance, or informative content, of a weight in a statistical model for a given set of input and output. FIM determines which weights are more important by computing the average KL-divergence between the output distribution with the original weights and the output distribution with a set of slightly perturbed weights. Due to its computationally expensive computation, recent works have focused on computing the approximation of FIM. Achille et al. (2019) used FIM to characteristic computer vision tasks to measure the similarities among tasks for meta-learning. Vu et al. (2020) applied FIM to textual data and NLP tasks and successfully use FIMs to create hidden representation to NLP tasks and predict their transferability. Since FIM provides a importance map over model weights given a set of input and output, FIM can then provide a more comprehensive representation of a model's internal response to a given input compared to hidden representation given by the last layers of a model. Hence, in this project, we attempt to use FIM to create textual input data representations.

Outlier Detection and Novelty Detection Outlier detection methods aim to Outlier detection, as known as anomaly detection, aims to recognize the expected behavior from a given dataset and then identify the patterns that are non-conforming to the expected behavior; the data points with these non-conforming patterns are referred to as outliers or anomalies. There are two general types of outliers, namely, *global outliers* and *local outliers*. For global outliers, the are far away from all normal data regions in the space. The local outliers, on the other hand, although far away from some normal regions, are relatively close to at least one normal region. Different unsupervised methods are developed for both detecting global and local outliers (Chandola et al., 2009). Novelty detection is a problem that is related to outlier detection in that it aims to detect previously unobserved patterns in the data, such as data from a new class. The difference between outlier defection and novelty detection is that the for outlier detection it is assumed that the given training data contains anomalies, however, for novelty detection it is assumed the training data should only contain the normal data.

3 Method

In this section, we introduce the methods we used in our exploration for this project.

3.1 FIM Generation

Following the work by Ippolito et al. (2020), we generate FIM using a pre-trained base, uncased BERT model. Since we want the FIM to capture the generic textual information, we select the next token prediction as the task to produce loss and generate gradient. For modules, we select the 12 encoder layer outputs and 12 multi-attention layer outputs to compute the the FIM for each input sentence. To observe if the FIMs can differentiate textual inputs of different sources, i.e., decoding strategies and human-written, we compute and observe the T-SNE 2D visualizations of the average FIMs from each source. As a results, we observe that the FIMs from the 12 multi-attention layers are not as distinctive as the FIMs from the 12 encoder layers. Hence, we decided to use only the encoder layer outputs to compute the FIM for each textual input. Therefore, for each data sentence, we compute a 12-by-768 dimension FIM where 768 is the hidden dimension of the pre-trained BERT.

3.2 Representation Learning

To further transform the representations to make the FIMs from different sources become more distinguishable, we apply metric learning techniques. First, we apply Deep Support Vector Data Description (Deep SVDD) (Ruff et al., 2018), which is a one-class classification technique. Deep SVDD learns a neural network transformation that aims the given hidden representations of data points into a hypersphere, such that the center of the hypersphere is a trainable parameter and the radius of the hypersphere is optimized to be as small as possible. As a one-class technique, it would only take the data from the "normal" class, i.e., the human-written class in attempts to map all the human-

written texts into the hypersphere and leave out the machine-generated texts as anomalies.

Triplet loss is another metric learning technique that is originally designed to learn image similarities such that similar images, i.e., images from the same class, are map closer together yet far from dissimilar images, i.e., images from other classes. It is another way of learning a transformation to group and segregate normal (human-written) and abnormal data (machine-generated) in the embedding space. The one caveat is that Triplet loss requires at least two classes: one positive and one negative. In this case, the learned representation could potentially overfit to the a available negative class similar to the classifiers with trained with direct supervision as discussed in Section 2. In this project, besides using Deep SVDD to compute representations, we also explore using a Triplet loss + Deep SVDD technique, where we jointly optimize the triplet loss and the deep SVDD loss together. The benefit is that through the triplet loss, the human-written texts would locate far away from the machine-generated text of one decoding strategy. Then, through the Deep SVDD loss, we also transform the human-written texts into a tight hypersphere so that they have a better chance of being separated from the machine-generated text from other decoding strategies other the one exposed to the model during training.

3.3 Detection

3.3.1 One-Class Classification

Following the Deep SVDD, one natural detection method perform one-class classification using the learned representation transformation and the center of the hypersphere: For a given FIM to a textual input, we first use the trained deep SVDD to transform into the learned representation; then, we compute the distance between the representation to the center of the hypersphere in the embedding space; finally, this distance is used as the "anomaly" score indicating how likely the given textual input belongs to the human-written class.

3.3.2 Unsupervised Outlier Detection

The one-class classification method discussed above is naturally detecting the global outliers to human-written data in the embedding space. Hence, in this method, we apply unsupervised outlier detection methods that aim to detect local outliers to the representation learned using the deep SVDD + triplet loss method. We also apply one supervised outlier detection method, HBOS, to detect the global outliers from the deep SVDD + triplet loss representation.

3.3.3 Novelty Detection

For the novelty detection, we follow the current state-of-the-art method from computer vision, Skip-GANomaly (Akçay and andToby P. Breckon, 2019), a generative adversarial network (GAN) using U-Net with skip connection as its encoder. Instead of image, in our case, we use the 12-by-768 dimension FIMs as the input to the U-Net. We use a vanilla U-Net as stated without additional modifications.

4 Experiment

4.1 Data

Ippolito et al. (2020) collected human-written texts from web and machine-generated texts produced by GPT-2. Specifically, the human-written texts are taken from popular web pages that are from the same distribution as the training data of GPT-2 data. The machine-generated texts are generated by GPT-2 LARGE model with 774M parameters with two decoding strategies: (1) *random* and (2) *top-k*. For the random decoding, the output token at each step is randomly sample with probability proportional to the predicted distribution over the entire vocabulary. Top-k decoding restricts the predicted distribution at each step to be limited to the top k most likely tokens, where k is a constant and set to be 40 for this dataset. And top-k randomly sample the output token at each step from a distribution over the top k most likely words. Although random and top-k seem similar, given that the predicted distribution over the entire vocabulary is a long-tail distribution, the output texts from these two decoding strategies can be very different since the words from the long-tail forms a substantial amount of probability mass.

For training data, there are 250K sentences from the human-written text gathered from web (WebText) and 250K sentences each for random decoding strategy (Random) and top-40 decoding strategy (Top-40). For testing data, there are 50K sentences for WebText, Random, and Top-40 each.

In our experiment for machine-generated text detection, there are two settings: (1) In-Domain and (2) Out-domain. For in-domain, we train on WebText and Top-40 (if the method requires instances from the negative class) and test on WebText and

Top-40. For out-domain, we train on WebText and Top-40 and test WebText and Random.

4.2 Baselines

For our experiments, we consider the following baseline models.

Fine-tuned BERT (FT-BERT). Following the work by Ippolito et al. (2020), we fine-tuned base, uncased pretrained BERT models to two classifiers; one for WebText and Random and another for WebText and Top-40. The BERT model is fine-tuned for sequence classification such that a multi-layer perceptron (MLP) is attached to the last hidden layer of the <CLS> token. FT-BERT represents the current state-of-the-art in machine-generated text detection.

FIM + SVDD + One-Class Classifier (SVDD-OC). One-class classifier is a natural extension to SVDD. Given the FIM of a textual input, use the trained SVDD network to compute the distance between its embedding and the centroid. The the distance can be used as the "anomaly score" to indicate how much this given instance belongs to the "normal", i.e., human-written, class. SVDD-OC is explored to examine if the representation learning on FIMs is helpful for machine-generated text detection.

FIM + Triplet-SVDD + One-Class Classifier (TS-OC). This model is similar to SVDD-OC only now the representation learning is done using a jointly trained triplet loss and SVDD loss. It uses the same way of computing the anomaly scores as SVDD-OC.

FIM + Skip-GANomaly (Skip-GAN). As discussed in 2, Skip-GAN is the state-of-the-art model for novelty detection for computer vision applications. Here we consider to use FIMs to represent our textual input and then use ResNet from the Skip-GAN to directly extract features from the FIMs. The rest follows the original Skip-GAN paper for its novelty detection setup. Skip-GAN explores if the use of novelty detection method can help the machine-generated text detection.

FIM + LOF (FIM-LOF). Here we apply an unsupervised outlier detection method, namely Local Outlier Factor (LOF) (Breunig et al., 2000), to FIM representation of the textual input. With these models, we explore if the use of outlier detection methods can help to improve machine-generated text detection. During the project, we explored with as many as 9 outlier detection methods, but we only present LOF here since it has the best performance.

4.3 Implementation Details

The algorithm used to compute FIM is adapted from the official implementation released by Vu et al. (2020). The pre-trained, base uncased BERT[3] model is implemented and maintained by the Huggingface's library, Transformers (Wolf et al., 2020). The code on the Skip-GANomaly's main architecture and loss computation is adapted from its official release[4] by Akçay and andToby P. Breckon (2019). The LOF and other outlier detection methods used in our experiments are implemented and maintained by the Python library PyOD[5] (Zhao et al., 2019). All the model and training parameters are set to their default values.

5 Results and Discussion

Detection Performance. We present the model performances on machine-generated text detection task in terms of AUC. As shown in Table 1, the FT-BERT model, as expected based on the results from Ippolito et al. (2020), achieves a near-perfect in-domain performance, with an AUC larger than 0.99. However, FT-BERT's out-domain AUC is a meager 0.45, which is worse than change and the lowest among all the models. This indicates that FT-BERT severely overfit to the training strategy provided during training.

For SVDD-OC, TS-OC, Skip-GAN and FIM-LOF, although the out-domain performances have seen some gain, ranging from 0.02 to 0.11, their in-domain performances have also suffered a great deal to the point that the in-domain performances is also around chance level. This indicates these methods have failed to learn any features that is useful in separating human-written and machine-generated texts. The commonality among these methods is that they all only do not receive any supervision that differentiate the human-written and machine-generated text during training. Some methods, i.e., Skip-GAN and SVDD-OC, are not exposed to the machine-generated texts during training at all. These results show that the learning algorithms needs some level of supervision to be able to extract useful features to distinguish human-written and machine-generated texts.

[3]https://huggingface.co/bert-base-uncased
[4]https://github.com/samet-akcay/skip-ganomaly
[5]https://pyod.readthedocs.io/en/latest/

Model	Top-40 (In-Domain)	Random (Out-Domain)
FT-BERT	**0.9972**	0.4504
SVDD-OC	0.5109	0.5677
TS-OC	0.8727	**0.6359**
Skip-GAN	0.4762	0.4764
FIM-LOF	0.4581	0.5411

Table 1: The in-domain and cross-domain performance for models in terms of AUC. For the in-domain experiment, the model is trained and evaluated on the top-40 decoding strategy. For out-domain, the model is trained on top-40 and evaluated on random decoding strategy. The best performances are in bold.

Lastly, for TS-OC, it achieves the highest out-domain AUC of 0.64 and a moderate in-domain AUC of 0.87. Although TS-OC's out-domain is still less than generally acceptable, it is already higher than the FT-BERT by 0.18. Moreover, comparing to FT-BERT, TS-OC's loss in out-domain gain of 0.18 outweighs its in-domain AUC loss of 0.13. Moreover, an in-domain AUC of 0.87 is still a generally acceptable level of performance. However, it does seem as though the gain of the out-domain AUC is obtained at the expenses of the loss of the in-domain AUC.

6 Conclusion and Future Work

In this project, we attempt to increase the generalizability of the existing machine-generated text detection method. Specifically, we aim to create a detection algorithm that is less dependent on the decoding strategy of the machine-generated text. In creating the algorithm, we explore the possibility of using FIM as representation for textual data instead of last hidden layer output from pre-trained LMs, and using various one-class classification and outlier detection algorithms instead of a binary classifier trained with direct supervision. As a result, we use a metric learning algorithm with one-class classification that is jointly optimized with triplet loss and SVDD loss to achieved a 0.18 AUC gain in out-domain scenario.

In terms of future work, one immediate step is to evaluate our method under more cross decoding strategy and cross GLM scenarios. A second direction would be to perform a careful study on the features that are helpful in performing the machine-generated text detection and analyze if the features needed for different decoding strategies are in conflict with each other in the sense that the features that are useful for detecting one decoding strategy's text is detrimental to detecting the other.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2Vec: Task Embedding for Meta-Learning. *arXiv e-prints*, page arXiv:1902.03545.

Samet Akçay and Amir Atapour-Abarghouei andToby P. Breckon. 2019. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 93–104, New York, NY, USA. Association for Computing Machinery.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3).

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.

Xingping Dong and Jianbing Shen. 2018. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

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. 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.

Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Akira Naruse, Chuan-Sheng Foo, and Rio Yokota. 2020. Scalable and practical natural gradient for large-scale deep learning. *CoRR*, abs/2002.06015.

Marco A. F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. Review: A review of novelty detection. *Signal Process.*, 99:215–249.

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

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Lucas Deecke, Shoaib A. Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.