DETECTION OF MACHINE-GENERATED FRAGMENTS IN TEXT

DRAFT

Anastasiya Voznyuk
Department of Applied Informatics and Mathematics
Moscow Institute of Physics and Technology
vozniuk.ae@phystech.edu

Andrey Grabovoy
Moscow Institute of Physics and Technology
Antiplagiat Company
grabovoy@ap-team.ru

ABSTRACT

This paper considers the problem of detecting machine-generated parts of text in the document. We introduce a model, that can detect such text fragments and distinguish their origin among several language models. To solve this problem we combine two models. The objective of the first one is to divide document into fragments of human-written text and machine-generated text. The second model aims to classify the generated fragments according to the language model from which they were obtained. In the computational experiment, we analyse the quality of such approach on dataset of documents with fragments generated by GPT-2 and GPT-3.

Keywords Text Generation, Detection of Machine-generated text, Neural Authorship Attribution, Machine Learning, Natural Language Processing

1 Introduction

In recent years there has been a rapid development of language models for text generation, transformers in particular, for example GPT[14], GPT-2[15], GPT-3[5], CTRL[9] and BERT[6]. Problem of detecting machine-generated text has become more relevant recently due to the released ChatGPT model by OpenAI. This model was learnt on massive amounts of data and now is able to provide texts that are hardly distinguishable from human texts. The opposite task of detecting machine-generated text becomes more important due to many possible malicious usages of this technology. One can analyse the whole text or fragments of text. We will focus on methods of subtracting the text-generated fragments in the set of documents.

With increased computational resources past statistical methods were applied to the novel detection problem. One of them was prediction entropy as an indicator of fake text, as described in [12]. Also perplexity[4] of the text or frequency of rare bigrams [8] or value of tf-idf [16] can be taken into account. Another approach is to use classifiers that try to label given texts [11].

For a long period recursive neural networks like LSTM were showing best results at solving the problem of detection. At 2018 the mechanism of self-attention[17] was introduced and transformers have become new state-of-the-art approach. Every new model has its own features and is usually learnt on larger amounts of data, but attention mechanism always stays. These model can be used in detection of machine-generated text in two ways[16]. The first method is using a language model that searches for artefacts from methods, which most models are using for generating concise texts [7]. Additional training on new data is not required. The second method is fine-tuning based detection: one fine-tunes a language model to "detect itself" with using some stochastic methods, for example top-k, top-p sampling etc. The first step is to divide each document into non-overlapping blocks of sentences. We take the baseline from work[11] on plagiarism detection. Also metrics and construction of loss function from the works about text segmentation problem can be borrowed. We suppose, that within a fragment with artificial text there is a finished idea and similar semantic structure. For example, loss function is described at [10], and metrics are described at [3]. The next step is to detect the origin of machine-generated text and this can be done with usage of transformer-based models. This papers presents computational experiments with these models to combine them in one model and analyze the parsmets on which the best performance will be obrained. Dataset will be constructed from existing datasets, such as GPT-2 output dataset[1] and RUaTD competition dataset[2].

2 Problem statement

Let $\mathbf{W}$ be our alphabet, tokens consists from characters from that alphabet.

Let

$$\mathbb{D} = \{[t_j] : t_j \in \mathbf{W}\}$$

be the space of our documents.

We have set of n documents

$$\mathbf{D} = \bigcup_{i=1}^{n} D^i, D^i \in \mathbb{D}$$

We have K classes of our text-generative models, that generated fragments of text in $\mathbf{D}$.

We have $K + 1$ labels, that we will use for multiclass classification, where 0 - label of human-written fragment, $1...K$ - labels that represent corresponding language model, let $\mathbf{C}$ be our set of labels.

Our model ϕ is a superposition of two mappings, $\mathbf{f}$ and $\mathbf{g}$. Mapping $\mathbf{f}$ is responsible for text segmentation. Mapping $\mathbf{g}$ is responsible for classifying obtained fragments.

$$\phi : \mathbf{g} \circ \mathbf{f}$$

$$\phi : \mathbb{D} \to \mathbb{T},$$

where

$$\mathbb{T} = \{[t_{s_j}, t_{f_j}, C_j]_{j=1}^{J} : t_{s_j} = t_{f_{j-1}}, s_j \in \mathbb{N}_0, f_j \in \mathbb{N}, C_j \in \mathbf{C}\}$$

and where J - number of fragments in segmentation of the document, t_{s_j} - starting index of the j-th fragment, t_{f_j} - finishing index of the j-th fragment, C_j - class of the j-th fragment.

2.1 Text fragmentation

The first step is to divide our text into fragments of different origin.

$$\mathbf{f} : \mathbb{D} \to \mathbf{T}^*$$

where

$$\mathbf{T}^* = \{[t_{s_j}, t_{f_j}]_{j=1}^{J} : t_{s_j} = t_{f_{j-1}}, s_j \in \mathbb{N}_0, f_j \in \mathbb{N}\}$$

Thus, $\mathbf{T}^*$ is a set of all possible sequences of non-overlapping blocks of texts, that covers complete document.

We assume that in each document blocks of text with the same origin is big enough. Thus, we search for style changes on sentence levels. Style change indicates that several next sentences staring from the beginning of the style change have different origin. Each sentence receive a label 0 or 1. A fragment is a sequence of neighbouring sentences of maximum possible length, that have the same label. We repeat this process for every document in given set of document. This can be viewed as a binary classification task. Thus, loss function

$$\mathbf{Loss_{frmg}}(x) = \sum_{i=0}^{J}[-y_i \log p_i - (1 - y_i)\log(1 - p_i)]$$

where $x \in \mathbb{D}$, y_i - predicted label of x, p_i - probabilty of that label.

2.2 Fragment classification

The second step is to classify each fragment with label from $\mathbf{C}$.

$$\mathbf{g} : \mathbf{T}^* \to \mathbf{C}$$

We apply that function to every text fragment, obtained from the previous step. This is a classical multi-classfication task. We will use cross-entropy as a loss function:

$$\mathbf{Loss_{cls}}(x) = \sum_{i=0}^{K} -p(x)\log q(x)$$

where $p(x)$ - probability of object x to be actually labelled with i, $q(x)$ - probability of object x to be labelled with i in prediction.

2.3 Metrics

Total loss of the model is calculated as sum of losses on both stages. This is an inner criterion of quality of our model.

$$\mathbf{Loss} = \mathbf{Loss_{cls}} + \mathbf{Loss_{frgm}}$$

Outer criterion of our model can be set as a metric, that compares the boundaries of some fragments with the reference. For If model doesn't detect a particular fragment at all, we set this metric to 0. Otherwise, we will calculate a distance between two vectors with indexes of beginning of the fragment and its endings. The more the distance, the smallest values we will assign to that pair.

3 Computational Experiment

The goal of computational experiment is to analyse how good baseline solution in classifying fragments of text. RUATD competition, which provided the dataset, also provided accuracy score for their baseline solutions. One soltuion uses tf-idf + logistic regression approach, the other solution uses fine-tuned BERT. We will use the second approach, but our own solution and compare our results with the ones, suggested by the authours of competition.

3.1 Dataset

For our experiment we took the data, that was collected for RUATD 2022 competition[2] for multitask classification. The test dataset contains 215105 short fragments, that were generated by different 13 models. Authors claim that various language models fine-tuned on different tasks: machine translation, paraphrasing, summarization, simplification and unconditional text generation - were used to generate texts. Moreover, the part of the set was annotated automatically by different generative models. Among models there are several versions of ruGPT, mT5 and ruT5 and M-BART.

3.2 Configuration of algorithm run

We used only a part of dataset for the sake of decreasing the time of training. For tokenization of sentences pretrained embedding from `bert-base-multilingual-cased`[1] were used. The same model was for fine-tuning for our sequence classification task. We trained our model for 3 epochs. For optimization we used AdamW optimization algorithm[13].

3.3 List of expected figures and tables

1. Table of train accuracy and validation accuracy on each epoch

2. Graph of validation / train loss versus epoch - Figure 1

3. Graph of Precision-Recall Curve for every class - Figure 2

4 Preliminary report

We received 56% of accuracy with 3 epoch of training only using the part of the dataset. It is les than the score, provided by the authors of competition, which is 59%, but this difference is explained by smaller size of the dataset. Also we may suggest that due to increasing loss on validation set, our model started to overfit.

Table 1: Table of train accuracy and validation accuracy on each epoch

Number of epoch	Train. loss	Valid.loss	Valid. accuracy
1	1.30	1.26	0.54
2	1.07	1.27	0.56
3	0.86	1.35	0.56

[1] `https://github.com/google-research/bert/blob/master/multilingual.md`

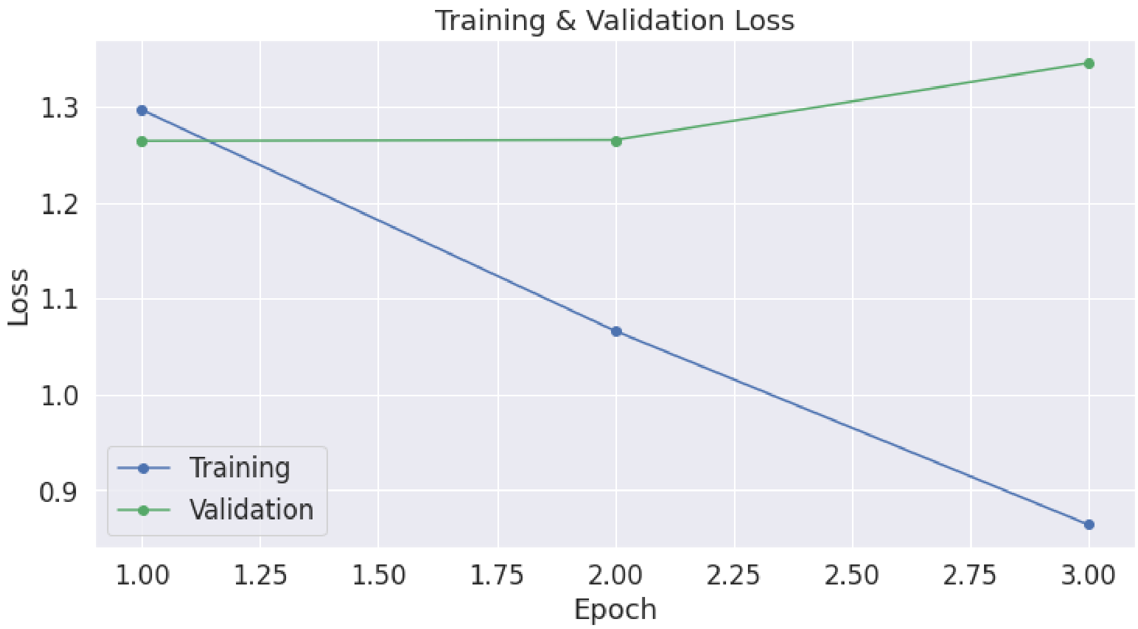

Figure 1: Graph of validation / train loss versus epoch

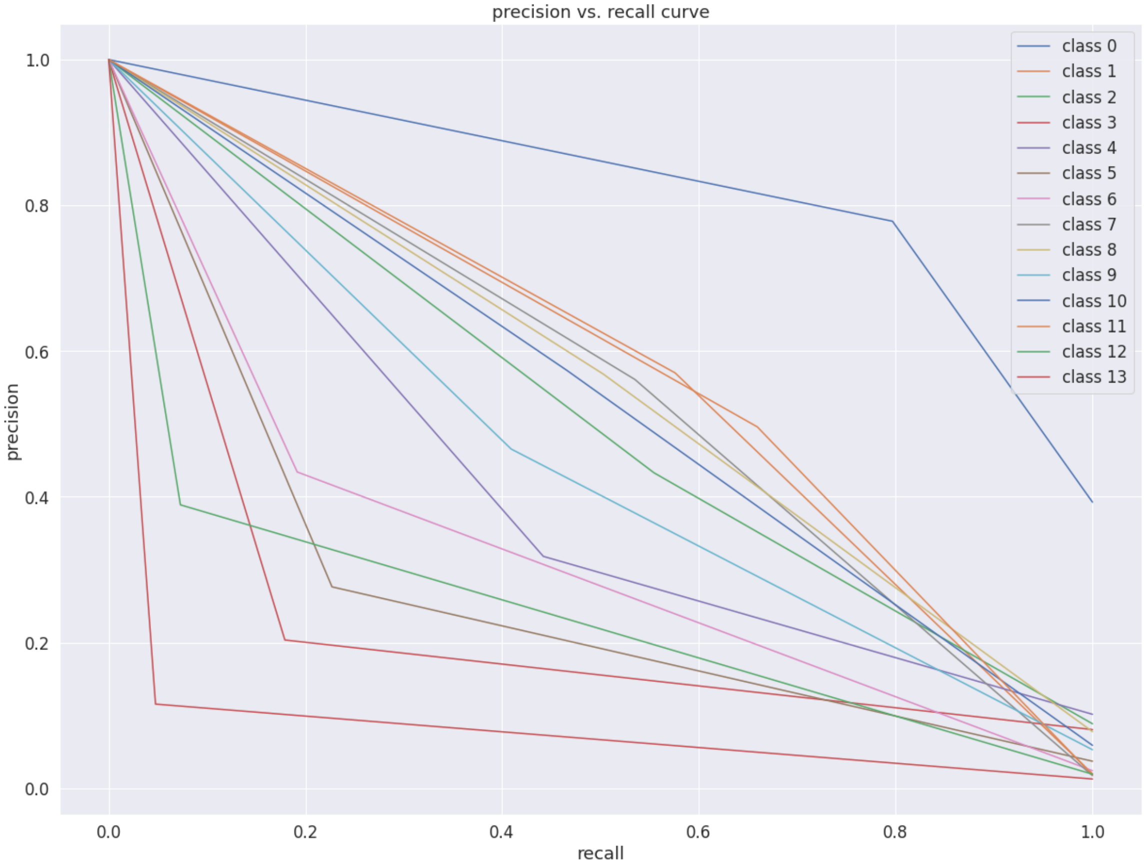

Figure 2: Graph of Precision-Recall Curve for every class

References

[1] Gpt-2 ouput dataset.

[2] Ruatd multitask dataset.

[3] Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. Attention-based neural text segmentation. In *European Conference on Information Retrieval*, 2018.

[4] Daria Beresneva. Computer-generated text detection using machine learning: A systematic review. volume 9612, pages 421–426, 2016.

[5] 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. Language models are few-shot learners. 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[7] Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 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, 2019.

[8] Kustarev A.A. Raigorodsky A.M Grechnikov E.A., Gusev G.G. Detection of artificial texts. 20089.

[9] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. 2019.

[10] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In *North American Chapter of the Association for Computational Linguistics*, 2018.

[11] Mikhail P. Kuznetsov, Anastasia Motrenko, Rita Kuznetsova, and Vadim V. Strijov. Methods for intrinsic plagiarism detection and author diarization. In *Conference and Labs of the Evaluation Forum*, 2016.

[12] Thomas Lavergne, Tanguy Urvoy, and François Yvon. Detecting fake content with relative entropy scoring. In *Pan*, 2008.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017.

[14] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

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

[16] 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. Release strategies and the social impacts of language models. 2019.

[17] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.