Ministry of Science and Higher Education
of the Russian Federation
ITMO University

Faculty of Digital Transformations
Subject «Advanced Natural Language Processing»

REPORT
on course-project «Neural detection of AI-generated text»

Students: Bogdan Chertkov, j4232c; Alexander Petrov, J4234c; Kirill Mukhin, J4234c; Igor Vernyy, J4234c

Head of Project: Maria Khodorchenko, PhD, Senior researcher; Timur Sohin, Assistant researcher

Project completed with grade _____
Commission member signatures:

_____
(signature)

_____
(signature)

Date _____

St. Petersburg
2023

LIST OF EXECUTORS

Executor, Master student of Big Data and Machine Learning	_____ (signature, date)	Bogdan Chertkov
Executor, Master student of Big Data and Machine Learning	_____ (signature, date)	Alexander Petrov
Executor, Master student of Big Data and Machine Learning	_____ (signature, date)	Kirill Mukhin
Executor, Master student of Big Data and Machine Learning	_____ (signature, date)	Igor Vernyy

TABLE OF CONTENTS

INTRODUCTION

In the ever-evolving landscape of artificial intelligence, the proliferation of advanced language models has ushered in a new era where machines can generate text that rivals human-written content. Among these cutting-edge models, the advent of OpenAI's GPT-3.5 has significantly augmented the capabilities of natural language processing, raising the bar for what can be achieved in the realm of automated content creation. As the power and sophistication of AI-generated text continue to advance, the need for robust tools to discern between human and machine-generated content becomes increasingly imperative.

This article delves into the intriguing realm of neural detection mechanisms specifically tailored for identifying AI-generated text, with a focus on content generated by Llama and its contemporaries. As these language models become more prevalent in various applications, ranging from chatbots and content creation tools to news generation and creative writing, the potential for misinformation, bias, and manipulation also grows. Understanding and developing effective strategies to detect the origin of the text, whether human or machine, is crucial for maintaining the integrity of information flow in our digital society.

Throughout this exploration, we will scrutinize the underlying principles of neural detection, the challenges faced in distinguishing between AI-generated and human-generated text, and the innovative approaches researchers and developers are employing to tackle this burgeoning issue. From the intricacies of natural language understanding to the nuances of contextual interpretation, we will navigate the complexities of differentiating between the artistry of human expression and the precision of machine-generated content.

According to Yandex search data [1], interest in neural networks has grown more than 15 times since the beginning of 2022. However, it is hard to find if the text was written by humans or just generated by a neural network due to significantly increased quality of generative

networks. This could lead to problems related to authenticity and regulations as well as other unethical usages of such network including plagiarism, generating fake news, spamming, and so on. Thus, the generated texts identification comes to be a relevant problem nowadays, so the goal of this work is to train a model that could successfully determine if the text was generated by a model or written by a human.

1 Related work

Early approaches in the field of text detection often focused on rule-based systems and statistical methods to identify patterns indicative of automated content. These foundational studies laid the groundwork for subsequent, more sophisticated approaches.

A significant body of work has emerged in the realm of natural language processing, leveraging techniques such as sentiment analysis, syntactic analysis, and semantic understanding to discern subtle nuances in language that may betray the origin of the text.

As the capabilities of AI-generated text advanced, researchers explored adversarial training methods to enhance the robustness of detection models. Studies have delved into the creation of countermeasures against adversarial attacks, acknowledging the evolving nature of the AI landscape.

Numerous studies have employed machine learning models, including traditional classifiers and more recent deep learning architectures, for text classification tasks. These models leverage features extracted from the text to distinguish between human and AI-generated content.

With the advent of transformer models, such as GPT-3 and BERT, the landscape of text generation and detection has witnessed a paradigm shift. Studies investigating the capabilities and limitations of these models contribute significantly to understanding their role in neural detection.

Currently, different solutions to the problem exist that differ in their accuracy [2] These other approaches for AI-generated text detection exist including the ones listed below:

— OpenAI's Classifier [3] is fine-tuned through supervised learning to categorize information into binary groups;

— DetectGPT [4], a text analyzer that works by crunching numbers, generating logarithmic probabilities to understand the content;

— GPTZero [5] calculates confusion values for text. Lower perplexity means the text is less random and more clarity;

— Watermarking [6] includes an “exclusion list” that avoids certain words, which makes the language model more accurate.

2 Methodology

2.1 Proposed pipeline description

The text recognition pipeline we developed is anchored in the utilization of an artificial model to discern and generate responses to a dataset sourced from the Yahoo Answers QA repository [7]. This dataset encompasses a rich collection of 90,000 questions posed and answered by real individuals on the Internet. The primary objective of our pipeline is to employ cutting-edge models, specifically Llama [8] from Meta and Phi [9] from Microsoft, to provide answers to these questions, thereby creating a comprehensive training dataset for subsequent classification models.

The pipeline begins with the ingestion of questions from the Yahoo Answers QA dataset, serving as the input for both the Llama and Phi models. These advanced neural networks generate responses to the input questions. The resultant dataset is a compilation of answers to questions authored by real individuals, responses generated by the llama model, and responses generated by the Phi model.

To research the robustness and accuracy of the classification process of different models, we employ three distinct classification models: DeBERTa [10], Recurrent retention, and T5. These models are trained on the aggregated dataset, which encompasses a diverse array of responses—human-generated, llama-generated, and Phi-generated. The training process involves exposing the classifiers to this amalgamated dataset, allowing them to discern patterns and nuances that distinguish responses originating from different sources. You can see schema of the pipeline at figure 2.1.

In essence, our text recognition sequence encapsulates a multi-step process:

1) Dataset Selection: Yahoo Answers QA dataset with 90000 questions and answers from real individuals on the Internet.

2) Model Implementation: Integration of state-of-the-art models, namely llama from Meta and Phi from Microsoft, to generate responses to the selected dataset.

3) Data Aggregation: Compilation of responses from real individuals, llama, and Phi into a cohesive dataset, serving as the foundation for subsequent classifier training.

4) Classifier Training: Implementation of three classification models—DeBERTa, Recurrent retention, and T5—trained on the aggregated dataset to discern and categorize responses based on their origin.

This comprehensive pipeline bridges the realms of human-generated and AI-generated content, harnessing the collective intelligence of real individuals and cutting-edge neural networks to create a nuanced dataset for training advanced classifiers. The resultant models, enriched by exposure to diverse responses, hold the potential to significantly advance the field of text recognition by accurately distinguishing between human and AI-generated content.

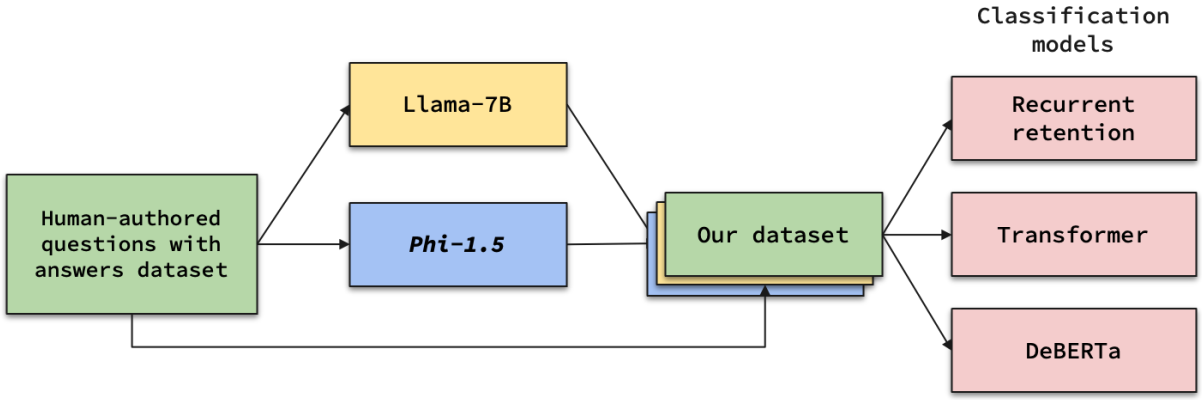

Figure 2.1 — Graphical visualization of the pipeline stages

2.2 Dataset analysis

As the initial dataset, Yahoo Answers QA dataset from Hugging Face portal was used. It contains roughly 90000 question-answer pairs written by humans. The data included is a question itself, best answer

to it and a branch of all other answers given to the question. Dataset file consists of the following columns:

— id (integer): unique identifier of the row;

— question (string): question that is expected to be answered;

— answer (string): most relevant answer to the question;

— nbestanswers (array): list of all answers to a question in descending order of relevance;

— main_category (string): category to which the question belongs.

We are most interested in the first three columns here. The rest make sense to remove.

During the EDA, the distribution of words and characters count in both questions and answers was calculated to determine the preferred length of sequence for LLMs. Figure 2.2 illustrates that most questions appeared to have slightly less than 10 words or 40 characters in it while answers have right skewed distributions.

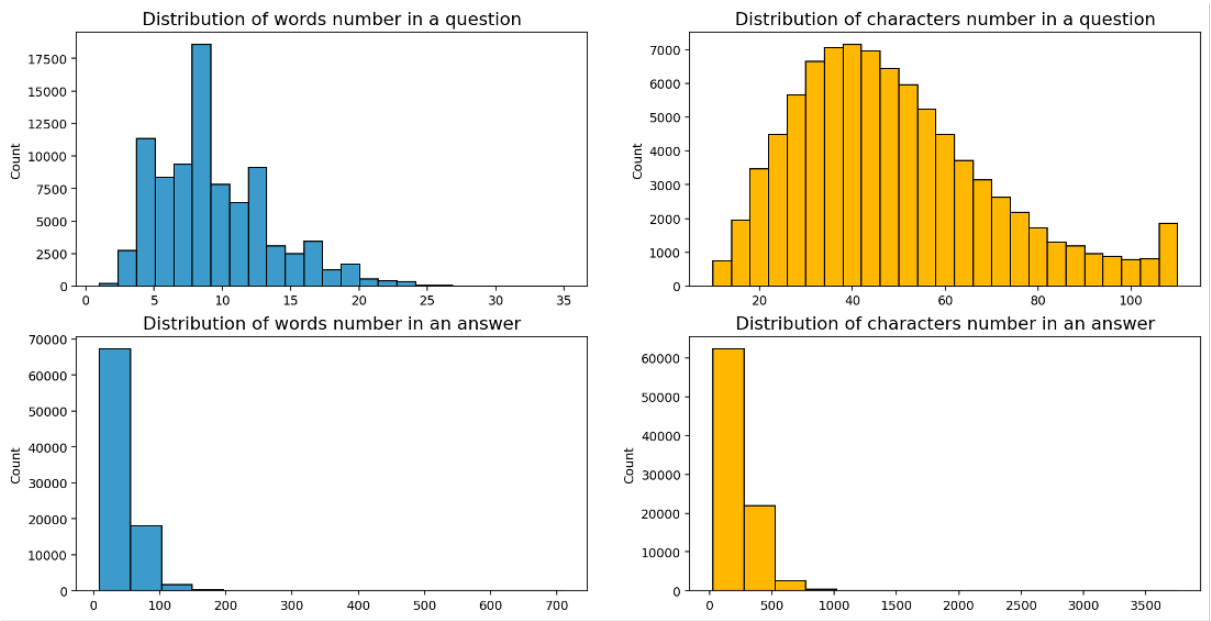

Figure 2.2 — Some plots of exploratory data analysis

A word cloud provides a visual representation of the most frequently occurring words in a given text or dataset. While it's not a detailed analysis tool, it offers valuable insights into the prominent

themes, topics, or keywords within the provided information. The size of each word in the word cloud corresponds to its frequency in the text. Larger words are used more frequently, indicating their significance in the context of the provided data. Figure 2.3 contains word clouds for two dataset columns: question and answer.

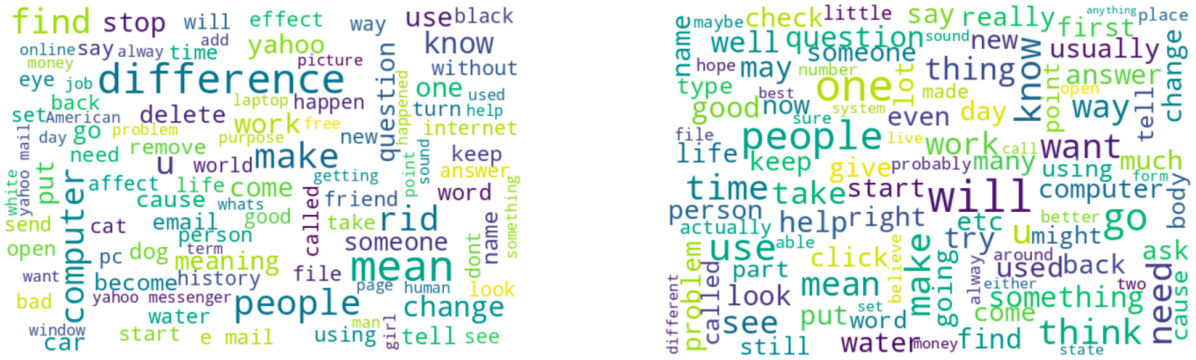

Figure 2.3 — Left plot - word cloud for questions, right - word cloud for answers

2.3 Text generation models

2.3.1 Llama

Llama is large language model developed by Meta and released in the beginning of 2023. Llama 2 is an auto-regressive language model that uses an optimized transformer architecture. Number of parameter of this network is ranging in scale from 7 billion to 70 billion parameters. To reduce the number of required computing resources, we will use the lowest version.

We are utilizing the Llama model to generate synthetic data, employing the C++ implementation known as llama.cpp. This implementation is compatible with MacOS. The large llama 7B network, comprising 7 billion parameters, typically demands around 28 gigabytes of memory in full precision mode. To optimize memory usage, we have quantized the network to 4 bits, resulting in an 8-fold reduction in memory consumption to 3.5 gigabytes.

Quantization of a neural network is the process of reducing the precision of its weights and activations. In the context of deep learning, neural networks typically use high-precision floating-point numbers, such as 32-bit or 64-bit, to represent parameters and intermediate values during training and inference. Quantization involves converting these high-precision numbers into a lower bit precision format, such as 8-bit or 4-bit integers.

2.3.2 Phi

Microsoft introduced Phi-1.5, a 1.3 billion parameter LLM, predominantly trained on 30 billion tokens of synthetic ”textbook-like”data. The model's performance on common sense reasoning, and question answering benchmarks is impressive, rivaling models with 5-10 times its size. Phi-1.5 demonstrated competencies in multi-step reasoning, elementary coding, and even displayed rudimentary in-context learning abilities.

Since the size of this network is several times smaller than Llama, requiring less than 3 gigabytes to run, quantization is not required in this case.

2.4 Classification models

2.4.1 Justification for the choice of models

We decided to use DeBERTa, Recurrent Retention, and T5 for text source classification because they represent different approaches to language representation. DeBERTa focuses on contextual embeddings, Recurrent Retention involves recurrent neural networks for sequential understanding, and T5 is a transformer-based model designed for text generation and understanding. Combining them could provide a diverse set of representations, capturing various aspects of language.

Each model may have strengths and weaknesses in recognizing patterns in the data. Our experiments will help to compare these models.

2.4.2 DeBERTa

DeBERTa [10] model takes BERT as a basis.

BERT is the first bi-directional (or non-directional) pretrained language model [11]. It uses self-supervised learning to learn the deep meaning of words and contexts. After pretraining, the model can be adapted to different tasks as well as different datasets with minimal adjustments.

While BERT is definitely a masterpiece, it also poses a number of limitations. Here, we briefly mention some of the most important ones. All of these will then be addressed in more recent work (that we will discuss in the next sections).

— During pretraining, BERT predicts each masked token independently of the others, which is an oversimplification, leading to suboptimal performance;

— BERT learns from only 15% of the input tokens;

— The Next Sentence Prediction task is too weak compared to Masked Language Modeling;

— The context length of BERT, which is fixed as 512, is small for some tasks;

— In the Transformer architecture, the Embedding dimension size must be set equal to the Hidden layer dimension size, which is a design constraint that does not have any semantic reason;

— The masking of training data is done once and reused for all epochs. This can be improved by dynamic masking at any epoch.

DeBERTa focuses intensively on positional encoding. Specifically, it makes improvements by introducing 2 novel techniques:

A disentangled attention mechanism, in which word content and position are separated (in contrast to be summed up as in BERT). An enhanced mask decoder, in which both relative and absolute position of words are taken into account (notice that in previous work, either absolute position, e.g. BERT, or relative position, e.g. XLNet, is used). Given the similar model size, DeBERTa outperforms former models (e.g. RoBERTa, XLNet, ALBERT, ELECTRA) in many tasks with only half of the training data, achieving SOTA performance.

2.4.3 Recurrent network

Recurrent Neural Networks (RNNs) are a class of neural network architectures designed to handle sequential data by capturing dependencies and patterns over time.

RNN networks for text processing tasks they have limitations such as difficulty in capturing long-range dependencies and issues with vanishing or exploding gradients. But we decided to choose this network because, despite its simplicity, it can show quite effective results.

RNNs process sequential data one element at a time while maintaining an internal state. In the context of text classification, each word or token in the input sequence is processed one at a time, considering the context of the previous words. At each time step, the RNN maintains a hidden state that captures information about the input sequence seen so far. This hidden state is updated based on the current input and the previous hidden state.

For text classification, the final hidden state of the RNN after processing the entire input sequence is used as the representation of the input text. This representation is then fed into a fully connected layer with a softmax activation function to produce the final classification probabilities.

2.4.4 T5

The T5 (Text-To-Text Transfer Transformer) model is a versatile transformer-based architecture developed by Google Research [12]. Although T5 is not specifically designed for text classification tasks, its unique text-to-text framework allows it to be applied to a wide range of natural language processing (NLP) tasks, including text classification.

T5 network architecture follows the transformer architecture, which was introduced in the paper "Attention is All You Need"[13]. T5 consists of multiple layers of transformer blocks. Each transformer block includes self-attention mechanisms and feedforward neural networks. The self-attention mechanism allows the model to weigh different parts of the input sequence differently, capturing contextual relationships effectively.

T5 can be fine-tuned on task-specific data, allowing it to adapt to the characteristics of the text classification problem at hand. The fine-tuning process refines the model's parameters to make it more effective for the target task.

This network has demonstrated state-of-the-art performance on various benchmark datasets and tasks. If it has proven effective in similar classification tasks, it might be a natural choice for the specific task of distinguishing between human and machine-generated text.

3 Experiments

3.1 Dataset generation using LLMs

We ran our dataset through two models Llama and Phi. The length of the input and output sequences were chosen based on the results of exploratory data analysis. The generated responses were then post-processed to remove various junk.

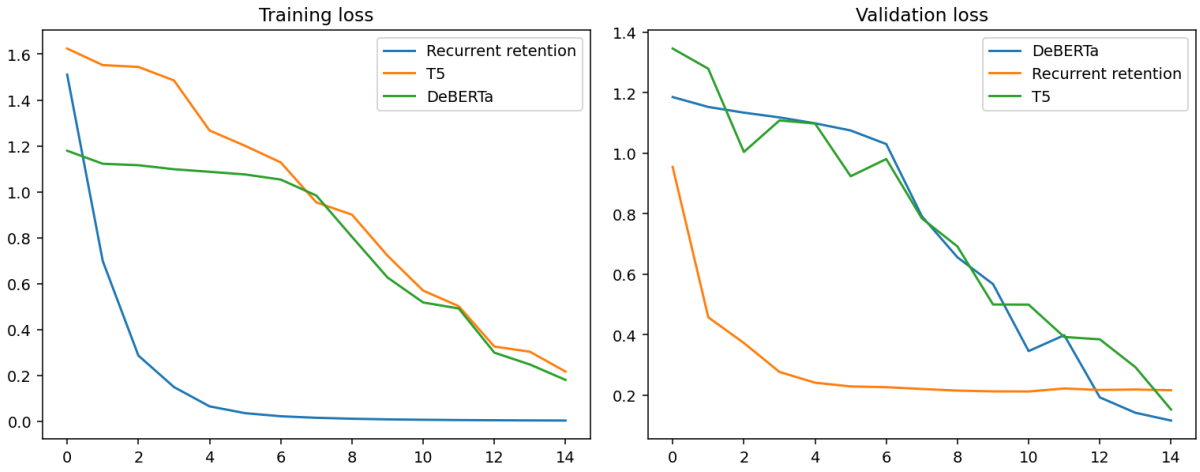

Figure 3.1 — Plots for losses during training and validation process

Below you can see examples of generated texts along with original question and answers written by real human.

— Question: «How to get rid of a beehive?»;

— Human-written answer: «Call an area apiarist. They should be able to help you and would most likely remove them at no charge in exchange for the hive. The bees have value and they now belong to you.»;

— Llama: «1. Hire a pro. 2. Don't do it yourself.»;

— Phi: «You can call a professional beekeeper or use a bee trap to relocate the hive.».

It is easy to see that networks often exhibit different patterns of text generation. For example, they «like» to generate an answer in the form of a series of numbered points.

To train classification models, the resulting dataset was divided into training, validation and test samples. The last two account for a share of 12.5 percent each.

3.2 Classification

Training process shows (the left side of figure 3.1) that Recurrent Retention has the least training loss after 15 iterations but underperforms on validation loss. DeBERTa in turn showed the best performance for the validation loss (the right side of 3.1).

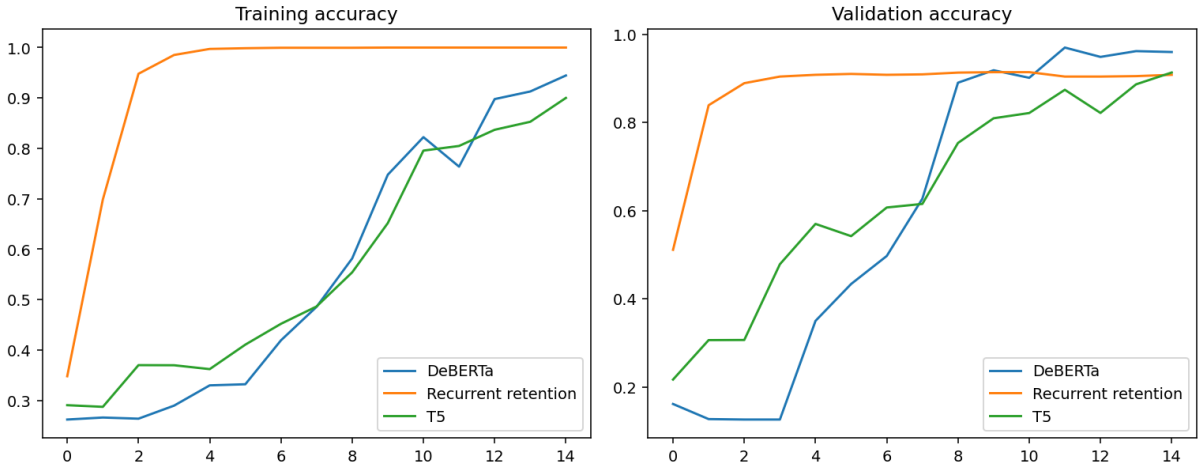

Figure 3.2 — Plots for accuracies during training and validation process

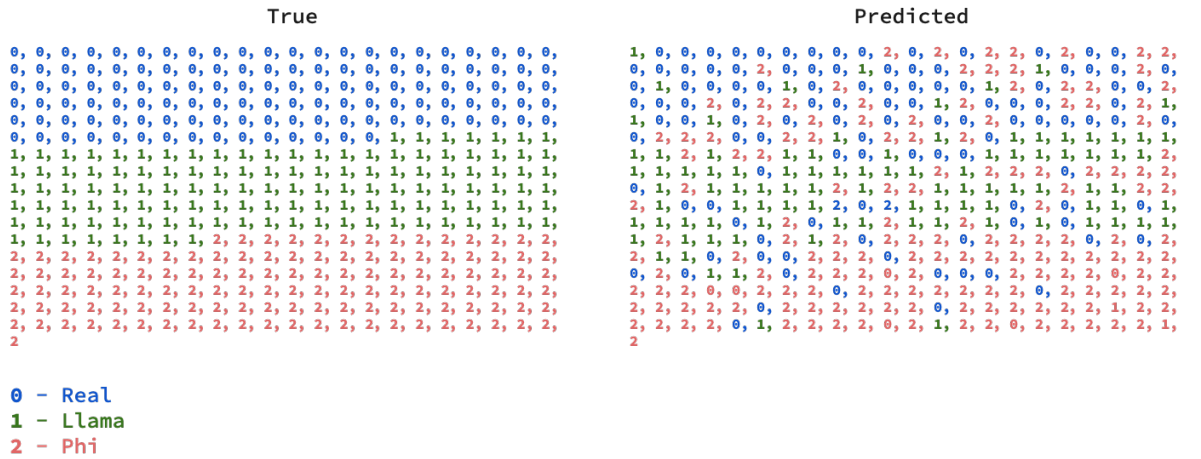

Figure 3.3 — Missclassification illustration for the Recurrent classificator

As for the accuracy (figure 3.2), Recurrent Retention outperformed other models on training data while DeBERTa showed best performance on the validation dataset.

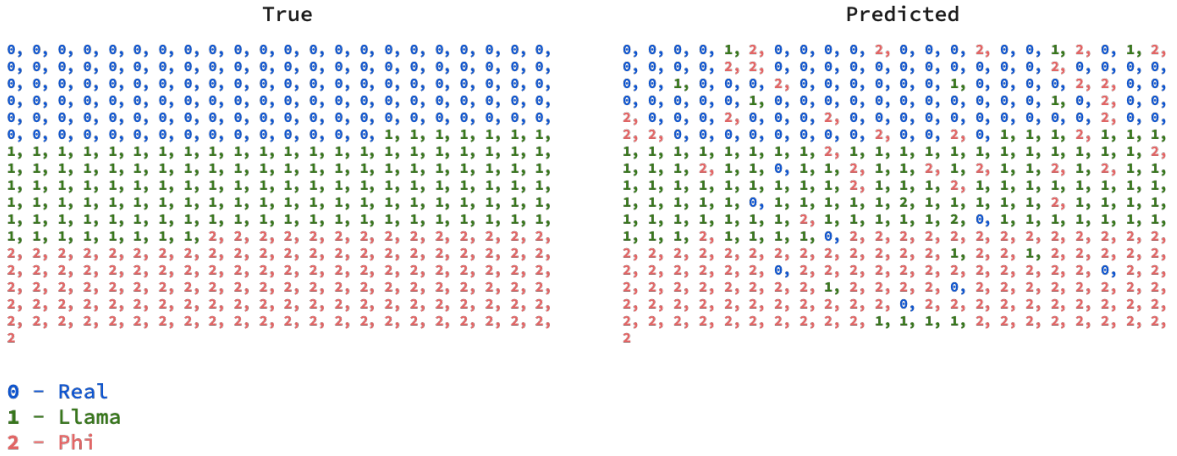

Figure 3.4 — Missclassification illustration for the T5 classificator

Experimental results are presented below for each of the models (figures 3.3, 3.4 and 3.5). Left side is devoted to the true representation of the test dataset. Whereas the right side contains the distribution of predicted classes. It is obvious that recurrent network makes a lot of mispredictions. However, even this result cannot be called accidental, since it is clear that the network captures the general trend, approximating the data distribution.

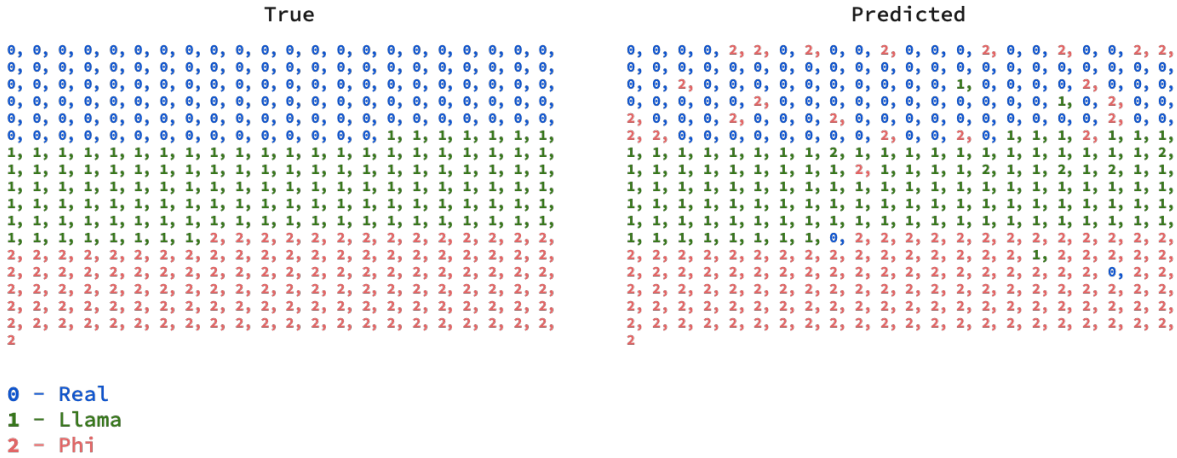

Figure 3.5 — Missclassification illustration for the DeBERTa classificator

T5 model shows much better results that previous one. However, there is still space for improvement.

That is where DeBERTa demonstrates the best result among proposed models. It can be seen that most often the network accepts real answers as the answers generated by Phi network.

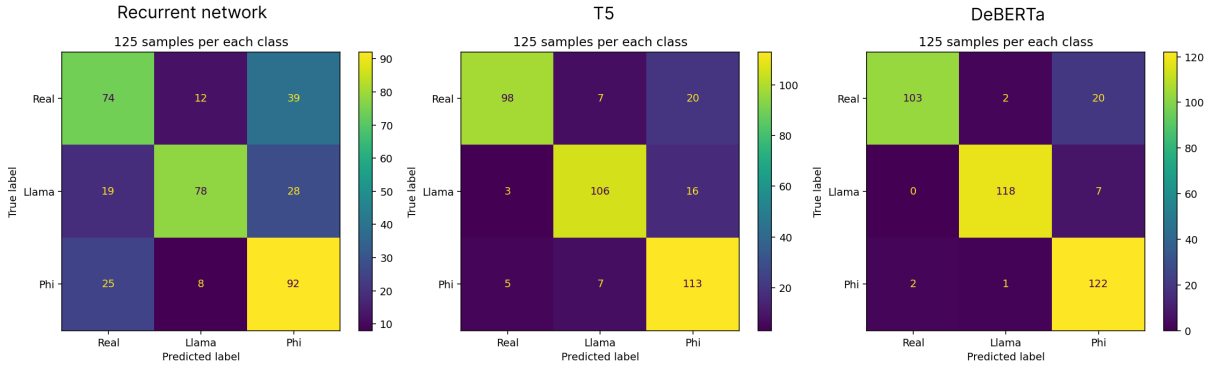

Figure 3.6 — Confussion matrices for all classification models

In order to aggregate results from previous slides we have built confusion matrices shown in figure 3.6. These matrices show pair-wised misclassification for each model and label. Here we can confirm that as I already said all models frequently accept real answers as the answers generated by Phi network.

Table 3.1 — Metrics for the test dataset

	Accuracy	Precision	Recall
Recurrent retention	0.650	0.667	0.650
T5	0.845	0.855	0.845
DeBERTa	0.915	0.925	0.915

To confirm the obtained results with numbers, key metrics such as accuracy, precision and reacall were also calculated. The metrics are available in table 3.1.

CONCLUSION

In the quest to demystify the origins of text and unravel the intricate interplay between human creativity and machine precision, our exploration into neural detection mechanisms has proven both enlightening and challenging. By training and comparing three prominent models — DeBERTa, T5, and Recurrent Network — we sought to navigate the complexities inherent in distinguishing between human-authored and AI-generated text. The results of our analysis paint a vivid picture of the diverse landscape of neural detection.

Undoubtedly, the star of our endeavor was DeBERTa, standing tall as the epitome of neural detection prowess. Its nuanced understanding of contextual intricacies and fine-grained analysis demonstrated the paramount importance of leveraging sophisticated language models for accurate discrimination between human and machine-generated text. DeBERTa's stellar performance serves as a beacon, illuminating the potential for achieving high levels of accuracy in the realm of neural detection.

While T5 exhibited commendable performance, it revealed a slightly diminished accuracy when compared to DeBERTa. The symphony of transformers within T5 resonated with linguistic depth, yet its discernment capabilities hinted at the nuanced challenges embedded in differentiating between human and AI-generated text. T5's performance, although slightly inferior, underscores the ongoing need for refinement and optimization in the pursuit of comprehensive neural detection.

In stark contrast, the Recurrent Network, despite its rhythmic echo reminiscent of human expression, lagged far behind in the race for accuracy. The challenges encountered by this model highlight the intricate nature of capturing the diverse patterns and nuances that define human language. While the Recurrent Network offered a poetic touch, its discernment capabilities fell significantly short in comparison to the formidable DeBERTa.

On the basis of the experimental results and obtained metrics, it is noticeable that the chosen models for detecting AI-generated texts are working well. They not only figure out if a text is AI-generated but can also tell us which model made it. Metrics comparison shows that the DeBERTa model is the top performer - better than the other two in terms of quality, as shown by accuracy, precision and recall metrics and performance visuals.

REFERENCES

1. Neurostat - statistics of generative neural networks. — Yandex, 2023. — Access mode: https://ya.ru/ai/stat.

2. Akram A. An Empirical Study of AI Generated Text Detection Tools. — arXiv, 2023.

3. Elkhatat A. International Journal for Educational Integrity. — Springer Nature, 2023.

4. Mitchell E. Lee Y. Khazatsky A. Manning C. D. Finn C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. — arXiv, 2023.

5. Habibzadeh F. GPTZero performance in identifying artificial intelligence-generated medical texts: a preliminary study. — Journal of Korean Medical Science, 2023. — 38 p.

6. Zhao X. Ananth P. Li L. Wang Y. X. Provable robust watermarking for ai-generated text. — arXiv, 2023.

7. Yahoo. Yahoo asnswers QA. — HuggingFace. — Access mode: https://huggingface.co/datasets/yahoo_answers_qa.

8. Touvron H. Lavril T. Izacard G. LLaMA: Open and Efficient Foundation Language Models. — arXiv, 2023.

9. Li Y. Bubeck S. Eldan R. Textbooks Are All You Need II: phi-1.5 technical report. — arXiv, 2023.

10. He P. Liu X. Gao J. Chen W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. — arXiv, 2020.

11. Devlin J. Chang M. Lee K. Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. — arXiv, 2019.

12. Raffel C. Shazeer N. Roberts A. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. — arXiv, 2019.

13. Vaswani A. Shazeer N. Parmar N. Attention Is All You Need. — arXiv, 2017.