UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Katariina Ingerma

Exploring the Human-like Ability of LLMs in Recognizing Self-generated Text

Bachelor's Thesis (9 ECTS)

Supervisor: Somnath Banerjee, PhD

Tartu 2024

Exploring the Human-like Ability of LLMs in Recognizing Self-generated Text

Abstract:

Large Language model (LLM) is a type of generative artificial intelligence model, that can generate human-like texts. The popularity of LLMs is rapidly increasing every day due to their ability to understand and generate texts that closely resemble human language. This textual content generation ability of LLMs continues to expand their acceptability in professional tasks such as advertising slogan creation, news composition, story generation, etc. The proliferation of LLMs in diverse areas expedites the malicious use of LLMs, which can be a serious threat to information ecosystems and public trust. Therefore, there is an imperative need to develop effective methods to distinguish between LLM-generated and human-written textual content. In this thesis, we have studied the linguistic differences between human-written and LLM-generated texts, the machine-generated text detection performance of an LLM that generates the textual content, and the effect of the textual length on the machine-generated text detection performance. The results reveal that LLMs having fewer parameters generate texts with higher Type-Token-Ratio compared to human-authored texts, while more advanced LLMs exhibit similarities to human writing. According to the obtained results, the more advanced and larger the LLM, the chances are less that it can detect its own generated texts due to their close resemblance to human-authored content. This research is conducive to addressing the problems that arise from the texts produced by LLMs, such as misinformation. It contributes to the development of new approaches for identifying the LLM-generated content.

Keywords:
Generated texts, Text classification, Large language models

CERCS: P170 - Computer science, numerical analysis, systems, control

Uurimus suurte keelemudelite võimest inimese kombel enda poolt genereeritud teksti ära tunda

Lühikokkuvõte:

Suur keelemudel on generatiivne tehisintellekti mudel, mis suudab genereerida inimkeelele lähedasi tekste. Suurte keelemudelite populaarsus kasvab jõudsasti iga päevaga, kuna nad on võimelised mõistma ja geneereerima tekste, mis sarnanevad väga tihedalt inimeste loodud tekstidele. Nende kasutamine on kiiresti levinud erinevates valdkondades, nagu reklaam, loosungite ja uudiste kirjutamine, lugude genereerimine jne. Teisalt levib ka keelemudelite pahatahtlik kasutamine, mis on tõsiseks ohuks infoökosüsteemidele ja avaliku arvamuse usaldusele. Seetõttu on hädavajalik töötada välja meetodeid, mis suudaksid eristada keelemudelite loodud teksti inimeste poolt kirjutatud tekstist. Käesolevas töös uurisime inimeste ja keelemudeli loodud tekstide keelelisi erinevusi, keelemudelite võimet tuvastada tekste, mis on nende endi poolt genereeritud ning teksti pikkuse mõju selle autori tuvastamisel. Tulemused näitavad, et väiksemate parameetritega keelemudelid genereerivad tekste millel on suurem on tekstisõnade ja teksti sõnavara (ingl Type-Token-Ratio) suhe võrreldes inim-autorite kirjutatud tekstidega, kuid samas on rohkem arenenud mudelite tekstidel inimeste kirjutatud tekstiga rohkem sarnasust. Saadud tulemused näitavad ka, et mida arenenum on keelemudel, seda väiksem on tõenäosus, et nad suudavad oma genereeritud teksti tuvastada, sest nende tekst meenutab rohkem inimeste kirjutatud teksti. See uuring on oluline, et mõista suurte keelemudelite loodud tekstidest tulenevaid probleeme nagu valeinfo. See aitab kaasa uute meetodite väljatöötamisele, et keelemudelite tehtud sisu tuvastada.

Võtmesõnad:
Genereeritud tekst, Teksti klassifitseerimine, Suur keelemudel

CERCS:P170 - arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Contents

1 Introduction **5**

2 Background **7**
2.1 Text Generation 7
2.2 Prompt Engineering 8
2.3 Related Works 9

3 Data **11**
3.1 AuTexTification Dataset 11
3.2 GPT-3.5-turbo Dataset 12

4 Methodology **13**
4.1 Model Selection 13
4.2 Prompt Engineering Implementation 14
4.3 Evaluation Metrics 16

5 Results **17**
5.1 Human-written vs LLM-Generated Text 17
5.1.1 Lexical diversity Measures 17
5.1.2 Readability Measures 18
5.1.3 Morphological Analysis 20
5.2 Detection of Machine-Generated Text 21
5.2.1 Detection with BLOOM-1b7 21
5.2.2 Detection with BLOOM-3b 21
5.2.3 Detection with GPT-3.5 Turbo 22
5.3 Length Effect in Text Source Detection 22
5.4 Implementation Details 23

6 Conclusion **25**

References **30**

Appendix **31**
I. Licence 31

1 Introduction

Large Language model (LLM) is a type of generative artificial intelligence model, that can generate human-like texts. The popularity of LLMs is rapidly increasing every day due to their ability to understand and generate texts that closely resemble human language. This textual content generation ability of LLMs continues to expand their acceptability in professional tasks such as advertising slogan creation, news composition, story generation, etc, The proliferation of LLMs in diverse areas expedites the malicious use of LLMs, which can be a serious threat to information ecosystems and public trust. LLMs are being used to distort public opinion by creating fake news, social media posts, or fake product reviews. Even they are being misused in academic settings, for example by students when using them for solving assignments or writing essays. Therefore, there is an imperative need to develop effective methods to distinguish between LLM-generated and human-written textual content.

Due to the rapid advancement of LLMs, the generated texts by LLMs are hard to discern between generated texts and human-written texts. The recent quality of the generated text is such that recent research reported that the state-of-the-art LLMs generated disinformation is more credible than the one generated by humans [SBAG23]. Considering the detection challenges and the evil impact of LLM-generated texts in diverse areas, researchers are fascinated and attracted to work on this topic. This motivated us to pursue this topic.

LLM-generated Text Detection Task: Whether a given text is generated by an LLM can be considered as a binary classification task. We can formally represent this task as:

$$C(x) = \begin{cases} 1 & \text{if } x \text{ generated by LLMs} \\ 0 & \text{if } x \text{ written by human} \end{cases}$$

Where $C(x)$ represents the classifier and x represents the text to be classified.

Research Question Formation: In this thesis, we would like to test one of the human-like capabilities of the LLMs. As human beings, we can easily recognize textual content that has been written by us. Therefore, if a textual content has been generated by an LLM, it could recognize its own generated texts like humans. Hence, according to this assumption, there is a high possibility that the LLMs can recognize their own generated text.

To this aim, in this thesis, we have addressed the following research questions:

- [RQ1] Are there differences in the writing style or pattern of Human authored and LLM-generated textual content?

- [RQ2] Do LLMs have the human-like ability to identify their generated texts?
- [RQ3] Does the length of the LLM-generated text affect its detection?

Thesis Organization: This thesis is divided into six chapters. Chapter 2 provides an overview of text generation techniques, prompt engineering, and discusses related works in the field. Chapter 3 describes the datasets used. Chapter 4 outlines the implementation details of prompt engineering and evaluation metrics for detecting machine-generated text. Chapter 5 presents the results, beginning with human versus LLM text analysis and comparison, followed by the binary classification results and then exploring the effect of length on text source detection. Finally, chapter 6 provides conclusions and discusses potential future research directions.

2 Background

Before giving a brief overview of the machine-generated texts, we present a brief overview of text generation focusing on Large Language Models (LLMs) in Section 2.1. Following that, there will be Section 2.2 about prompt engineering. Finally, we present an overview of related works that classify machine-generated and human-written text.

2.1 Text Generation

Text generation, also known as natural language generation, has been one of the most important sub-fields in natural language processing (NLP). Text Generation aims to produce plausible and readable text in human language from input data. The generated text is expected to satisfy some desired language properties such as fluency, naturalness, and coherence. Text generation can be instantiated into different kinds of applications such as machine translation, text summarization, dialogue systems, story generation, etc. Text generation techniques have evolved significantly, transitioning from traditional methods to the emergence of Transformer-based [V^{+}17] models. Traditional approaches to text generation, such as rule-based or template-based systems, probabilistic models like Markov models or n-grams, and recurrent neural networks (RNNs), have long been utilized. However, a paradigm shift occurred with the rise of LLMs, marking a new era in AI-driven text generation.

A LLM is a generative artificial intelligence model, that has been trained on vast amounts of textual data. LLMs get their name from the large amounts of parameters, variables, weights, and datasets used for training. These models can create human-like text content based on natural language inputs. They can be used for text generation, summarization, translation, classification, and chatbots.

LLMs are founded upon the transformer architecture introduced by Google in 2017 [V^{+}17]. One of the most popular transformer-based text generation model families is the Generative Pre-trained Transformers (GPT)[RNS^{+}18] family. GPT refers to a series of decoder-only language models developed by OpenAI. In the literature, GPT [RNS^{+}18] was the first causal language model for the text generation task. Furthermore, GPT-3 [Bea20] showed that scaling model parameters can significantly improve the downstream generation tasks, with a few examples or prompts. So far, the causal LMs have been widely adopted as the dominated architecture of recent large LMs with over ten billion parameters, such as ChatGPT, GPT-4 [Ope23]. Earlier models, GPT-1 (2018), and GPT-2 (2019) are open-source, but the latest and more advanced GPT-3 (2020), GPT-3.5 models (2022), ChatGPT (2022), and GPT-4 (2023) can be accessed only via API. [M^{+}24]

GPT-3 is trained on 175 billion parameters and has achieved impressive results across zero-shot, one-shot, and few-shot settings, nearly matching the performance of state-of-the-art fine-tuned systems in certain cases. Its strong performance spans multiple NLP tasks, including translation, question-answering, and the cloze tasks, as well as several ones that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, and 3-digit arithmetic [Bea20].

GPT-3.5, a successor to the GPT-3 is the most capable and cost-effective GPT-3.5 model. ChatGPT (Chat Generative Pre-trained Transformer) is fine-tuned from a model in the GPT-3.5 series. ChatGPT is designed to receive instruction in a prompt and generate detailed responses accordingly [Ope22].

In March 2023, the introduction of GPT-4 marked another milestone in the evolution of language models. This model is multimodal and processes both text and image inputs to generate text outputs. Though not surpassing humans in all scenarios, it achieves human-level performance on various benchmarks, like scoring in the top 10 percent of a simulated bar exam. As with previous GPT models, GPT-4 was pre-trained on large text datasets and fine-tuned using Reinforcement Learning from Human Feedback (RLHF) to align with human-desired behaviors [M^{+}24].

BLOOM [1] is an open-access decoder-only Transformer language model trained on the ROOTS dataset comprising 46 natural and 13 programming languages. This model has 176 billion parameters and is very similar to GPT3 in architecture. Its development was coordinated by an open research collaboration BigScience. It was trained on the French public supercomputer Jean Zay, supported by a large-scale public computing grant from GENCI and IDRIS (CNRS). BLOOM models are primarily meant for text-generation tasks. Apart from the 176B parameter BLOOM, there are many smaller BLOOM versions trained on the same dataset, including BLOOM-1b7, BLOOM-3b, and BLOOM-7b1, with 1.7, 3, and 7 billion parameters, respectively [S^{+}23b].

2.2 Prompt Engineering

Prompt engineering is a relatively new field in natural language processing. Prompt engineering is a method that uses text prompts to guide LLMs toward desired results without changing the model's weights or parameters [Cha]. In recent years a standard approach in NLP model learning has been to pre-train and fine-tune. This involves training language models on large datasets and later fine-tuning them with objective functions for specific tasks [L^{+}21]. As of 2021, this approach has been replaced with the "pre-train, prompt, and predict." paradigm, consequently, instead of modifying pre-trained language models to perform tasks like classification, sentiment analysis, machine translation, etc. The tasks are adjusted to resemble those already tackled during the initial training of the model. This is done with the textual prompts. A prompt is a textual input given to a model, from which the model generates the outputs [L^{+}21].

Basic prompts have a few key elements, that can vary through the task [Reg24]. These elements are:

1. Instructions: Detailed instructions for the task or action the model should perform.

[1] https://huggingface.co/bigscience/bloom

2. Context: External data or extra context that aids in the model's comprehension of the task and produces more accurate responses.

3. Input Data: Specific data or information provided to the model as input, which it will process to generate the desired output.

4. Output Indicator: Defined format or type for the desired output. This should indicate how the model should structure its response.

For example, prompt to classify text:

```
    Classify the text into neutral, negative, or positive
Text: I think the food was okay.
Sentiment:
```

This prompt has instructions "Classify the text into neutral, negative, or positive.", input data: "I think the food was okay." and "Sentiment:" for indicating an output. Although context isn't used specifically in this example, it can be added to the prompt to give the model more information. Not every prompt requires all four components and the format may change depending on the particular assignment or model. Organizing the prompts is important so the model can generate accurate responses while effectively conveying the user's intent [Reg24].

LLMs are trained on huge datasets and are tuned to follow instructions. Therefore, they can do some tasks with "zero-shot" settings. This means that they can perform certain tasks without any examples or demonstrations [Sar22].

Whereas, few-shot prompting is a technique that involves providing a small number of examples that include the input and output for the task [Z$^+$21]. Few-shot prompting is also referred to as "in-context" learning as it allows the model to learn without parameter updates. Few-shot prompting allows to "rapidly prototype" LLMs because when altering the prompt it results in a new model. Besides, prompting saves memory and money, when compared to fine-tuning [Z$^+$21].

2.3 Related Works

The advancement of LLMs in generating texts raises the need for the detection of generated content. A study reported that even state-of-the-art LLMs generate disinformation, which is more credible than the one generated by humans [SBAG23]. Considering the detection hardness and the evil impact of LLM-generated texts in diverse areas, shared tasks (e.g., [S$^+$23a],[KHdW$^+$22], etc.) are being organized. Notably, a huge number of research teams participated in these shared tasks. For example, 36 research teams participated in the AuTexTification shared task. Also, there were attempts to create LLM-generated misinformation datasets, such as LLMFake[CS23].

The employed approaches in generated texts are categorized into the following categories:

- Human-assisted Methods: These methods leverage human prior knowledge and analytical skills, providing notable interpretability and credibility in the detection process, such as [ULS+23],[DIK+23],[WLZ+24], etc.

- Zero-shot Methods: Within the zero-shot setup, without the need for additional training through supervised signals. This approach assumes access to LLMs, where their inputs are evaluated based on distinctive linguistic features and statistics. Some notable approaches are DetectGPT [MLK+23], DetectLLM [SZWN23], GLTR [GSR19], [VAR+23],[KKS+03].

- Fine-tuning LMs Methods: These methods involve fine-tuning Transformer-based LMs to discriminate between input texts that are generated by LLMs. [LLC+23], [BGO+19a], [BGO+19b]. TweepFake [FFG+21]

- Adversarial Learning Methods: [H+24], [S+23c], [YJL23], [KKO24] utilizing adversarial learning in generated texts detection.

3 Data

In this study, for LLM-generated text, we planned to test one proprietary LLM and one open-source LLM. To this aim, we considered OpenAI's GPT-3.5-turbo as a proprietary LLM and BigScience's BLOOM as an open-source LLM. Table 1 shows the dataset used in this thesis.

Table 1. Dataset

Written/Generated	Dataset source	Avg. length	#samples
Human	AuTexTification Dataset	336	1000
BLOOM-1B7	AuTexTification Dataset	335	1000
BLOOM-3B	AuTexTification Dataset	292	1000
GPT-3.5-turbo	Prepared by our own	292	1000

3.1 AuTexTification Dataset

The AuTexTification dataset[2] was created for the shared task AuTexTification organized at IberLEF 2023 Workshop. The dataset contains textual samples written by humans as well as generated from prompts using six LLMs. Human-authored texts were collected from datasets MultiEURLEX [CFA21], WikiLingua [L^{+}20], and TSATC [Naj12]. Machine texts were generated with BLOOM models (BLOOM-1B7[3], BLOOM-3B[4], BLOOM-7B1[5]), and three GPT-3 models (babbage, curie, and text-davinci-003).

However, we could not make use of the GPT-based generated texts of this dataset. Because those models have been deprecated by the OpenAI and are not available at this moment. According to the research question of this thesis, we need exactly the same LLMs that were used to generate the textual content. Hence, we had no choice but not to consider the samples generated by the GPT models of this dataset.

The AuTexTification dataset has training and test split, training data has 33845 entries and the testing set has 21800 entries. Training data contains texts from 3 different domains: legal texts, wiki articles, and tweets. However, taking into account the time and computational resource constraints, we used a subset of the AuTexTification dataset. For this thesis, 3000 samples were used from the AuTexTification training set. Specifically, we randomly (to avoid bias) selected 1000 samples from two BLOOM models and humans as authors.

[2] https://huggingface.co/datasets/symanto/autextification2023

[3] https://huggingface.co/bigscience/bloom-1b7

[4] https://huggingface.co/bigscience/bloom-3b

[5] https://huggingface.co/bigscience/bloom-7b1

3.2 GPT-3.5-turbo Dataset

As we could not include the samples by the GPT models in the AuTexTification dataset, we searched for available GPT-based (version 3.5 and higher) datasets. However, we could not find any dataset having generated texts by GPT-3.5 models. Hence, we had no choice but to prepare samples using GPT-3.5. While generating the samples using the GPT-3.5-turbo, we utilized identical prompts as in the AuTexTification dataset. These prompts were then used with the GPT-3.5-turbo model to generate texts. Additionally, while generating the texts, we varied their lengths to match those in the AuTexTification dataset. Moreover, if the model generated longer texts we truncated them to make 630 characters texts.

4 Methodology

This chapter describes the methodology. The first subsection provides an overview of the models and explains the model selection process. Following that, sections propose the approach used in detecting machine-generated text. Afterwards, an overview of the evaluation metrics is provided.

4.1 Model Selection

Taking into account the research question (see Chapter 1), we wanted to employ exactly the same models as classifiers that were used to generate the text samples. Initially, we collected data from the AuTexTification Dataset (ref), in which the generated text samples were generated through BLOOM-1B7 and BLOOM-3B. Hence, we selected BLOOM-1B7 and BLOOM-3B LLMs initially. These models are open-source LLMs. Then, we planned to employ one proprietary LLM. The GPT-3.5 turbo is regarded as the most capable model within the GPT-3.5 series and is much cheaper than the GPT-4 (See Figure 1). Thus, we selected three LLMs:

- BigScience's BLOOM-1b7 (Open-source);
- BigScience's BLOOM-3b (Open-source);
- OpenAI's GPT-3.5 turbo (Proprietary).

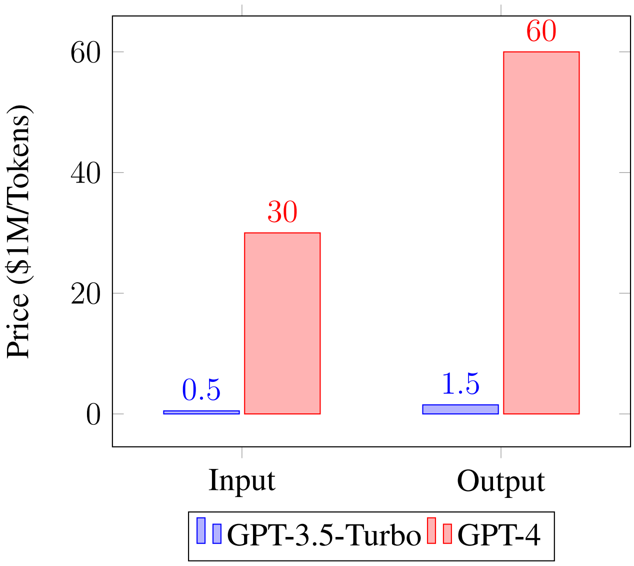


Figure 1. GPT models price comparison [ope]

4.2 Prompt Engineering Implementation

In this work, we obtained the label of a text sample by interacting with an LLM model through a prompt. So, the outcome of the classification task depends highly on the employed prompt. In order to propose an effective prompt for each pre-selected LLM model, we do the following steps:

1. **Test dataset collection:** From the AuTexTification Dataset, we randomly collected 15 samples, of which 5 samples were BLOOM-1b7 generated, 5 samples were BLOOM-3b generated, and 5 samples were Human-written. Also, we included 5 samples that were generated by GPT-3.5-turbo. Thus, we created a test dataset of size 20 samples with equal presence of every dataset. We would like to mention that the samples of this test dataset are not present in the dataset that has been described in Table 1. We considered a small test dataset of size 20 because the LLM models take significant time to process large datasets; moreover, every request costs for GPT models, so it is advisable to evaluate the effectiveness of prompts on a smaller dataset.

2. **Prompt selection:** For a specific LLM model, Start with applying a prompt to the 20-sample testset one by one. Then, we evaluated the prompt based on the following:

   - total number of classified samples out of 20 samples, and the number of samples it failed to classify and generated random output;
   - output consistency;
   - F1 score, accuracy.

3. If this prompt was ineffective, we changed the prompt by using paraphrasing tools such as *QuillBot AI*[6], *ChatGPT*[7], and *Gemini AI*[8] and we started the experiments with the new prompt from Step 2.

4. If the prompt worked well based on our prompt selection criteria, we reformulated the prompt using the same paraphrasing tools mentioned in step 3, to see if we could get even better results on the 20 samples.

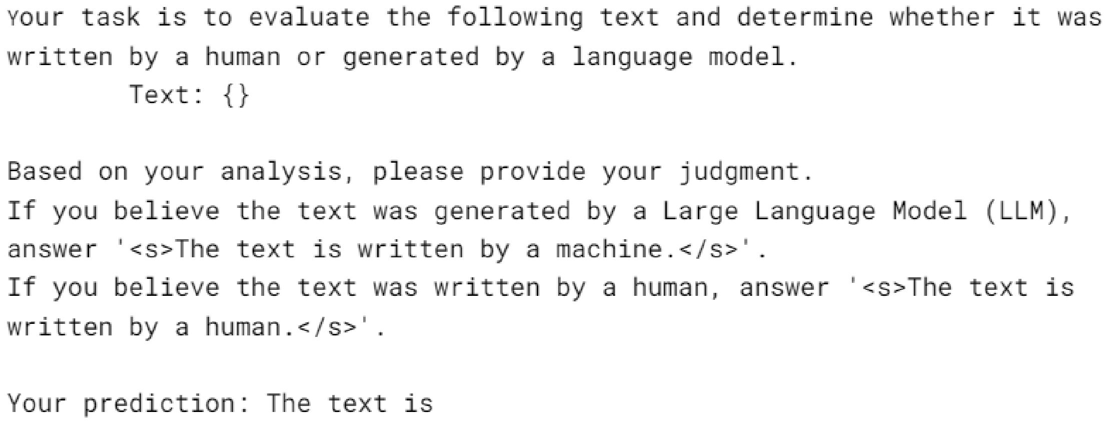
We continued with the above-mentioned iterative process until we found a suitable prompt for a specific LLM. Thus, we obtained the most effective prompt for the 3 LLMs (See Figures 2, 3, and 4). In order to obtain consistent outcomes, we adjusted the temperature parameter of the three predefined LLMs. The temperature values are 0.5 for

[6] https://quillbot.com/

[7] https://chatgpt.com/

[8] https://gemini.google.com/

BLOOM-1b7, 0.2 for BLOOM-3b, and the default value for GPT-3.5-turbo. Throughout the experimentation, we tested 50 prompts for GPT-3.5-Turbo and 38 prompts for BLOOM models. Notably, making prompts for BLOOM models was more challenging than for GPT. BLOOM-1b7 and BLOOM-3b often struggled with understanding the task. In order for the BLOOM models to perform the task specified in the prompt and avoid generating irrelevant text afterward, we found it necessary to provide full-sentence answer options with sentence-beginning and ending indicators (See from prompts 2 and 3).

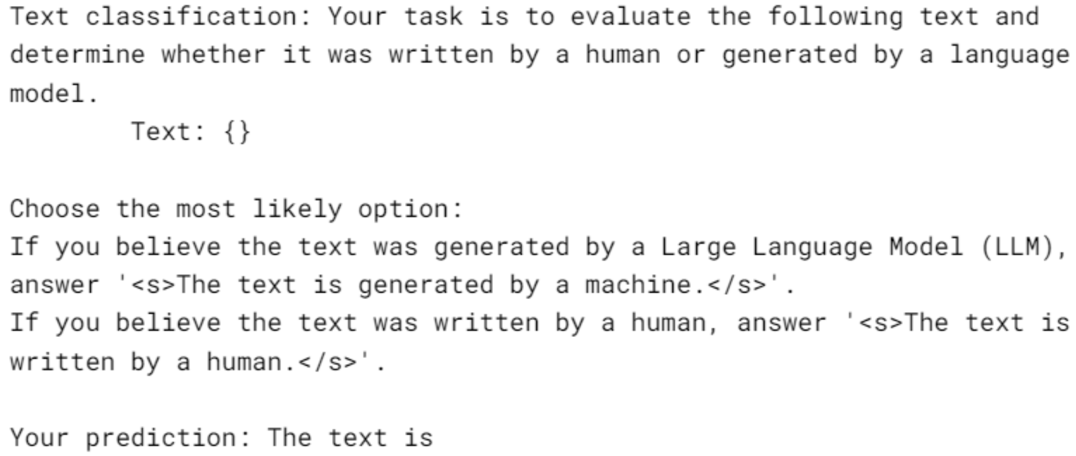


```
Your task is to evaluate the following text and determine whether it was
written by a human or generated by a language model.
        Text: {}

Based on your analysis, please provide your judgment.
If you believe the text was generated by a Large Language Model (LLM),
answer '<s>The text is written by a machine.</s>'.
If you believe the text was written by a human, answer '<s>The text is
written by a human.</s>'.

Your prediction: The text is
```

Figure 2. Prompt Used with BLOOM-1b7

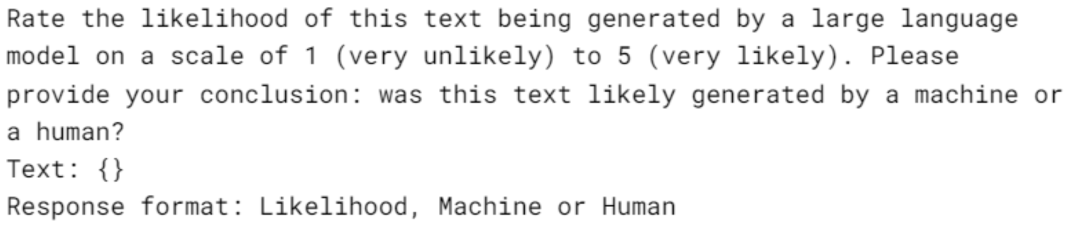


```
Text classification: Your task is to evaluate the following text and
determine whether it was written by a human or generated by a language
model.
        Text: {}

Choose the most likely option:
If you believe the text was generated by a Large Language Model (LLM),
answer '<s>The text is generated by a machine.</s>'.
If you believe the text was written by a human, answer '<s>The text is
written by a human.</s>'.

Your prediction: The text is
```

Figure 3. Prompt Used with BLOOM-3b



```
Rate the likelihood of this text being generated by a large language
model on a scale of 1 (very unlikely) to 5 (very likely). Please
provide your conclusion: was this text likely generated by a machine or
a human?
Text: {}
Response format: Likelihood, Machine or Human
```

Figure 4. Prompt Used with GPT-3.5-Turbo

4.3 Evaluation Metrics

Evaluation metrics are quantitative measures to assess how the machine learning model performs. In our work, the possible results for the classification task are:

- **True Positive** (TP) refers to the text samples (LLM-generated text) correctly classified by the model.
- **False Positive** (FP) refers to the text samples (LLM-generate text) incorrectly classified by the model.

According to our research questions, we are focusing on the true positive of the LLM-generated text samples, specifically the performance of LLMs in identifying their own generated text. Hence, In this work, we can use *Accuracy* as the evaluation metric.

$$\text{Accuracy} = \frac{\text{Number of Correctly Classified Samples (LLM-generated)}}{\text{Total Number of Samples (LLM-generated)}} \tag{1}$$

5 Results

This chapter presents the findings of the linguistic difference between human-written and machine-generated textual content in terms of a few linguistic measures; and the detection of textual content in different scenarios, such as the source of generation and the length of the generated text. The codebase can be accessed from the author's GitHub repository [9].

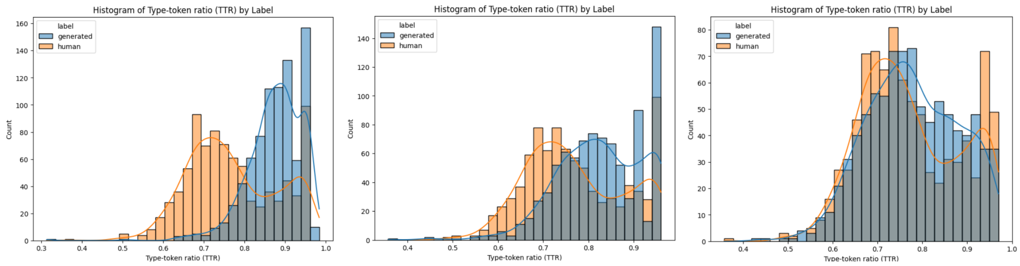
5.1 Human-written vs LLM-Generated Text

For the first research question, the following comparisons were conducted:

1. Human-authored text versus BLOOM-1b7-generated text
2. Human-authored text versus BLOOM-3b-generated text
3. Human-authored text versus GPT-3.5-Turbo-generated text

5.1.1 Lexical diversity Measures

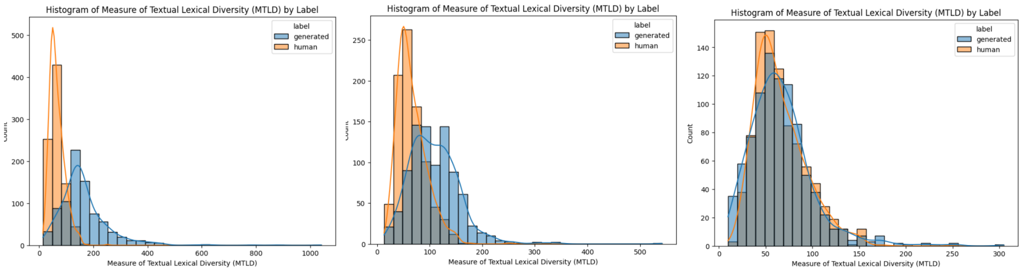
Lexical diversity measures the text quality and linguistic complexity in a text. Type-Token-Ratio (TTR) is a commonly used lexical diversity metric, where a higher value indicates a greater diversity of vocabulary [K+22]. In our analysis (See Figure 5), we found that the TTR range of BLOOM-generated texts appears narrower, with values leaning towards the higher end, indicating that BLOOM-generated texts are overall more lexically complex. Whereas, in human-written texts the TTR is more around the middle of the range, suggesting consistent but less varied vocabulary compared to BLOOM-generated texts. When comparing GPT-3.5-Turbo and human texts, we observe that the TTR of GPT-3.5-Turbo is more similar to that of humans, indicating a comparable level of text complexity.



(a) BLOOM-1b7 (b) BLOOM-3b (c) GPT-3.5-Turbo

Figure 5. Histograms of Type-Token Ratio (TTR) for LLMs vs Human Texts

[9] https://github.com/KatariinaIngerma/thesis



(a) BLOOM-1b7

(b) BLOOM-3b

(c) GPT-3.5-Turbo

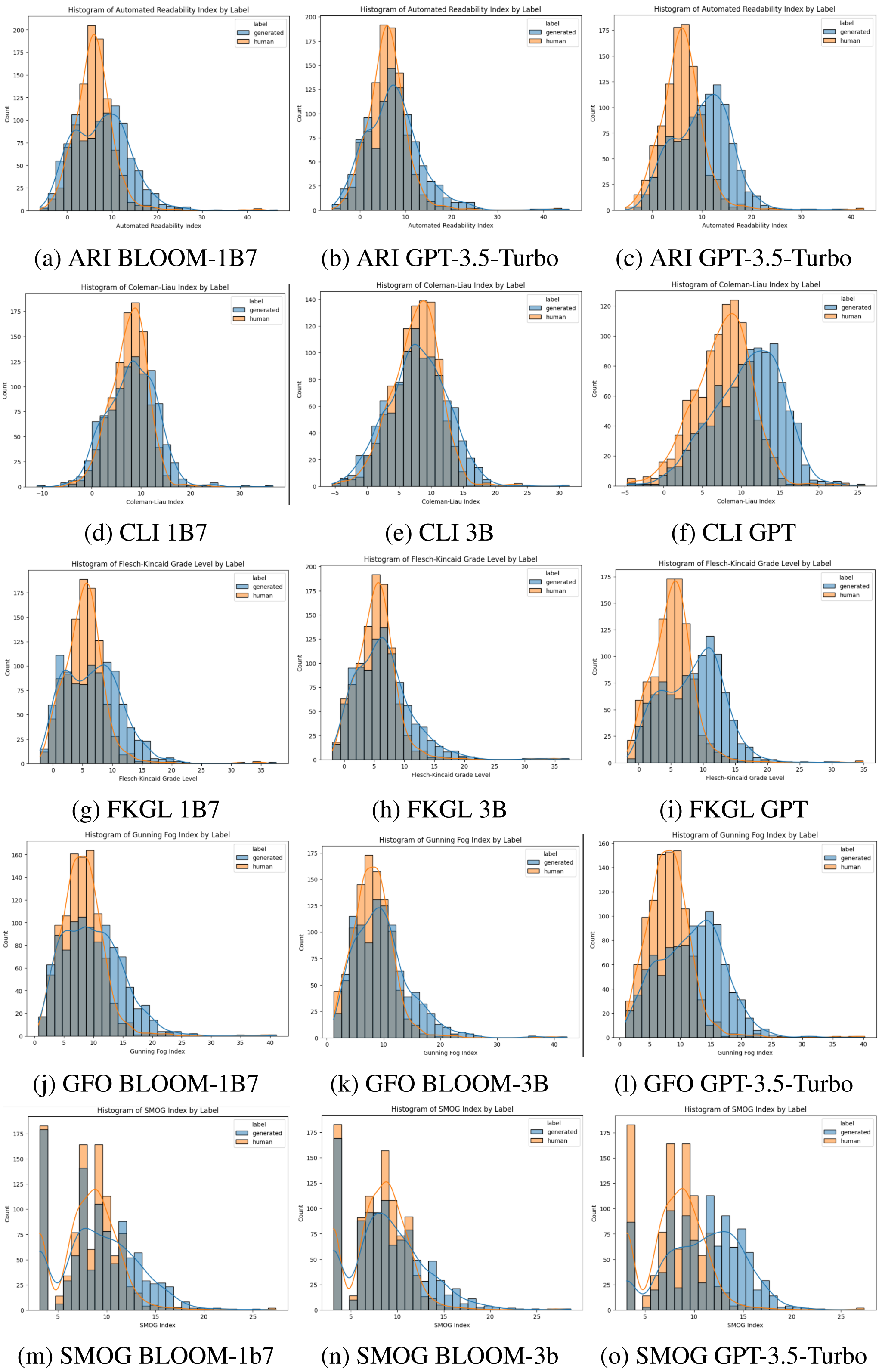
Figure 6. Histograms of Measure of Textual Diversity (MTLD) for LLM vs Human Texts

The same case is for Textual Lexical diversity shown in Figure 6. The Measure of Textual Lexical Diversity metric evaluates the diversity of words and the dispersion of word frequencies in a text [K^{+}22].

Based on the lexical diversity metrics TTR and MTLD the GPT-3.5 texts have the most similar characteristics to human texts. Whereas, the smallest model BLOOM-1B7 results differ significantly, and the BLOOM 3B, with three times more context than BLOOM-1B7 falls in between.

5.1.2 Readability Measures

Additionally, we measured several readability metrics, like the Automated Readability Index (ARI), Coleman-Liau Index (CLI), Gunning Fog Index (GFI), Flesch-Kincaid Grade Level (FKGL), and SMOG Index. Many of these metrics are commonly used in US schools to assess the difficulty of the text. For example, the Automated Readability Index (ARI) score correlates to a specific grade level. ARI scores from 3-4 correspond to fourth-grade reading levels, while 8-10 are suitable for middle school students. Scores of 12-14 are high school reading level, and 16 and above are typically for college level [rea]. Figure 7 illustrates the distribution of readability metrics for the texts generated by all models.



(a) ARI BLOOM-1B7 (b) ARI GPT-3.5-Turbo (c) ARI GPT-3.5-Turbo

(d) CLI 1B7 (e) CLI 3B (f) CLI GPT

(g) FKGL 1B7 (h) FKGL 3B (i) FKGL GPT

(j) GFO BLOOM-1B7 (k) GFO BLOOM-3B (l) GFO GPT-3.5-Turbo

(m) SMOG BLOOM-1b7 (n) SMOG BLOOM-3b (o) SMOG GPT-3.5-Turbo

Figure 7. Comparison of several readability metrics of different LLMs

5.1.3 Morphological Analysis

Furthermore, we compared morphological information, such as part-of-speech (POS) tags and verb tenses. POS tags are used in text analysis to detect which category the word belongs, such as nouns, verbs, adjectives, adverbs, etc. In the context of LLM-written text, we can compare whether LLM text has a similar distribution of POS tags as human-authored text. Figure 8 presents the distribution of Part-of-Speech and Figure 9 presents verb tenses across BLOOM-1b7, BLOOM-3b, and GPT-3.5-Turbo models.

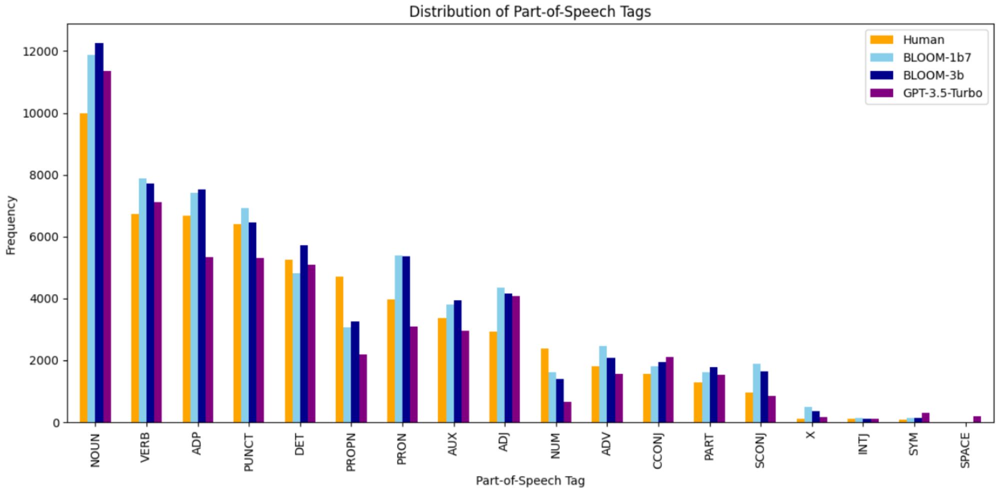


Figure 8. POS tags among human and LLM texts

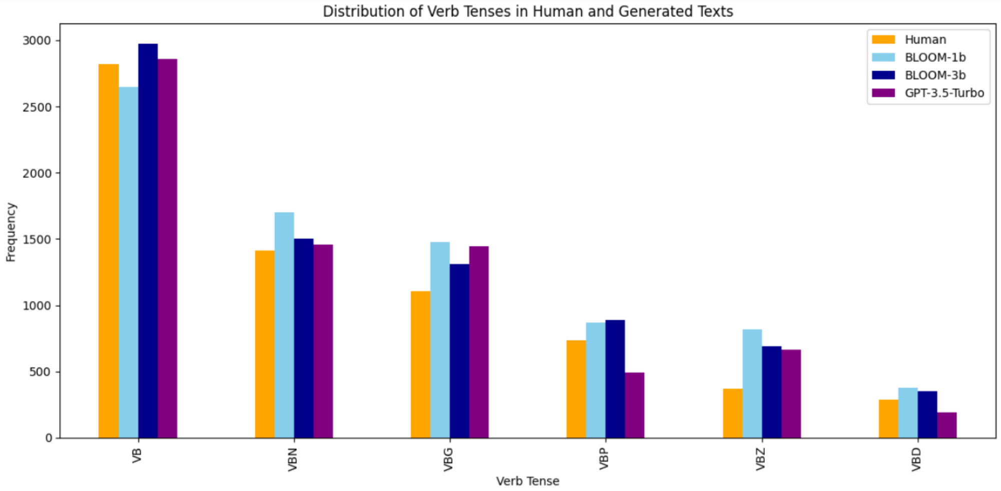


Figure 9. Verb tenses among Human and LLM texts

5.2 Detection of Machine-Generated Text

This subsection reports the findings of the detection capabilities of the three pre-selected LLMs (see Section 4.1) while applying them to the textual content of different sources. Hence, the following subsection presents the detection capabilities of the employed LLMs on the different sources of textual content.

5.2.1 Detection with BLOOM-1b7

Table 2 shows the detection capability of the BLOOM-1b7 model in terms of classification accuracy. We observed the best detection capability (*accuracy*:≈0.64) of the BLOOM-1b7 model when the BLOOM-1b7 model was used as the generator of that text sample. This outcome is expected, as the model could most likely consider their texts to be machine-generated. It detected slightly fewer texts generated by BLOOM-3b and GPT-3.5-Turbo. The model's detection capability declined as the generators are of better LLMs than it. However, BLOOM-1b7 performance was notably lower when identifying human-authored texts, with only correctly classifying 376 samples out of 1000. Potential explanations could be that the prompt used for classification may be biased towards some language patterns that are more common in LLM-generated texts, thereby classifying human-written texts as machine-generated. In addition, the model may fail to analyze the human-written text patterns due to its detection capability.

Table 2. BLOOM-1b7 Classification Results

	Human	Bloom-1b7	BLOOM-3B	GPT-3.5-Turbo
Correctly Identified	376	639	609	600
Incorrectly Identified	623	361	390	399
Accuracy	0.38	**0.64**	0.61	0.60

5.2.2 Detection with BLOOM-3b

The classification results presented in Table 3 reveal how well the BLOOM-3b model performs in classifying different datasets. Being another model of BLOOM, we expected its detection capability to be in line with the BLOOM-1b7 (Table 2). However, in this case, BLOOM-3b generated texts were detected the least, often misclassified as human-authored. Surprisingly, Human-written texts were detected with the highest accuracy (0.58), and a slightly lower accuracy (0.51) was obtained in the GPT-3.5-Turbo dataset. Irrespective of a larger model than the BLOOM-1b7, this model's detection capability is far behind the BLOOM-1b7 model.

Table 3. BLOOM-3b Classification Results

	Human	Bloom-1B7	BLOOM-3B	GPT-3.5-Turbo
Correctly Identified	578	449	429	507
Incorrectly Identified	421	551	571	493
Accuracy	**0.58**	0.45	0.43	0.51

5.2.3 Detection with GPT-3.5 Turbo

The detection performance of the most powerful LLM among the selected models, namely the GPT-3.5-Turbo model, is shown in Table 4. GPT-3.5-Turbo demonstrated a better ability to identify human and BLOOM model-generated texts. The observed performance of the GPT model can be explained by taking into account the fact that the GPT-3.5-Turbo model has more parameters and is trained on more data than BLOOM models and it can generate more human-like texts. It is also expected that the accuracy of detecting BLOOM-1b7 texts is higher than that of the BLOOM-3b model. This is due to a smaller context and less training data in BLOOM-1B7, making the text they generate more distinguishable.

However, the GPT model showed the least effectiveness (accuracy: 0.34) when detecting texts generated by itself. While generating texts like the human-written, its objective is to generate textual content such that it seems like human-written text. This might make the GPT-generated texts to the GPT model more challenging to distinguish them from human-authored texts. Possibly for this reason, the GPT-3.5-Turbo-generated texts are classified mostly as human-authored. Hence, this could be the reason that it could not identify its own generated texts.

Table 4. GPT-3.5-Turbo classification results

	Human	Bloom-1b7	Bloom-3b	GPT-3.5-Turbo
Correctly Identified	551	663	568	337
Incorrectly Identified	449	337	432	663
Accuracy	0.55	**0.66**	0.57	0.34

5.3 Length Effect in Text Source Detection

In this thesis, we also studied the potential impact of the text length on its detection accuracy. We examined the text source detection outcomes across three distinct length ranges:

- short texts (up to 100 characters);
- medium-length texts (101 to 500 characters);
- longer texts (over 501 characters).

From table 5, we can observe that BLOOM-1b7 and BLOOM-3b show higher accuracy when the texts are longer. The same conclusion could not be drawn about the GPT-3.5-Turbo model because its accuracy remains approximately the same for shorter and longer texts.

Table 5. Classification Accuracy Across Different Text Lengths on All Data

	Short Texts	Medium Texts	Long Texts	Dataset
BLOOM-1b7	0.51	0.55	**0.61**	All
BLOOM-3b	0.53	0.45	**0.60**	All
GPT-3.5-Turbo	**0.58**	0.51	0.57	All

Moreover, we explored how the LLM classification accuracy varies using only self-generated data. Table 6 illustrates the fairly consistent accuracy of the BLOOM-1b7 model on self-generated data. For BLOOM-3b, the accuracy is very low in shorter texts and the longer the texts are the better the accuracy is. As we see in Table 4, GPT model does not recognize its own generated texts very well, resulting in low accuracy on all text lengths. From Table 1 and Table 5 Table 6, we can conclude that the BLOOM-3B model works well in the detection of long text rather than short text.

Table 6. Classification Accuracy Across Different Text Lengths on Self-Generated Data

	Short Texts	Medium Texts	Long Texts	Dataset
BLOOM-1b7	**0.68**	0.62	0.66	BLOOM-1b7
BLOOM-3b	0.24	0.38	**0.65**	BLOOM-3b
GPT-3.5-Turbo	**0.36**	0.35	0.28	GPT-3.5-Turbo

5.4 Implementation Details

Practical tasks were conducted using Google Colab Pro[10] with Jupyter Notebooks utilizing the Python programming language [VRD09]. The choice of Colab Pro was due to its higher RAM capacity and longer runtime, which is necessary for handling the computational demands of the tasks. The Python libraries employed were Pandas [M+10]

[10] https://colab.research.google.com/

for data manipulation, Transformers [Wea20] for BLOOM model implementation, LexicalRichness [She22], and spaCy [HM17] for text analysis. The GPT model was accessed and utilized through the OpenAI API [11].

[11] https://platform.openai.com/docs/api-reference

6 Conclusion

The human-like textual content generation ability of LLMs continues to increase their popularity on a daily basis. On the other hand, motivated malicious users leverage this high-quality and human-like content generation ability of LLMs to generate texts in diverse areas, which can be a serious threat to information ecosystems and public trust. In this context, this bachelor's thesis focuses on the topic of LLM-generated text detection. In this thesis, we have studied the linguistic differences between human-written and LLM-generated texts, the machine-generated text detection performance of an LLM that generates the textual content, and the effect of the textual length on the machine-generated text detection performance. We utilized the AuTexTification Dataset for data collection and generated samples for the GPT-3.5 Turbo model. In this study, we employed two BigScience's BLOOM models (open-source) and OpenAI's GPT-3.5-turbo (proprietary LLM) as LLMs in zero-shot settings. Here, we summarize our findings by answering the three research questions which are formed at the beginning of this thesis.

RQ1. Are there differences in the writing style or pattern of Human authored and LLM-generated textual content?

According to the obtained results in Section 5.1.1, it appears that smaller LLMs, such as BLOOM-generated texts exhibit greater lexical complexity when compared to human-written texts. In contrast, texts generated by more advanced LLMs like GPT-3.5-Turbo the lexical complexity is very similar to that of human-written texts. Considering readability measures, we observe variations in several metrics. Moreover, LLM and human written texts exhibit differences in morphological information.

RQ2. Do LLMs have the human-like ability to identify their generated texts?

Based on our classification results in Section 5.2, it appears that LLMs like GPT-3.5-Turbo have limited human-like ability to identify their own generated text because the text they generate resembles closely human-written text. A less capable model, BLOOM-1b7 demonstrated some level of ability to recognize their own generated texts. Finally, BLOOM-3b detected some portion of the texts correctly, still its detection capability was behind BLOOM-1b7 when detecting their own generated texts.

RQ3. Does the length of the LLM-generated text affect its detection?

The potential impact of the text length on its detection accuracy is depicted in Table 5 and Table 6. BLOOM-1b7 model accuracy remained consistent when detecting shorter and longer texts, while BLOOM-3b detection accuracy improved significantly when texts were longer. As for GPT-3.5-Turbo, the model that struggled the most to detect its own generated texts resulted in low accuracy on all text lengths. However, we observed significant improvement when the GPT model was involved in classifying short texts.

Limitations: The limitations of this study are:

- Considering the time frame of this bachelor's thesis, we used a fair enough dataset (4000 samples) to conclude this study. However, the results could be more accurate with a larger dataset.
- The dataset from which we collected BLOOM and Human samples contains texts from three domains (tweets, legal, and how-to articles). Hence, our study is restricted to these three domains only.
- One of the limitations of this work could be only employing three LLMs.
- Moreover, there are many lexical diversity metrics, and because we only looked at some of them, we might have missed some important information regarding LLM vs. human-written text.

Future Work: As part of the future work, to obtain more accurate results, the limitations of this thesis needed to be addressed:

- In the future, we would like to study the research questions on larger datasets employing a number of proprietary and open-source LLMs. Besides GPT and BLOOM, there is a wide range of LLMs, both proprietary and open source, that can be tested.
- Additionally, exploring further lexical diversity metrics could offer a more comprehensive analysis of the differences between human-authored and LLM-generated texts.
- Regarding few-shot learning, it did not fit into our current time limit, but it is a technique with the potential to improve the prompt engineering approach.
- Besides English, non-English text samples would be included in the dataset, specifically some European languages.

References

[Bea20] Tom Brown and et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[BGO$^+$19a] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*, 2019.

[BGO$^+$19b] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text, 2019.

[CFA21] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[Cha] Aman Chadha. Prompt engineering. `https://aman.ai/primers/ai/prompt-engineering/`. (04.28.24).

[CS23] Canyu Chen and Kai Shu. Can llm-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*, 2023.

[DIK$^+$23] Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771, 2023.

[FFG$^+$21] Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):1–16, 05 2021.

[GSR19] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M 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.

[H$^+$24] Xinlei He et al. Mgtbench: Benchmarking machine-generated text detection, 2024.

[HM17] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[K$^+$22] Noemi Kapusta et al. Assessing the linguistic complexity of german abitur texts from 1963–2013. In *Conference on Natural Language Processing*, 2022.

[KHdW$^+$22] Yury Kashnitsky, Drahomira Herrmannova, Anita de Waard, Georgios Tsatsaronis, Catriona Fennell, and Cyril Labbé. Overview of the dagpap22 shared task on detecting automatically generated scientific papers. In *Third Workshop on Scholarly Document Processing*, 2022.

[KKO24] Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples, 2024.

[KKS$^+$03] Leonid A Kalinichenko, Vladimir V Korenkov, Vladislav P Shirikov, Alexey N Sissakian, and Oleg V Sunturenko. Digital libraries: Advanced methods and technologies, digital collections. *D-Lib Magazine*, 9(1):1082–9873, 2003.

[L$^+$20] Faisal Ladhak et al. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization, 2020.

[L$^+$21] Pengfei Liu et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021.

[LLC$^+$23] Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. Deepfake text detection in the wild, 2023.

[M$^+$10] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[M$^+$24] Shervin Minaee et al. Large language models: A survey, 2024.

[MLK$^+$23] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR, 2023.

[Naj12] Ibrahim Naji. TSATC: Twitter Sentiment Analysis Training Corpus. In *thinknook*, 2012.

[ope] OpenAI API Pricing. https://openai.com/api/pricing/. (10.05.2024).

[Ope22] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022.

[Ope23] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.

[rea] The automated readability index. https://readable.com/readability/automated-readability-index/. (14.05.2024).

[Reg24] Aishwarya Naresh Reganti. Applied llms mastery 2024, 2024. 01.05.2024.

[RNS+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[S+23a] Areg Mikael Sarvazyan et al. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, 2023.

[S+23b] Teven Le Scao et al. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[S+23c] Zhouxing Shi et al. Red teaming language model detectors with language models, 2023.

[Sar22] Elvis Saravia. Prompt Engineering Guide. https://github.com/dair-ai/Prompt-Engineering-Guide, 12 2022.

[SBAG23] Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. Ai model gpt-3 (dis) informs us better than humans. *Science Advances*, 9(26):eadh1850, 2023.

[She22] Lucas Shen. LexicalRichness: A small module to compute textual lexical richness, 2022.

[SZWN23] Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, 2023.

[ULS^{+}23] Adaku Uchendu, Jooyoung Lee, Hua Shen, Thai Le, Dongwon Lee, et al. Does human collaboration enhance the accuracy of identifying llm-generated deepfake texts? In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 11, pages 163–174, 2023.

[V^{+}17] Ashish Vaswani et al. Attention is all you need. volume 30. Curran Associates, Inc., 2017.

[VAR^{+}23] Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis. *arXiv e-prints*, pages arXiv–2305, 2023.

[VRD09] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[Wea20] Thomas Wolf and et al. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[WLZ^{+}24] Luoxuan Weng, Shi Liu, Hang Zhu, Jiashun Sun, Wong Kam-Kwai, Dongming Han, Minfeng Zhu, and Wei Chen. Towards an understanding and explanation for mixed-initiative artificial scientific text detection. *Information Visualization*, page 14738716241240156, 2024.

[YJL23] Lingyi Yang, Feng Jiang, and Haizhou Li. Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text, 2023.

[Z^{+}21] Tony Z. Zhao et al. Calibrate before use: Improving few-shot performance of language models, 2021.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Katariina Ingerma**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Exploring the Human-like Ability of LLMs in Recognizing Self-generated Text,
   *(*title of thesis)

   supervised by Somnath Banerjee.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Katariina Ingerma
15/05/2024