University of Tartu
Institute of Computer Science
Computer Science Curriculum

Martin Kõnnussaar

Exploring the Capability of Large Language Models to Detect AI-generated Academic Texts

Bachelor's Thesis (9 ECTS)

Supervisor:
Somnath Banerjee, PhD

Tartu 2024

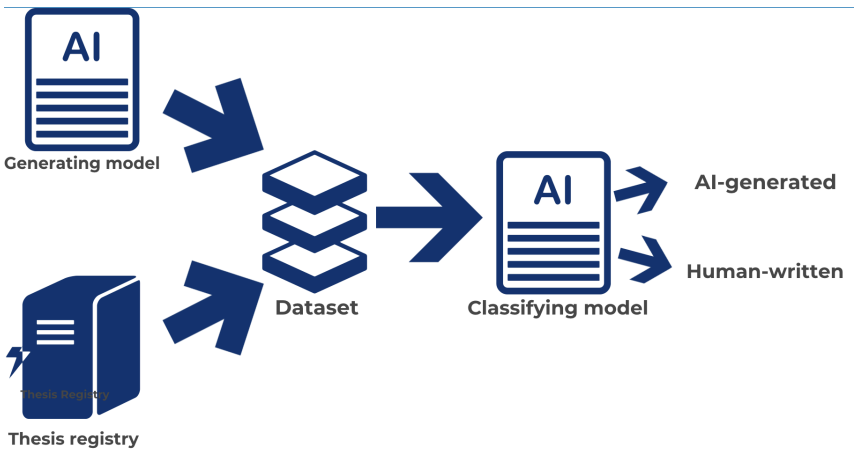
Exploring the Capability of Large Language Models to Detect AI-generated Academic Texts

Abstract:

The increasing prevalence of Large Language Models (LLMs) poses significant challenges to authorship verification, particularly in the academic context. This thesis addresses this challenge by evaluating state-of-the-art LLMs (Claude 3.7 Sonnet, Gemini 2.5 Pro, Deepseek R1, o4-mini) to classify academic abstracts as AI-generated or human-written. A novel bilingual dataset comprising of human-authored and LLM-generated abstracts was created as part of this study. Our investigation focused on three crucial factors: (1) language resource availability, (2) size of the LLM that generates the content, and (3) prompting techniques. The language resource availability influences detection performance, where the LLMs generally perform better on English (a high-resource language) than Estonian (a low-resource language) abstracts. The size of the generating LLM also proved significant; smaller models were more detectable than larger ones. While metacognitive and chain-of-thought prompting techniques demonstrated strong results, no single strategy proved universally superior. Detecting LLM-generated academic abstracts is a nuanced challenge; however, currently available LLMs demonstrate strong potential as detectors. Nevertheless, their varied effectiveness emphasizes the need for continued research and tool development to safeguard academic integrity in the era of advanced AI.

Keywords: Large Language Models, AI Text Detection, Academic Integrity

CERCS: P176 Artificial Intelligence



Uuring Suurte Keelemudelite võimekusest tuvastada AI-genereeritud akadeemilisi tekste

Lühikokkuvõte:

Suurte keelemudelite (LLM) kasutus on viimastel aastastel hüppeliselt kasvanud, mis mõjutab muuhulgas akadeemilises kontekstis tekstide autorsuse kontrollimise keeruliseks. Antud uurimistöö uurib, kui efektiivsed on uusimad LLMid (Claude 3.7 Sonnet, Gemini 2.5 Pro, Deepseek R1, o4-mini) eristades inimeste poolt kirjutatud ja LLMide poolt genereeritud uurimistööde lühikokkuvõtteid. Selleks eesmärgiks loodi uus kahekeelne andmestik AI-genereeritud ja inimkirjutatud lõputööde lühikokkuvõtetest. Lisaks uuriti, kui palju mõjutavad LLMide tuvastussuutlikkust keeleressursside olemasolu, teksti loova mudeli suurus ja kasutatud viipamistehnika. Keeleressursside olemasolu mõjutas tuvastustäpsust märgatavalt, mudelite täpsus oli kõrgem ingliskeelsete lühikokkuvõtete kui eestikeelsete lühikokkuvõtete puhul. Ka teksti genereeriva LLMi suurus osutus oluliseks faktoriks; väiksemad mudelid olid kergemini tuvastatavad kui suuremad. Kuigi metakognitiivsed ja mõttepõhised arutluskäigu (Chain-of-Thought) viipamistehnikad andsid häid tulemusi, ei osutunud ükski strateegia üldiselt parimaks. LLMide poolt genereeritud akadeemiliste lühikokkuvõtete tuvastamine on keeruline väljakutse, mille täpsust mõjutavad keel, teksti genereeriv mudel ja viipamistehnika, mida kasutatakse. Kuigi uusimatel LLMidel on potentsiaali tuvastusvahenditena, rõhutavad nende puudujäägid vajadust jätkuva uurimistöö ja tööriistade arendamise järele, et kaitsta akadeemilist terviklikkust kõrgtehnoloogilise tehisintellekti ajastul.

Võtmesõnad: Suured keelemudelid, AI teksti tuvastamine, Akadeemiline ausus

CERCS: P176 Tehisintellekt

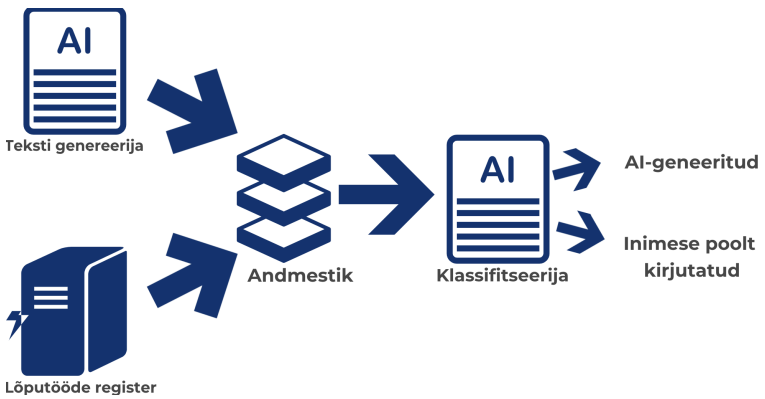


Table of Contents

1. Introduction

As Large Language Models (LLMs) have grown in their potency, so has their popularity. Semrush[1] and Similarweb[2] have ChatGPT placed at the ninth and seventh most visited websites on the internet. LLMs are used for anything language-based, and nowadays LLM-based agents can do anything that a human could on a computer. AI is predicted to severely change the landscape of the global economy by giving people tools that can speed up their work in most sectors, and in many cases, make some jobs completely obsolete. The same goes for academia: LLMs can be used for things like research assistance, writing aid or data analysis to considerably shorten the time between research being done and a paper being published.

However, though LLMs' text generation capability is useful for many fields, it comes at a cost: verifying integrity, assessing originality and the evaluation of students' work and research contributions is now harder than ever. Detecting AI-generated text is essential for maintaining originality, ethics, and accuracy in academic settings. Prohibiting AI-generated content in assignments encourages students to develop critical thinking skills by expressing their own ideas. Relying on AI instead of personal effort can violate academic integrity, equating to plagiarism, which undermines trust in academic systems. Additionally, AI systems sometimes generate "hallucinations"—factually incorrect or fabricated claims—which can lead to misinformation in research and public discourse. Proper citation practices are also at risk, as AI may not accurately attribute sources, resulting in unintentional intellectual theft. Therefore, detection tools are crucial for protecting educational objectives, upholding ethical standards, preventing misinformation, and ensuring the production of reliable, well-credited knowledge.

Wide use of LLMs also cause another issue: having an option to take a shortcut and skip all of the hard work and figuring out how things work or how to make them work is often very tempting, but for example, when writing a paper in high school, the purpose is not to have a paper written about a topic, but to have the student learn to look for and analyze sources, write a narratively connected and logically sound text and come to conclusions based on that.

[1] https://www.semrush.com/trending-websites/global/all

[2] https://www.similarweb.com/top-websites/

Having LLMs do mentally taxing or effort-requiring tasks for students leads to them lacking skills and traits they might later wish they had acquired.

Research has been conducted into detecting LLM-generated text, with DetectGPT (Mitchell et al., 2023) detecting up to 97.88% of AI-generated texts, though the rate of false positives was also high.

A common issue is that even though a method might work on the datasets it's tested on, in real world use the tool's efficacy drops considerably, as it will be used for detection in languages or domains that are outside what the tool was trained or tested on. Because of this, most commercially available tools for that purpose have high false positive rates and are generally considered unreliable. Additionally, LLMs are developing at such a rapid pace that developing a solution for a single version of a model is not feasible, as the next big thing will likely come out very soon after a solution is developed. Generalist solutions are also held back by the rapid improvements in the AI space, because LLMs write more and more like humans.

LLM-generated Academic Text Detection Task: Whether a given academic textual content 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 academic text to be classified.

Research Question Formation: In this thesis, we investigate the detection of AI-generated academic text, focusing on three variables: i) language resource availability, ii) scale of generative AI models, and iii) diverse prompting strategies.

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

> **RQ1. Language resource impact:** How do disparities in linguistic resource availability (high-resource English vs. lower-resource Estonian) across languages influence the performance and generalizability of LLM-based detectors in identifying authentic academic textual content?

RQ2. Model Size vs. Detectability Trade-off: How does the parameter count of LLMs (e.g., GPT-4o mini's compact design vs. Claude 3.5 Haiku's balanced architecture) influence the detectability of their generated thesis abstracts, particularly in low-resource Estonian compared to high-resource English?

RQ3. Prompting technique impact: How does the choice of prompting technique influence the detectability of machine-generated academic textual content, and to what extent does language resource availability (high-resource English vs. lower-resource Estonian) impact their detection under these prompting strategies?

The contributions presented in this work are:

1. **Novel bilingual dataset:** We created a bilingual dataset for the AI-generated academic textual content detection task in English (a high-resource language) and Estonian (a resource language). The dataset consists of 577 human-authored samples and 577 AI-generated samples for each language.
2. **Empirical evaluation of LLMs as AI-generated text detectors:** We studied different LLMs' (namely- Claude 3.7 Sonnet, o4-mini, Gemini 2.5 Pro Preview (March 25) and Deepseek R1) capability to check the authenticity of academic textual content in two settings: low-resource (Estonian) and high-resource (English).
3. **Investigation of factors influencing detection performance:** We analyzed the impact of three aspects impacting the performance of LLMs in detecting AI-generated texts in the context of academic textual content:
    1. **Language resource availability:** Comparing LLMs' AI-generated content detection performance in two settings: English (a high-resource language) and Estonian (a low-resource language).
    2. **Model scale:** Evaluating how the size (number of parameters) of a model impacts the detectability of text generated by it.
    3. **Prompting strategies:** Evaluating the effectiveness of different prompting techniques (Chain-of-Thought and Metacognitive prompting), zero-shot and few-shot prompting and different ways of splitting the text before evaluating it.

Availability: The codebase can be accessed from the author's GitHub repository[3].

[3] https://github.com/martink6nnu/LLM-academic-text-detection

2. Background and Related Work

This chapter provides a baseline understanding for what LLMs are and the different ways of improving their performance in specific tasks. It also covers previous research regarding AI-generated text detection techniques.

2.1 Large Language Models

LLMs are machine learning models designed for natural language processing tasks, most commonly text generation. Though initially used for simple text generation, these days they can be used for most things one would do on a phone or computer. According to Orq.ai[4] LLMs have use cases in the healthcare, finance, education, legal and customer service, but also agriculture, human resources and cyber security. LLMs even power integrated development environments, helping users in designing and writing code[5]. LLMs are interacted with using "prompts", querying the model with a message to receive a response. Most commercial models are trained to follow instructions given in the prompt.

Most modern LLMs are based on the transformer architecture, which is a deep learning architecture proposed by (Vaswani et al., 2017) in "Attention is all you need". The architecture works by converting input text into tokens or numeric embeddings. Since the model reads inputs parallelly, to take positions of tokens into account it adds positional encodings to embeddings. Then it uses multi-head self-attention layers for detecting connections between tokens, using that information to output the most likely next token.

From this initial transformer architecture, several model families emerged, notably the Generative Pre-trained Transformers (GPTs) developed by OpenAI[6], with the first GPT model released in 2018. These models use a decoder-only variant of the transformer architecture, focusing on text generation rather than the translation tasks that encode-decode transformers were designed for. GPTs are pre-trained on massive text corpora and then often fine-tuned to follow human instructions. For example, OpenAI's GPT-3 model was trained on 499 billion tokens, equivalent to around 2.5 million 300-page books in English(Brown et al.,

[4] https://orq.ai/blog/llm-use-cases
[5] Some examples of AI-powered IDEs include GitHub Copilot, Cursor, Windsurf, Cline.
[6] https://openai.com/

2020). Other notable decoder-only transformer models include Anthropic's Claude[7] and Meta's Llama[8] as well as Google's Gemma and Gemini[9] models.

2.2 Detecting AI-generated Text

An increase in AI use also necessitates the development of a reliable detection tool for AI-generated content. The necessity comes from multiple issues: an increasing amount of misinformation generated with AI and spread on the internet, a risk of recursive degradation for training future models and loss of trust in digital communications (Wu et al., 2025). Loss of trust in digital communications includes communications in academic institutions, e.g. submitting essays online, as a paper written at home could easily be generated by a LLM but confirming that an AI did write the paper is difficult, as LLMs are trained to generate human-like text. Though the first commercially available models were not all that capable, current LLMs can generate content mostly indistinguishable from human-written texts. There have, however, been multiple approaches to doing just that.

One approach has been analyzing the language used in generated texts and spotting patterns that are not found in human-written texts. (Gehrmann et al., 2019) developed Giant Language model Test Room (GLTR), a tool that gets the probabilities of each word in the text from a model (GPT-2) as well as the probability of that word being there compared to the top probability word and the entropy per word. It does the same with BERT for bidirectional probabilities and then visualizes the results for users. We would note that this likely works worse on texts generated by newer, bigger models as the models used in GLTR are far smaller and less capable than current options, leading to token choices that better emulate human-written texts and thus changing the token probabilities considerably. found that vocabulary has shifted more since ChatGPT's release than earlier major events such as the COVID-19 pandemic, specifically pointing out use of words like "delves", "intricate", "pivotal", "comprehensive", and "noteworthy".

Another approach is to use LLMs, or tools bundled with LLMs for detecting generated text. (Mitchell et al., 2023) created DetectGPT, a tool based on the theory that when rewriting a text generated by an AI, nearly all changes will lead to a decrease in the probability curve of that model for that text. This, however, requires access to the scoring systems of the models,

[7] https://claude.ai/
[8] https://www.llama.com/
[9] It is important to note that while the specific architectural details of the Gemini models are not publicly disclosed, Google has stated that the Gemma models are built on the same research and technology as Gemini, suggesting a shared foundational architecture. Gemini can be found at https://gemini.google.com/

which is often not available for commercial models. RADAR (Hu et al., 2023) is a tool that works by pre-training two separate models, one model for differentiating between purely AI-generated texts, paraphrased texts and human-written texts, and the other model for generating text. The generating model is then rewarded for fooling the detection model and the detection model is rewarded for correctly classifying the text. (Kirchenbauer et al., 2024) proposed a different approach where the model is instead incentivized to semi-randomly use some words over others, thus making the detection process simpler by having the evaluator only need to consider the amount of "green" words over "red" words for the same generation seed. In our assessment, for this strategy to work, it would need all LLM providers to use the same system for generation and provide the seeds publicly, which is not likely to happen from sheer good-will.

2.3 Prompt Engineering

It was quickly discovered that the prompt could considerably change the response by the LLM. Prompt engineering is a process in which the user words the prompt in a way that makes the LLM perform better or output the answer in a specific way for the requested task. The process is often iterative: an initial prompt is designed, then inference is run on the dataset with the prompt, results are analyzed and then a new prompt is created to improve on the shortcomings of the previous prompt. A commonly used example of prompt engineering is few-shot prompting, where examples of the process are shown before asking the model to work on a similar task.

Vatsal and Dubey (2024) compiled a survey of prompt engineering techniques. Of those, some are applicable to our use case. Chain-of-Thought (CoT) prompting makes the LLM "think" by having it write out step-by-step logic for finding the final answer. Plan-and-solve (PS) prompting makes the LLM initially come up with a plan by breaking the larger task into multiple smaller steps and then complete those steps sequentially. It should be noted that most current state-of-the-art models already have reasoning built in, so in these cases it doesn't end up making too big of a difference, but CoT prompting gives considerable quality boosts to the responses of non-reasoning models. (X. Wang et al., 2023) developed self-consistency prompting, which changes the default greedy decoder to instead sample from the model's decoder by first changing the temperature and top-k parameter, then querying the model with the same prompt multiple times, eventually getting multiple diverse reasoning paths. The responses are then parsed, and the most common final answer is chosen. (Y. Wang

and Zhao, 2024: 3) designed metacognitive prompting (MP), where the prompt tells the LLM to solve the task in 5 steps: “1) understanding the input text, 2) making a preliminary judgment, 3) critically evaluating this preliminary analysis, 4) reaching a final decision accompanied by an explanation of the reasoning, and 5) evaluating the confidence level in the entire process,” improving the average performance across all models and tested datasets.”

3. Methodology

This chapter covers the methodology of each part of the research process, providing an overview of the processes of dataset building, prompt engineering, and evaluating results.

3.1 Data Collection: Human-authored Dataset

As this research focuses on authenticating academic texts, we required a corpus of academic texts. We opted to use abstracts of theses from the University of Tartu's Institute of Computer Science's thesis registry from the years 2015-2018. We chose this time span because we hypothesized that the abstracts written before 2018 were unlikely to have been generated or changed by text-generating AI. Had we used abstracts from after 2018, we'd have run into the risk of including fully or partially AI-generated texts in the human-written part of our dataset, which might lead to issues in assessing the classification task's results.

We planned to investigate the impact of linguistic resource availability during the classification task, so we collected abstracts in both English (representing a high-resource language) and Estonian (representing a lower-resource language).

For the abstract collection, a Python script was created. The script, using the Beautiful Soup[10] library to extract data from HTML files, made requests to the thesis registry[11], getting a list of links to the theses published in the time span specified in the script. Then, it made queries to each link in the list, extracting the author's name, the thesis' title, ID and the abstract, writing them to a file.

In total, 577 abstracts were collected in both English and Estonian, leading to a total of 1144 human-written abstracts.

3.2 Data Generation: AI-generated Dataset

Once the human-written dataset was collected, we developed a dataset of AI-generated abstracts as well, with the explicit goal of getting a balanced dataset. Since different models are trained on diverse datasets with diverse approaches, they also have differences in their text generation. Because of this, we decided to have different AI-generated datasets, one for each model. As the current state-of-the-art models are quite expensive, we opted for using slightly older, mostly non-reasoning models for data generation. Please see Section 2.3 more details on LLM selection.

[10] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[11] https://comserv.cs.ut.ee/ati_thesis/

Since our human-written abstract dataset was limited to theses from the institute of computer science and language or terms from outside the LLM's training dataset can impact the accuracy of its output, we chose to make the AI-generated dataset based on our human-written abstract dataset. To this aim, a Python script was utilized. The script first reads the human-written abstracts. For each abstract, it makes a request to the specified LLM to extract 5-10 keywords from the abstract. Once keywords are extracted, a separate connection is made, requesting the LLM for an abstract to be generated based on the extracted keywords.

Then, the datasets of both human-written and AI-generated abstracts are combined, leaving us with a file for each generating model that consists of 577 human-written and 577 AI-generated abstracts for a total sample size of n=1154, with each item having an id corresponding to the id of the thesis in the registry, the full text of the abstract and a class specifying whether the text is AI-generated or not. Datasets can be split in half to get either only human-written abstracts or only AI-generated abstracts for a specific model.

As each dataset includes abstracts generated for a single model, the generating model's name is kept in the name of the .csv file. This leaves us with 4 English and 4 Estonian datasets, following the naming scheme of "combined_abstracts_language_generating-model.csv".

3.3 Large Language Model Selection

For the purpose of abstract generation, we selected LLMs developed by frontier AI labs. However, we did not exclusively select the most recent and powerful LLMs by these labs. This decision was based on our two hypotheses:

- Users prioritize cost-effective solutions over access to cutting-edge architectures. For example, OpenAI's ChatGPT site typically defaults to using GPT-4o, which was released in May 2024. However, it may switch the underlying model depending on the prompt or the tools being used. This strategy is followed with the intent of keeping costs down, as these slightly older models have significantly lower costs per token compared to the newest ones.
- The LLM-based detection models must utilize a more advanced architecture than the LLMs used for text generation. Our intuition was that, in principle, the detection systems need superior technological capabilities to effectively distinguish synthetic outputs from human-written text.

Taking into account these considerations, the models we selected for generating synthetic academic text (i.e., in our case, synthetic abstracts) were:

- Anthropic's Claude 3 Haiku (estimated at 20B[12])
- Deepseek's R1 (approx 671B parameters, 31B activated during forward pass[13])
- Google's Gemini 2.0 Flash (no estimations)
- OpenAI's GPT-4o-mini. (approx 8B parameters as per (Abacha et al., 2025))

For the classification task, we selected the most recent state-of-the-art models from the same AI labs. Thus, we chose the following models:

- Anthropic's Claude 3.7 Sonnet
- Deepseek's R1
- OpenAI's o4-mini
- Google's Gemini 2.5 Pro Preview (March 25)

[12] https://lifearchitect.substack.com/p/the-memo-special-edition-claude-3 It should be noted that this estimation is not peer-reviewed, and the author's estimation is not explained anywhere.
[13] https://huggingface.co/deepseek-ai/DeepSeek-R1

4. Experiments

Since the goal is to accurately detect AI-generated and human-written abstracts, we chose a variety of prompts and models to test with the goal of eventually determining the most accurate model as well as the most effective prompting technique.

For the experiment, we created a python script that:

1. Starts by taking a dataset, model identifier, prompt template, number of samples, API provider name and optionally temperature, top-k, maximum output tokens and a flag for only running the experiment on human-written abstracts.
2. Then it reads existing results files, getting a list of items that the experiment has been run with the goal of not running the experiment on any abstract twice with the same prompt type.
3. Sends a request to the API provider with the specified prompt for a random unprocessed abstract from the specified dataset.
4. Parses the response.
5. Write the response to the next line in the appropriate results file.
6. Repeats until the specified number of samples is achieved.

The script used multithreading and a token bucket for staying within rate limits.

4.1 Prompt engineering

Inspired by the prompting techniques for the classification task in Vatsal and Dubey's (2024) survey and the prompt engineering techniques defined in OpenAI's best practices for ChatGPT are being used universally, we initially planned to employ CoT, PS, self-consistency and MP as prompting techniques for our classification task. However, due to the high cost, we later did not experiment with PS and self-consistency.

We tested LLMs' capabilities with different textual settings: full text, thought and sentence.

- *Full text* analysis means the entire abstract is analyzed as a whole, helping spot larger patterns that occur throughout the text.
- *Thought*-based analysis asks the LLM to split the abstract into parts by the "thought" that is being communicated. For example, an abstract might have an introduction, an overview of the problem and methodology and a conclusion. Each one of those will be considered as a separate "thought".

- *Sentence*-based analysis asks the LLM to analyze each sentence, looking at a small part of the text at a time.

Each prompt template consists of 4 parts: task overview, input, instructions, and output format. Task overview and output format are kept the same for all prompt templates, but instructions are changed based on the used technique. For each template, we had a two-shot and a zero-shot version.

Description of Prompt Template: Figure 1 depicts a prompt template particularly designed for the CoT-based detection. The structure of the prompt template is organized to guide the LLM in determining whether a given academic text was written by a human or an AI-model. It begins with a *Task Overview,* which outlines the objectives and potential complexity. The *Input* section takes the academic abstract to be evaluated. This is followed by the *Instructions* section, which contains the following: a step-by-step analysis process and the final judgment. The *Analysis framework* subsection helps to analyse the given academic text by a series of analyses – *Linguistic Patterns*, *Logical Flow*, *Contextual Awareness*, *Stylistic Elements*, and *Pattern Analysis*. Finally, the *Output* section presents a structured output format including *reasoning*, *judgment*, *explanation*, and *confidence rating*.

In our thesis, there are a total of 8 prompt templates designed for the detection tasks. Across all prompts, the *Task Overview* and *Analysis* framework are the same. More details can be found in Section 5.3.

```
# AI vs Human Academic Text Detection System - full text

# TASK OVERVIEW
    You are a language analysis expert conducting analysis of different academic texts in {language} to determine whether they were written by a human or an LLM.
    Vigilantly look for signs pointing towards it being written by an ai.
    Consider that these are actual academic texts - being concise and factually correct is a necessity, even for human-written papers.
    Consider that academic texts are impersonal and have no emotion by design.
    Consider that if the text was AI generated, it is a highly advanced model, so exercise constant vigilance.
    Do not output the response as a json.

# INPUT
    {abstract}

# INSTRUCTIONS
    1. Do the following analysis step-by-step:
        I. Read the Text Thoroughly:
            - Identify the main topic and purpose of the text.
            - Note any key arguments, evidence, or conclusions presented.
        II. Assess Language and Style:
                - Does the text exhibit overly polished yet generic phrasing?
            - Are there unnatural or repetitive sentence structures?
            - Identify if the vocabulary is overly simplistic or overly complex.
            - Check for misuse of technical terms or jargon.
        III. Assess Logical Coherence & Flow:
            - Does the argument progress logically with clear transitions?
            - Are there inconsistencies or abrupt shifts in ideas?
        IV. Assess Depth of Analysis & Critical Thinking:
            - Does the text present original insights, or does it rely on surface-level information?
            - Check for a superficial treatment of the topic without critical analysis.
        V.  Compare with Human-Written Text:
            - Think about how human-authored academic texts typically present ideas.
            - Note any deviations from typical human writing patterns.
    2. Evaluation of the given text with justification:
         - AI-generated or human-written

## ANALYSIS FRAMEWORK

### For the entire text, evaluate:

#### LINGUISTIC PATTERNS
- **Lexical diversity**: AI tends to use a more limited vocabulary range with repeated phrases
- **Complexity variation**: Human writing shows more natural variation in sentence complexity
- **Transition quality**: AI often uses generic transitions while humans create more nuanced connections
- **Passive voice frequency**: AI typically overuses passive voice
- **Use of jargon**: AI may use technical terms with less precision or context than human experts
- **Prepositional phrases**: AI often uses longer chains of prepositional phrases
- **Syntactic parallelism**: AI tends to overuse parallel structures

#### LOGICAL FLOW
- **Concept development**: Humans develop ideas organically; AI can make awkward topic shifts
- **Argument coherence**: Human writing shows more consistent logical progression
- **Citation integration**: Humans integrate citations more naturally into their argumentation
- **Counter-argument handling**: Humans address opposing views more deeply and authentically

#### CONTEXTUAL AWARENESS
- **Specificity**: Humans provide more precise details relevant to their field
- **Disciplinary norms**: Humans better adhere to discipline-specific conventions
- **Research gap identification**: Humans identify research gaps more convincingly
- **Theoretical framework usage**: Humans apply theoretical frameworks more appropriately

#### STYLISTIC ELEMENTS
- **Formulaic expressions**: AI overuses standard academic phrases
- **Hedging language**: AI may use excessive or inappropriate hedging
- **Personal voice**: Humans often maintain a more consistent personal voice
- **Metaphor usage**: Humans use more creative and appropriate field-specific metaphors

### Pattern Analysis Across Multiple Sentences:
- **Repetition patterns**: AI tends to reuse phrases and structures
- **Paragraph coherence**: Human paragraphs show more organic development
- **Thematic consistency**: Humans maintain more consistent themes
- **Methodological description**: Humans describe methods with more practical insight
- **Literature review quality**: Humans connect literature more meaningfully to their work
- **Results discussion**: Humans interpret results with more nuance and field awareness

# OUTPUT FORMAT

    1. Step-by-step reasoning:
        - [Insert reasoning for each step here]
    2. Final judgement:
        - [0 if human, 1 if AI-generated **NO OTHER TEXT, JUST THE NUMBER**]
    3. Explanation:
        - [Summarize the key reasons supporting your judgment]
    4. Confidence:
        - [" [your confidence]/10 " WITH THE SPACES]
```

Figure 1. CoT prompt template for full text analysis.

4.2 Evaluation Metrics

For evaluation of results, we compare the accuracies of each model and prompt combination on all datasets. Each run of the experiment was run on the same 50 samples from both human-written and AI-generated datasets.

In our study, the possible results for the classification task are:

- True Positive (TP) refers to the text samples (LLM-generated text) correctly classified by the detection model.
- True Negative (TN) refers to the text samples (human-written) correctly classified by the detection model.
- False Positive (FP) refers to the text samples (human-written) incorrectly classified as AI-generated by the detection model.
- False Negative (FP) refers to the text samples (LLM-written) incorrectly classified as not AI-generated by the detection model.

In our study, we are focusing on the text samples that are correctly classified, i.e., the true positives and true negatives. Hence, considering the balanced distribution of classes, we employed *accuracy* as our evaluation metric, which we can define as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

In cases where we considered results across both human and AI datasets, we found combined accuracy, which we defined as:

$$\text{Accuracy}_{\text{Combined}} = \frac{\text{Accuracy}_{\text{Human}} + \text{Accuracy}_{\text{AI}}}{2}$$

5. Results

This chapter presents the results obtained from our experiments, which include the performance of the four selected LLMs across English and Estonian, the detection approaches with different prompt settings, and the detection complexity of AI-generated text by the pre-selected four LLMs.

5.1 Test Dataset

In Section 3.2, we presented a dataset that has been compiled as part of this thesis work. Although the size of the dataset is 2885 for a single language, experimenting with such a number of samples incurs significant costs. Therefore, we experimented with only a subset of the dataset. The details of the test dataset are given below:

- Subset-1: Estonian-testset
    - AI-generated samples: 200 (50 Estonian samples generated by each selected model)
    - Human-authored: 50 Estonian abstracts
- Subset-2: English-testset
    - AI-generated samples: 200 (50 English samples generated by each selected model)
    - Human-authored: 50 English abstracts

5.2 Model results

For the research questions R1, R2 and R3, we investigated the detection performance of 4 LLMs across three language settings with different prompting approaches. We experimented with these three language settings: Estonian, English and both. The details of the prompting settings are discussed in Section 3.2. Table 1 shows the obtained results along with the best prompt for each employed model. For Table 1, we obtained the results with the entire testset; however, Table 2 is based on the human-authored testset whereas Table 3 is based only on the AI-generated testset.

Model Performance Summary (Combined Data)

Language	Classification Model	Avg Combined Acc	Best Combined Acc	Best Prompt (Combined Acc)
ET	claude-3.7-sonnet	0.52	0.78	metacognitive-0-shot
ET	deepseek-r1	0.74	0.81	thought-0-shot
ET	gemini-2.5-pro-preview-03-25	0.52	0.61	thought-0-shot
ET	o4-mini	0.75	**0.82**	**metacognitive-0-shot, thought-0-shot**
EN	claude-3.7-sonnet	0.76	0.87	full-text-0-shot
EN	deepseek-r1	0.77	0.81	thought
EN	gemini-2.5-pro-preview-03-25	0.70	0.82	thought-0-shot
EN	o4-mini	0.86	**0.89**	**full-text**
Combined	claude-3.7-sonnet	0.64	0.82	metacognitive-0-shot
Combined	deepseek-r1	0.76	0.81	thought-0-shot
Combined	gemini-2.5-pro-preview-03-25	0.61	0.71	thought-0-shot
Combined	o4-mini	0.80	**0.85**	**metacognitive-0-shot**

Table 1. Overview of LLMs' accuracies across all datasets. Best results per language for accuracy are in bold.

Similarly, Table 2 presents an overview of the results for the human-written dataset

Model Performance Summary (Human Data)

Language	Classification Model	Avg Acc (Human)	Best Acc (Human)	Best Prompt (Human Acc)
ET	claude-3.7-sonnet	0.95	**1.00**	**full-text, full-text-0-shot, sentence, sentence-0-shot**
ET	deepseek-r1	0.60	0.80	metacognitive
ET	gemini-2.5-pro-preview-03-25	0.98	**1.00**	**full-text, metacognitive, metacognitive-0-shot**
ET	o4-mini	0.79	0.92	sentence
EN	claude-3.7-sonnet	0.81	**1.00**	**full-text, thought**
EN	deepseek-r1	0.33	0.54	metacognitive
EN	gemini-2.5-pro-preview-03-25	0.96	**1.00**	**full-text, full-text-0-shot, sentence, sentence-0-shot**
EN	o4-mini	0.45	0.54	full-text
Combined	claude-3.7-sonnet	0.88	**1.00**	**full-text, full-text-0-shot, sentence, sentence-0-shot, thought**
Combined	deepseek-r1	0.47	0.80	metacognitive
Combined	gemini-2.5-pro-preview-03-25	0.97	**1.00**	**full-text, full-text-0-shot, metacognitive, metacognitive-0-shot, sentence, sentence-0-shot**
Combined	o4-mini	0.62	0.92	sentence

Table 2. Overview of LLMs' accuracies across all human-written datasets. Best results per language for accuracy are in bold.

Table 3 gives an overview of the results for AI-generated datasets.

Model Performance Summary (AI Data)

Language	Classification Model	Avg Acc (AI)	Best Acc (AI)	Best Prompt (AI Acc)
ET	claude-3.7-sonnet	0.41	0.77	metacognitive-0-shot
ET	deepseek-r1	0.77	**0.99**	**thought-0-shot**
ET	gemini-2.5-pro-preview-03-25	0.41	0.52	thought-0-shot
ET	o4-mini	0.74	0.87	thought-0-shot
EN	claude-3.7-sonnet	0.75	0.98	metacognitive
EN	deepseek-r1	0.89	**0.99**	**thought-0-shot**
EN	gemini-2.5-pro-preview-03-25	0.63	0.84	thought-0-shot
EN	o4-mini	0.96	0.99	thought-0-shot
Combined	claude-3.7-sonnet	0.58	0.87	metacognitive-0-shot
Combined	deepseek-r1	0.83	**0.99**	**thought-0-shot**
Combined	gemini-2.5-pro-preview-03-25	0.52	0.68	thought-0-shot
Combined	o4-mini	0.85	0.93	thought-0-shot

Table 3. Overview of LLMs' accuracies across all AI-generated datasets. Best results per language for accuracy are in bold.

For human abstracts, Gemini 2.5 Pro Preview (March 25, 2025) and Claude 3.7 Sonnet had the best results. When making predictions on AI-generated datasets, Deepseek's R1 had the highest results. When considering all datasets, o4-mini had the highest accuracy.

5.3 Prompt Results

For the research question R3, we investigated the impact of detection capability in different prompt settings. Table 4 shows average accuracy for each classifying model using the prompt across the entire test dataset. We experimented with the following prompt settings:

- *full-text* and *full-text-0-shot*: instructs the model to consider the text as a whole, looking for patterns that could indicate it being AI-generated across the whole text.
- *thought* and *thought-0-shot:* Instructs the model to first split the text into separate thoughts, then consider each one separately when looking for signs of AI text.
- *sentence* and *sentence-0-shot:* Instructs the model to split the text into sentences, then consider each sentence separately.
- *metacognitive* and *metacognitive-0-shot:* Instructs the model to read through the text, then give a preliminary analysis and classification. Then, critically evaluate the analysis and come to a final conclusion.

Prompt Performance by Model (Accuracy, COMBINED)

prompt	claude-3.7-sonnet_accuracy	deepseek-r1_accuracy	gemini-2.5-pro-preview-03-25_accuracy	o4-mini_accuracy
full-text	0.69	0.75	0.59	0.81
full-text-0-shot	0.70	0.77	0.57	0.79
metacognitive	0.80	0.74	0.68	0.84
metacognitive-0-shot	**0.82**	0.72	0.65	**0.85**
sentence	0.45	0.72	0.49	0.70
sentence-0-shot	0.43	0.74	0.57	0.81
thought	0.56	0.81	0.61	0.78
thought-0-shot	0.67	**0.81**	**0.71**	0.84

Table 4. Shows accuracy for each model-prompt combo across all datasets. The highest accuracy for each model is in bold.

Table 5 shows accuracy by prompt for each dataset source. OBJ

Prompt Performance by Source (Accuracy, COMBINED)

prompt	claude-3-haiku_accuracy	deepseek-r1_accuracy	gemini-2.0-flash_accuracy	gpt-4o-mini_accuracy	human_accuracy
full-text	0.79	0.42	0.73	0.83	0.78
full-text-0-shot	0.80	0.45	0.72	0.81	0.76
metacognitive	0.88	0.55	**0.80**	0.89	0.70
metacognitive-0-shot	0.89	0.53	0.79	**0.90**	0.70
sentence	0.70	0.27	0.51	0.69	**0.80**
sentence-0-shot	0.73	0.37	0.52	0.78	**0.80**
thought	0.80	0.39	0.66	0.85	0.76
thought-0-shot	**0.90**	**0.66**	0.75	0.90	0.57

Table 5. Shows average accuracy of each prompt for each dataset source. the biggest positive difference from the average of all other prompts is in bold.

Metacognitive prompting without examples performed best for Claude 3.7 Sonnet and o4-mini, while zero-shot CoT thought-based prompting performed best on Deepseek's R1 and Gemini 2.5 Pro.

Zero-shot thought-based CoT prompting worked best on abstracts generated by Claude 3 Haiku and Deepseek's R1, while Metacognitive prompting worked best on abstracts generated by Gemini 2.0 Flash and GPT 4o-mini. For human-written abstracts, zero-shot sentence-based CoT prompting yielded the best results.

5.4 Results by Abstract Source

Model Performance by Source (Accuracy, COMBINED)

model	claude-3-haiku_accuracy	deepseek-r1_accuracy	gemini-2.0-flash_accuracy	gpt-4o-mini_accuracy	human_accuracy	average
claude-3.7-sonnet	0.67	0.37	0.55	0.72	0.88	0.64
deepseek-r1	0.90	0.55	**0.94**	**0.92**	0.47	0.76
gemini-2.5-pro-preview-03-25	0.75	0.21	0.35	0.77	**0.97**	0.61
o4-mini	**0.92**	**0.69**	0.89	0.90	0.62	0.80
OVERALL AVERAGE	**0.81**	**0.46**	**0.68**	**0.83**	**0.74**	**0.70**

Table 6. Average accuracy of models for each data source, combined over results for English and Estonian datasets.

We also investigated the detection complexity of AI-generated texts, which were generated by the pre-selected 4 LLMs. From Table 6, it can be seen which AI-generated abstracts are difficult to detect. Abstracts generated by GPT 4o-mini have the highest rate of detection, followed by Claude 3 Haiku, which is detected more often than Gemini 2.0 Flash. Deepseek R1 is the least detected data source.

5.5 Language-based Results

For research question R1 (see Section 1.), we investigated the language-based performance of our selected LLMs. Figures 2 and 3 depict the difference in performance on English and Estonian data for each model and prompt, respectively.

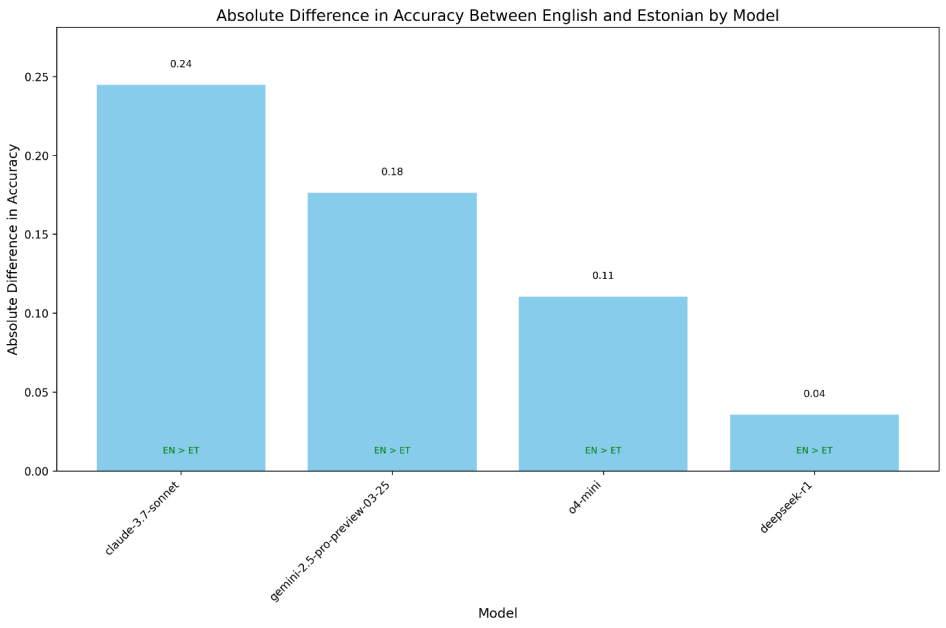


Figure 2. Bar chart showing the absolute difference in accuracy between classifications in English and Eston for each model. Lower is better.

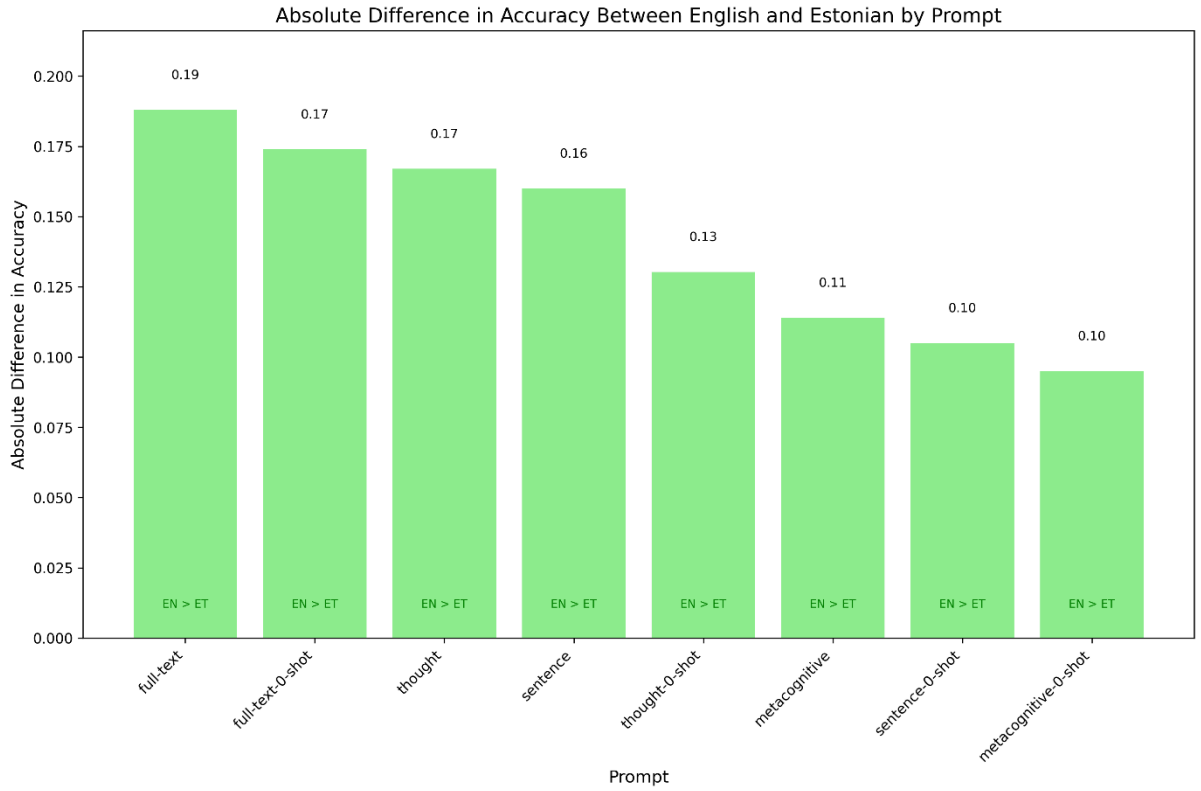


Figure 3. Bar chart showing the absolute difference in accuracy between classifications in English and Estonian for each prompt. Lower is better.

Deepseek's R1 had the smallest difference across languages. For prompts, metacognitive-0-shot had the smallest difference.

6. Discussion

This chapter discusses the findings of the thesis, offering reasons for why the results came out as they did, bringing out shortcomings of the thesis and offering ideas on further work based on this thesis.

6.1 Detection based on LLMs

Our findings showed that when looking at detection across human and AI datasets, OpenAI's o4-mini obtained the highest accuracy with zero-shot metacognitive prompting and Deepseek R1 correctly detected 99% of AI-written abstracts when prompted with the zero-shot thought prompt. However, Gemini 2.5 Pro and Claude 3.7 Sonnet outperformed both when looking at human-written abstracts, both achieving 100% accuracy on the full text prompt. However, this doesn't necessarily mean that it's a good model for detecting AI-generated texts, as any false positives in an academic context could lead to huge consequences for innocent people. Thus, the best option for practical uses would be a solution that provides no false positives and the highest amount of true positives. When looking for the model-prompt combination that has both a human detection accuracy over 95% and the highest AI detection rate, we get Anthropic's Claude 3.7 Sonnet, which achieves a 97% accuracy on detecting human-written texts and a 62.75% accuracy on AI-generated texts with the zero-shot full text prompt and accuracies of 100% on human-written text and 61.25% on AI-generated texts with the 2-shot full text prompt. If our goal was to pick a single model and prompt combination to utilize in a tool for an AI-generated text detection tool across all languages, we would use that. Limiting the results to English only, we have an accuracy of 100% on human-written texts and 81.5% on AI-generated texts with the 2-shot full text prompt.

It should be noted that the abstract generating LLMs were mostly (except for Deepseek R1) not state-of-the-art. Because of this, we can't extrapolate the average results to all models. When only looking at classifications on abstracts generated by R1, accuracy is often more than halved.

6.2 Impact of Language on Detection

Looking at detection accuracy based on language, we observed that Deepseek's R1 obtained the smallest difference between detection accuracy on the English and Estonian datasets. A possible reason for this outcome is that the model was likely trained on more multilingual datasets than the other models. As Deepseek used Deepseek V3 as the base for R1

(DeepSeek-AI, Guo, et al., 2025), and the technical report for V3 specifically notes having more multilingual coverage than V2 (DeepSeek-AI, Liu, et al., 2025), the documentation also somewhat backs up this potential reason, though we lack the information to compare the training corpus of V3 to other AI labs' corpuses.

Overall, classification accuracy was higher on English datasets compared to Estonian abstracts.

6.3 Impact of Model Size on Detection

Assuming that 4o-mini is the smallest model and Deepseek R1 the biggest, and Claude 3 Haiku intuitively seeming smaller than Gemini 2.0 Flash, we expected the detection accuracy to follow the estimated parameter count, with the smallest model being detected most easily. This expectation was proven true.

However, the size of the model did not play a linear role in its detection accuracy. OpenAI's o4-mini was consistently one of the better models, even though it is a distilled version of a larger o4 model.

6.4 Impact of Prompting on Detection

Per Table 4, it can be argued that some prompts outperform others consistently. For example, sentence and zero-shot sentence prompts performed considerably worse than other prompts when used with Gemini 2.5 Pro and Claude 3.7 Sonnet. However, Deepseek-R1 and o4-mini had quite consistent results across all prompts, though metacognitive prompting had the overall best results.

Though recommended by OpenAI's prompting guide, including examples in the prompt did not lead to consistently better performance.

6.5 Limitations

Detecting AI-generated academic formal content based on just an abstract is likely one of the hardest tasks due to abstracts' nature: they are – by design – impersonal, concise, and avoid getting into much detail. Thus, the results found in this paper are likely not exactly interpolable to other kinds of academic texts.

Although an abstract (in our case, the sample of an academic textual content) of a thesis is only a summary of an entire thesis, we restricted our scope in this work to the abstract of a

thesis. Investigating the entire thesis would cost a significant budget, which is not very feasible for a bachelor's thesis.

Taking into account the high token cost for LLMs, we conducted the detection experiments only on 50 samples of each sub-dataset and reported the results based on that subset of the dataset. Perhaps, we might have missed some more insights due to our experimental settings.

Additionally, as we focused on binary classification and the dataset was also built around that, we do not have information on how the classification accuracy would be impacted by texts that were initially written by humans, then rephrased or otherwise changed by LLMs. The instructions for generating abstracts did not include instructions to sound as human as possible either, meaning that the models weren't tested in regard to actively trying to avoid detection, which is what one would do when committing academic fraud.

Also, as AI is still a very rapidly developing field, these results are only a snapshot of a point in time. For true AI-generated text detection, major changes should be made by AI labs.

6.6 Further work

From this work's findings and limitations, we have identified the following topics to expand on this thesis:

1. Evaluating this technique for other types of academic texts, e.g. full papers or essays.
2. Fine-tuning an open-source model for detecting AI-generated texts.
3. Expanding the dataset by including more languages, more fields or other types of academic texts.
4. Using newer, state-of-the-art models for generating abstracts and subsequently evaluating them.

7. Conclusion

The aim of this thesis was to investigate the capability of current state-of-the-art LLMs' capability to detect AI-generated text, in particular, LLM-generated text, as well as the impact of language resources, model size and prompting techniques on that capability, particularly focusing on detecting AI-generated abstracts in English and Estonian. Our work revealed multiple insights. We observed the following:

1. **RQ1. Language resource impact:** Language resource availability significantly influences detection accuracy, with most models performing considerably better on high-resource English texts compared to low-resource Estonian, although Deepseek R1's performance was only minimally affected by the language.
2. **RQ2. Model Size vs. Detectability Trade-off:** The scale of the LLM used for generating abstracts played a considerable role in the detectability of the text generated by it. Texts by the smallest model, GPT 4o-mini, had the highest rates of detection, while abstracts generated by Deepseek R1 went undetected more often than not.
3. **RQ3. Prompting technique impact:** The prompting technique used for classifying LLM played a considerable role in the detection accuracy. Specifically, metacognitive and thought-based CoT prompting displayed the strongest results with specific models, though no single prompt proved to be clearly better than the rest.

The contributions of this research came to be threefold:

1. We presented a novel bilingual benchmark dataset for the research of AI-generated academic textual content detection. This is the first and only available dataset in Estonian for the AI-generated content detection task. The size of this dataset is 577 human samples and 2188 of AI-generated samples for both English and Estonian. The dataset is publicly shareable and does not require compliance with GDPR.
2. An evaluation of current state-of-the-art LLMs (Anthropic's Claude 3.7 Sonnet, OpenAI's o4-mini, Google's Gemini 2.5 Pro (March 25 preview) and Deepseek R1) in the context of detecting AI-generated academic abstracts. The evaluation establishes the performance of these models to give an overview of potential use by the academic community.

3. A systematic analysis of key factors impacting detection performance. These factors are language resource availability, the scale of the LLM that generated the abstract and the prompting technique used for detection.

The challenge of reliably distinguishing human-written academic texts from AI-generated texts is complex and subject to continuous evolution, as AI capabilities will continue to improve. The findings suggest that although the current models can contribute to the process of evaluating the authorship of texts, their effectiveness is influenced by a variety of factors – ranging from the language to the model that generates the evaluated text. Due to the importance of differentiating between AI-generated and human-written texts, the task of developing a reliable mechanism for AI text detection remains essential. Continued research into AI detection models for both general and academic domains is crucial for safeguarding integrity in both general online discourse and academic environments, as artificial intelligence will only be utilized more and more.

References

Abacha, A. B., Yim, W., Fu, Y., Sun, Z., Yetisgen, M., Xia, F., & Lin, T. (2025). *MEDEC: A Benchmark for Medical Error Detection and Correction in Clinical Notes*. https://arxiv.org/abs/2412.19260

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., … Zhang, Z. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. https://arxiv.org/abs/2501.12948

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., … Pan, Z. (2025). *DeepSeek-V3 Technical Report*. https://arxiv.org/abs/2412.19437

Gehrmann, S., Strobelt, H., & Rush, A. (2019). GLTR: Statistical Detection and Visualization of Generated Text. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 111–116. https://doi.org/10.18653/v1/P19-3019

Hu, X., Chen, P.-Y., & Ho, T.-Y. (2023). *RADAR: Robust AI-Text Detection via Adversarial Learning*. https://arxiv.org/abs/2307.03838

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2024). *A Watermark for Large Language Models* (No. arXiv:2301.10226). arXiv. https://doi.org/10.48550/arXiv.2301.10226

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). *DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature* (No. arXiv:2301.11305). arXiv. https://doi.org/10.48550/arXiv.2301.11305

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Vatsal, S., & Dubey, H. (2024). *A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks*. https://arxiv.org/abs/2407.12994

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2023). *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. https://arxiv.org/abs/2203.11171

Wang, Y., & Zhao, Y. (2024). *Metacognitive Prompting Improves Understanding in Large Language Models*. https://arxiv.org/abs/2308.05342

Wu, J., Yang, S., Zhan, R., Yuan, Y., Chao, L. S., & Wong, D. F. (2025). A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions. *Computational Linguistics*, *51*(1), 275–338. https://doi.org/10.1162/coli_a_00549

Appendices

License

I, Martin Kõnnussaar ,

(author´s name)

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

reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis

Exploring the Capability of Large Language Models to Detect AI-generated Academic Texts

(title of thesis)

supervised by Somnath Banerjee, PhD ;

(supervisor´s name)

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the 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. am aware of the fact that the author retains the rights specified in points 1 and 2;

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

Martin Kõnnussaar

15/05/2025