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

Eduard Rudi

Extracting Lexical Relations from Large Pre-trained Language Models

Master's Thesis (30 ECTS)

Supervisor(s): Mark Fišel
Heili Orav

Tartu 2024

Generating Lexical Relations with Large Pre-trained Language Models

Abstract:
With the rise of OpenAI's ChatGPT, large language models have become extremely popular. ChatGPT is powered by a generative pre-trained transformer (GPT), with the latest version being GPT-4. This thesis aims to test the capabilities of GPT-4, which is currently considered state-of-the-art. The approach is to generate a Wordnet, a collection of words and their relationships. This is an effective method of testing GPT-4 as it allows for testing in multiple languages, including resource-rich languages like English and resource-poor languages like Estonian. Previous attempts at generating Wordnet relied heavily on machine translation, which is typically ineffective for resource-poor languages. Unfortunately, GPT-4 did not perform as well as expected, struggling to generate all meanings and over-generating relationships in both languages. Ultimately, generative LLMs work best when context already exists, such as in summarization or generating unit tests.

Keywords: large language models, LLM, Wordnet, GPT-4, generating Wordnet

CERCS: P176 Artificial Intelligence

Leksikaalsete suhete genereerimine suurte eeltreenitud keelemudelitega

Lühikokkuvõte:
Alates OpenAI ChatGPT ilmumisest on suured keelemudelid muutunud äärmiselt populaarseks. ChatGPT põhineb generatiivsel eeltreenitud transformeril (GPT), mille uusim versioon on GPT-4. Selle lõputöö eesmärk on testida GPT-4 võimeid, mida praegu peetakse suurte keelemudelite tipptasemeks. Lähenemisviisiks on *wordnet*-tüüpi sõnastiku genereerimine, mis on sõnade ja nende suhete võrgustik. See on tõhus meetod GPT-4 testimiseks, sest võimaldab testida mudelit mitmes keeles, sealhulgas ressursirohketes keeltes nagu inglise keel ja ressursivaestes keeltes nagu eesti keel. Varasemad katsed *wordnet*i genereerimisel tuginesid suuresti masintõlkele, mis tavaliselt ei ole ressursivaeste keelte puhul tõhus. Kahjuks ei osutunud GPT-4 sooritus siinses töös nii heaks, kui oodati. Enim esines raskusi sõna kõigi tähenduste genereerimise ja suhete üle genereerimisega. Need probleemid esinesid mõlemas keeles. Lõppkokkuvõttes töötavad generatiivsed suured keele mudelid kõige paremini, kui kontekst juba eksisteerib, näiteks kokkuvõtete loomisel või ühiktestide genereerimisel.

Võtmesõnad: suured keelemudelid, wordnet, GPT-4, sõnastiku genereerimine

CERCS: P176 Tehisintellekt

Contents

1 Introduction

Language resources play a vital role in language technologies, as they are utilized to train and develop artificial intelligence models. However, creating these resources can be costly due to the expenses associated with hiring human annotators or lexicographers. As a result, the efficient generation of Wordnet has long been an area of interest for researchers and scientists. There have been numerous approaches to generating Wordnet for a specific language, with many relying on machine translation for automation. However, the process of generating Wordnet presents challenges due to the varying relationships between words in different languages and the existence of words in one language that may not have close equivalents in another.

With the introduction of ChatGPT[1], generative artificial intelligence (AI) has become incredibly popular. Initially, people were amazed at the flexibility of AI and its ability to solve problems that it did not learn, but as more people experimented with it, they began to realize its limitations. Therefore, this thesis aims to evaluate GPT-4's ability to generate Wordnets.

GPT-4 was selected over other options due to its cutting-edge technology and superior performance when dealing with resource-limited languages like Estonian [BCE+23, Ope24]. Specifically, GPT-4 Turbo[2] version, identified as *gpt-4-1106-preview*, was chosen for its speed and cost-effectiveness, providing consistent results at a lower expense. If the results are promising, this AI technology could potentially save time and money in Wordnet generation, as well as other projects and language resources. To put things into perspective, generating Wordnet for just five words using GPT-4 Turbo typically costs around 50 cents.

This thesis aims to explore the capabilities of GPT-4 for generating Wordnet in both resource-rich and resource-poor languages, specifically English and Estonian and ultimately answer these questions:

1. Can GPT-4 generate Wordnet for English, which is a resource-rich language, and how good is it?

2. Can GPT-4 generate Wordnet for Estonian, which is a resource-poor language, and how good is it compared to English?

3. What are the strengths and weaknesses and where are the main problems with the current generation of generative large language models for generating lexical relations?

4. How does the prompt affect the output of the results?

[1]ChatGPT. https://openai.com/blog/chatgpt. Accessed: 2024-04-07

[2]GPT-4 Turbo. https://help.openai.com/en/articles/8555510-gpt-4-turbo-in-the-openai-api. Accessed: 2024-04-29

The thesis is divided into two parts: a code component and a written component. The code is publicly available on the author's GitHub repository[3]. The repository has the code and generated Wordnets. The code itself includes the prompts, utilization of the OpenAI library to get the output from GPT-4, Wordnet construction into an XML file and automatic evaluation.

The thesis is going to begin with a background section providing an overview of Wordnet's history, structure and relations, as well as large language models (LLMs), GPT-4 and related work. The following chapter will detail the methodology employed in generating Wordnet, including the steps taken and the reasoning behind them. This chapter is also going to present experimental results and examples. The discussion chapter is going to examine the results and offer the author's opinion, along with suggestions for future developments. Finally, the conclusion chapter is going to provide a brief summary of the thesis.[4]

[3]Access the code repository at https://github.com/EduardNot/Wordnet-with-GPT4

[4]The author of this thesis used Grammarly's AI-powered generative feature to improve readability and grammar.

2 Background

This section aims to provide a comprehensive overview of WordNet [Uni10], its origins, structure and relations, as well as large language models (LLMs), with a focus on the intricate details of GPT-4. Finally, an overview of related work will be provided.

2.1 WordNet

In the 1980s, George Miller and his team at Princeton University created WordNet [Uni10], a database comprising lexical conceptual relations [Vos02]. Similar to a thesaurus, Wordnet's synsets group together words or phrases that share the same meaning and part of speech. Some synsets consist of only one word since there is no other way to convey the concept. Links connect the synsets based on relations between concepts. Essentially, a thesaurus like Wordnet helps find words with similar meanings.

English WordNet gained immense popularity, leading to the launch of EuroWordNet[5], a European project aimed at creating multilingual databases for WordNet in 1996. Initially targeting Dutch, Spanish and Italian languages, it expanded in 1997 to include German, French, Czech and Estonian. The primary objective of EuroWordNet was to link these Wordnets to the English database [Vos02]. Due to the EuroWordNet project, the development of Estonian Wordnet (EstWN) started, which uses the EuroWordNet relations and was developed and funded until June 2022 [est]. As of April 2024, EstWN has 92438 synsets and 343025 relations between them, according to [est].

Following the conclusion of the EuroWordNet project, individuals continued to maintain their language Wordnets and even created new ones using the EuroWordNet template [Vos02]. As a result, many people sought assistance and guidance. In response, Princeton WordNet and EuroWordNet coordinators formed a non-profit organization known as the Global WordNet Association (GWA)[6], which remains active to this day [Vos02].

As evident in the thesis, both *WordNet* [Uni10] and *Wordnet* are referenced. It is worth noting that *WordNet* is a trademark owned by Princeton University. However, it is available for research and commercial use without charge as long as the licensing agreement is adhered to. On the other hand, *Wordnet* refers to a Wordnet-like dictionary, which is concept-based and has relationships between concepts. Wordnet is also called linguistic ontology. Essentially, both terms describe the same notion, with one being the trademarked product and the other representing the concept for this type of dictionary or ontology.

[5]EuroWordNet. https://archive.illc.uva.nl//EuroWordNet/. Accessed 2024-04-07

[6]Global WordNet. http://globalwordnet.org/. Accessed: 2024-04-07

2.1.1 Relations

The foundation of the Wordnet lies in the synonym or synset. Each synset is connected to another synset and the Princeton WordNet [Uni10] included fourteen relations for nouns and verbs. The EuroWordNet had a more difficult task. Since EuroWordNet deals with multiple languages, linking a synset between languages using the Princeton WordNet relations did not work because each language has its own specifics. Thus, they had to add more relations, which made linguistic ontologies rather than ontologies for making inferences only. EuroWordNet and Global WordNet added more than double the original relations. The Princeton and non-Princeton WordNet relations can be found on the Global WordNet GitHub page[7]. With each different language, Wordnet has its relations.

The relations chosen by the author to generate Wordnets are the following:

- Synonyms: words with similar meanings, such as *male*, *dude* and *guy*.
- Hyponyms: words that represent a subtype of another word, like *cat* being a hyponym of *animal*.
- Hypernyms: words that represent a supertype of another word, such as *animal* being a hypernym of *dog*.
- Meronyms: words that refer to a part of a whole, such as *hands* in the phrase *I need a pair of hands* referring to people.
- Holonyms: words that refer to the whole that a part belongs to, as in *people* being the holonym of *hands* in the phrase *I need a pair of hands*.
- Antonyms: words with opposite meanings, like *girl* being the antonym of *boy*.

These relations were chosen because they are the most important and form the basis of all other relations.

2.1.2 Structure

The GWA format documentation[7] specifies that the XML must always start with Figure 1 code and the following information is required.

- Id: a short name for the resource
- Label: the full name for the resources
- Language: BCP-47 format language code

[7]Global WordNet Formats. https://globalwordnet.github.io/schemas/. Accessed: 2024-04-07

- Email: contact email address
- License: the license of the resource
- Version: a string identifying the version
- URL: a URL for the project homepage
- Citation: the paper to cite for the resource
- Logo: a link to a logo for the project.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LexicalResource SYSTEM "http://globalwordnet.github.io/
    schemas/WN-LMF-1.3.dtd">
<LexicalResource xmlns:dc="http://purl.org/dc/elements/1.1/">
```

Figure 1. XML file mandatory start.

The structure of Wordnet can be divided into two parts: lexical entries, which are word definitions, and relations, which include all word relations. Figure 2 shows an example from EstWN. The GWA format documentation also specifies that the Wordnet can be provided in JSON-LD or RDF formats, both of which follow the same structure as XML. Following the explanation of Wordnet, the next important aspect to be explained in the thesis is the large language model.

2.2 Large language models (LLMs)

The concept of the large language model has already been known for quite some time, but the current architecture is based on transformers. The groundbreaking paper *Attention Is All You Need* introduced this new architecture proposed by Vaswani et al. [VSP+17]. Their method utilizes self-attention to establish global dependencies between input and output, resulting in improved translation quality and efficiency. This approach replaced the recurrent layers in the recurrent neural network, resulting in a more parallelizable and faster training process. As a result of these advancements, state-of-the-art performance and results have been achieved for English-to-German and English-to-French translation tasks.

Initially, the general public had little interest in transformers and they were mainly used in the machine translation domain. Only after the release of Bidirectional Encoder Representations from Transformers (BERT) [DCLT18] and generative pre-trained transformer (GPT) [RNSS18] people started to make language models out of transformers. The largest attention that the transformer architecture got was not until OpenAI

```
<LexicalResource>
    <Lexicon>
    <LexicalEntry id="w438433">
        <Lemma partOfSpeech="n" writtenForm="aastaeelarve" />
        <Sense id="s-aastaeelarve-n1" status="unchecked" synset="
            estwn-et-28013-n">
            <Example language="et">Tallinna eelmise aasta eelarve
                tulemused olid head.</Example>
        </Sense>
    </LexicalEntry>
    ...
    <Synset confidenceScore="0.5" dc:type="C" id="estwn-et-14243-n"
        ili="" status="unchecked">
        <Definition dc:source="EKILEX" language="et" note="
            @is_primary">teadus, mis uurib informatsiooni salastamise
            viise</Definition>
        <SynsetRelation confidenceScore="1.0" relType="hypernym"
            status="unchecked" target="estwn-et-419-n" />
        <SynsetRelation confidenceScore="1.0" relType="hyponym"
            status="unchecked" target="estwn-et-44474-n" />
        <SynsetRelation confidenceScore="1.0" relType="hyponym"
            status="unchecked" target="estwn-et-45523-n" />
        <SynsetRelation confidenceScore="1.0" relType="similar"
            status="unchecked" target="estwn-et-26461-n" />
    </Synset>
    ...
    </Lexicon>
</LexicalResource>
```

Figure 2. XML file mandatory start.

introduced ChatGPT[8], a sibling of InstructGPT [OWJ+22], that LLMs and transformers gained widespread popularity and hype.

InstructGPT was a chatbot that was useful for specific domains, while ChatGPT was a more general chatbot. Although ChatGPT initially used OpenAI's GPT-3.5[9], it can now also utilize GPT-4 [BCE+23, Ope24]. When the general public got their hands on ChatGPT, they were blown away by the possibilities of artificial intelligence. The model works by receiving prompts from the user and then predicting the sequence of words to form sentences based on its learning data. ChatGPT was also a unique experience for many users who had to unlearn Googling patterns and instead write detailed instructed prompts.

[8]ChatGPT. https://openai.com/blog/chatgpt. Accessed: 2024-04-07

[9]GPT-3.5. https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates. Accessed: 2024-04-07

2.2.1 Transformer structure

According to Vaswani et al. [VSP+17], the transformer structure consists of two main layers: an encoder and a decoder, which can be seen in Figure 3. The encoder is responsible for converting the input sequence into a continuous numerical sequence, which is then passed on to the decoder layer. The encoder layer comprises a series of identical layers, each containing two sublayers - a multi-head attention layer and a fully connected feed-forward network. Meanwhile, the decoder is tasked with generating the output sequence based on the sequence obtained from the encoder, as well as the decoder output from the previous step. Like the encoder, the decoder is composed of a stack of identical layers, but it also includes a masked multi-head attention layer in addition to the two sublayers.

While the encoder and decoder are crucial layers, it is important to also explore their sublayers. One such sublayer is multi-head attention, which is essentially multiple self-attentions. Each *head* operates on different input projections, enabling the model to focus on various features in the input. The decoder block includes an extra multi-head attention feature, which is masked to prevent cheating. Although the decoder generates tokens sequentially, the entire input is used for parallel and efficient training. Without the mask, the decoder would have access to future information, but the mask limits the attention scores to only previous and current word scores.

Another sublayer is the feed-forward network, which consists of weights that are updated during training using backpropagation. This network takes the input of self-attention and applies the same transformation to each word's contextualized embedding. It then communicates with the output layer to feed the next layer normalization. The data in the feed-forward network flows only forward, as the name suggests and in the transformer, the neurons are all connected to the next layer's neurons.

Lastly, it is important to discuss the concept of positional encoding. When dealing with models that lack recurrence or convolution, like transformers, it becomes necessary to incorporate information regarding the relative or absolute position of the tokens. This is achieved by adding positional encoding to the input for the embeddings, allowing for a model of order information. In their work, Vaswani et al. [VSP+17] utilized sine and cosine functions with varying frequencies to represent relative positions, with each dimension possessing its sinusoid.

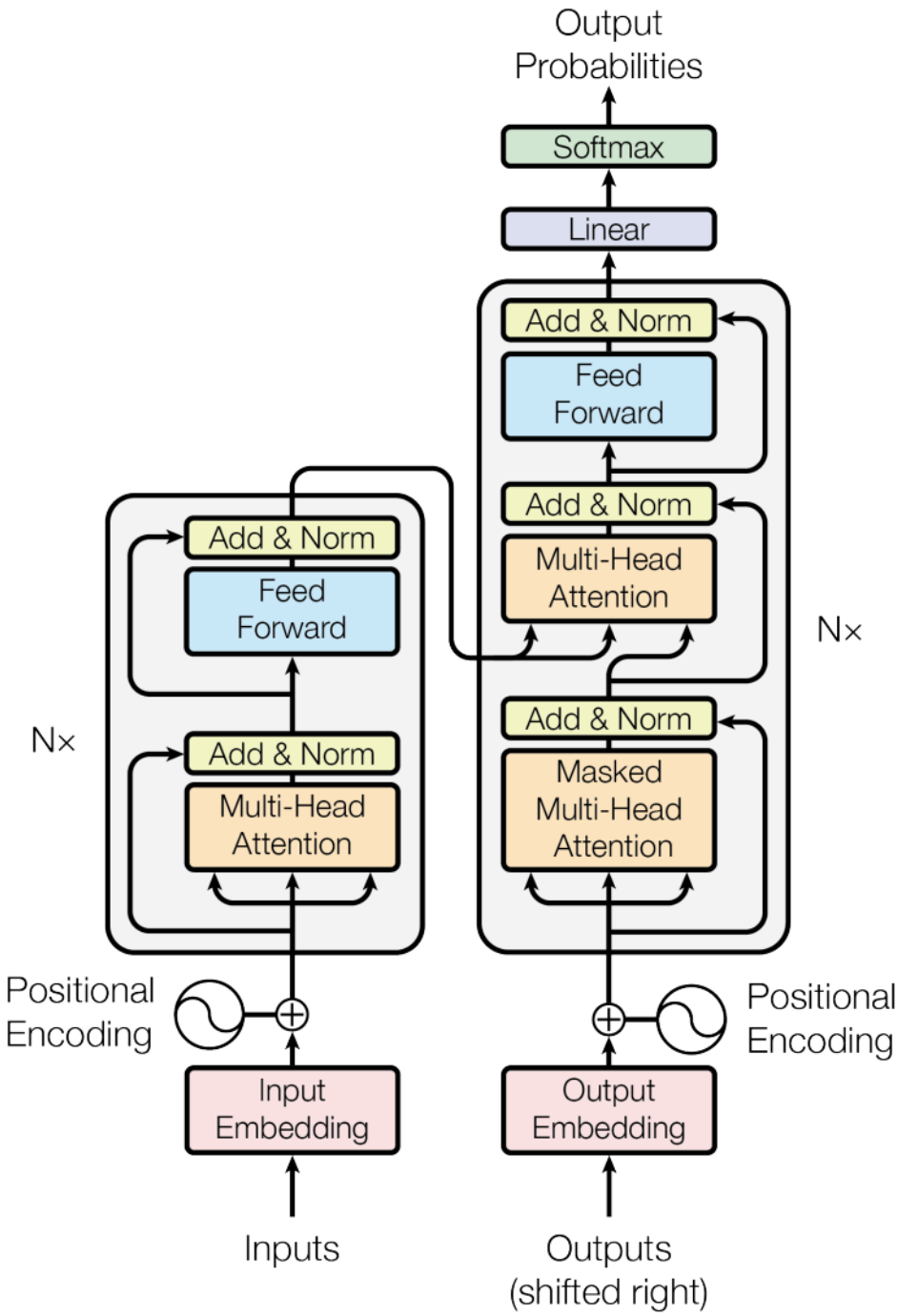

Figure 3. Transformer architecture [VSP+17]

2.2.2 GPT architecture and GPT-4

Shifting focus to GPT architecture, one can notice that it deviates slightly from the transformer architecture. Specifically, GPT architecture exclusively uses the decoder component of the transformer architecture, with modifications that exclude the second multi-head attention and normalization block, as illustrated in Figure 4. These decoders are stacked to make the model more complex and capable. While the decoder block continues to function the same, there are distinctions in the model's training and how it generates output.

Unfortunately, the release of GPT-4 did not come with a research paper from OpenAI but rather a blog post. However, based on what is known, the GPT-4 model has the same structure and architecture as GPT-3. According to Brown et al. [BMR+20], the GPT-3 and presumably the GPT-4 training process involves a large amount of text data and the objective is to train the model to predict the next word. One way this is achieved is by masking a word in a sentence, which can be done for different words. This generates multiple examples, each with a different masked word that the model has to predict. The model is updated based on the numerical error value calculated from comparing the predicted and actual words. Additionally, the model is fine-tuned with specific instructions to guide its behavior.

The GPT models work in generating text in a similar manner to the training process, as their aim is to predict the next word. However, the process is autoregressive, implying that the GPT model utilizes the current prompt to predict one word, then adds that word to the prompt and predicts the next one until the entire output is generated.

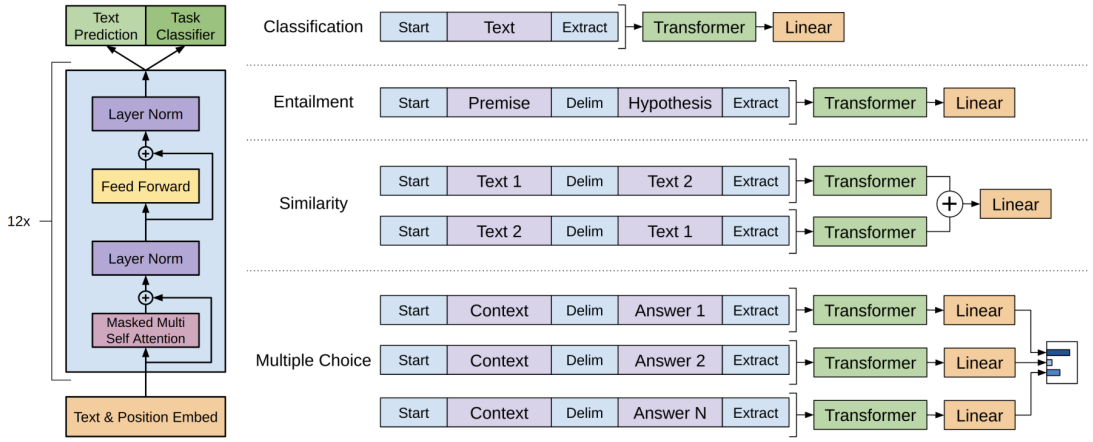

Figure 4. GPT architecture [RNSS18]

The latest model from OpenAI, GPT-4, represents a significant improvement over previous versions. While some consider it the fourth revision, it could also be regarded as the fifth due to major upgrades in GPT-3.5. Like its predecessors, GPT-4 is a generative pre-trained transformer but benefits from an expanded dataset and the ability to process both images and text [BCE+23, Ope24]. For people working in languages beyond English, GPT-4 is a valuable tool, particularly for those, like the author of this thesis, who are focusing on Estonian. Performance has also seen a marked increase, with GPT-4 now ranking in the top 10%, up from the bottom 10% with GPT-3.5 [BCE+23, Ope24]. However, reliability remains an issue.

2.3 Related work

Finding similar work can be challenging as both ChatGPT and GPT-4 are relatively new. While research has been conducted on generating Wordnet, the majority of it involves using existing Wordnets or other resources or translating Wordnets from one language to another. Lam et al. [LATK14] utilized machine translation and a singular bilingual dictionary to generate Wordnet for both resource-rich and resource-poor languages. Similarly, Khodak et al. [KRFA17] also used machine translation and existing resources to generate Wordnet, but their focus was on resource-rich languages such as French and Russian. It remains unclear how their model would perform in resource-poor languages like Estonian.

The closest related work discovered involves generating lexicon entries using ChatGPT. Schryver et al. [dS23] describe this as a shift towards machines taking over the majority of lexicography work while humans serve as validators. In a YouTube video[10], the authors express optimism about ChatGPT, claiming that the future is here. Nonetheless, in their paper, they remain cautiously optimistic, pointing out that *the mere existence of ChatGPT gives us the illusion that this is possible*.

[10]On how ChatGPT can take over all of the dictionary maker's tasks. https://www.youtube.com/watch?v=mEorw0yefAs. Accessed: 2024-04-08

3 Methodology

This section is going to explain the methodology used in this study. It is going to provide a detailed explanation of the tools utilized, the process of generating the Wordnet, the output produced by the GPT-4 language model and the precise functions executed by the Python script. The objective of this section is to provide a clear and comprehensive explanation of the methodological underpinnings of the study.

3.1 Wordnet generation

In this study, a Python script was developed to generate Wordnet. The script is constructed as Jupyter Notebook[11] file, allowing Python code blocks to be run separately. If an error occurs, XML files can be saved and other operations can be performed. Python was chosen due to its simplicity and ease of use for quickly developing scripts. Since this study did not require any complex applications to be developed, Python was the perfect choice.

The initial step in generating Wordnet using GPT-4 was to ensure that GPT-4 was up and running. The University of Tartu is a customer of OpenAI, which enables university personnel to use OpenAI's language models via the Microsoft cloud service Azure[12]. Once access was granted, the next logical step was to experiment with GPT's API. OpenAI provides its own Python library[13], which streamlines the process of using GPT and makes it fast and effortless. To stay as consistent as possible, the *gpt-4-1106-preview* version for GPT-4 was chosen to conduct all experiments.

3.1.1 Generating with XML

After getting access and setting up GPT-4, the next task was to determine the necessary GPT-4 prompts for generating content. Initially, the focus was only on the English side, including the English prompt and the creation of an English Wordnet. As the development of the prompt began, the thought of how the output would look came up. Initially, it was decided that GPT-4 would produce all meanings and Wordnet relationships and present them in XML format.

In Figure 5, the initial output for generating word meanings is displayed. The XML was parsed to retrieve the word, meaning and example, which were then passed back to GPT-4 to acquire Wordnet relations. Figure 6 depicts the resulting relations generated by this process. These relations were subsequently added to the definition section of the XML. Each relation is generated separately.

[11] Jupyter Notebook. https://jupyter.org/. Accessed: 2024-03-18

[12] Microsoft Azure. https://azure.microsoft.com/. Accessed: 2024-03-09

[13] OpenAI library. https://github.com/openai/openai-python. Accessed: 2024-03-09

```
<definitions>
    <definition>
        <word>[Given word]</word>
        <type>[adjectives/adverbs/conjunctions/determiners/nouns/
            prepositions/pronouns/verbs]</type>
        <meaning>[Meaning of the word]</meaning>
        <example>[An example sentence with given word]</example>
    </definition>
    ...
</definitions>
```

Figure 5. Initial output for getting word definitions.

```
<[relation]s>
    <[relation]>[first relation]</[relation]>
    <[relation]>[second relation]</[relation]>
    <[relation]>[third relation]</[relation]>
    ...
</[relation]s>
```

Figure 6. Initial output for getting word definitions.

Once the initial prompts were established, automation was the next step. The following procedure was created:

1. Reading in a text file and processing it line by line, where each line contains a unique word.
2. Creating a root for the XML file.
3. Obtaining all meanings of the word using GPT-4 with Figure 5 format.
4. Iterating over each meaning.
5. Iterating over each relation type and generating relations using GPT-4 in Figure 6 format.
6. Appending the generated relations to the definition XML section.
7. Returning the generated definition and relation XML to the root XML.

Once the initial script was created, Estonian prompts were created. At first, the Estonian prompts were identical to the English ones, with the only difference being that GPT-4 would receive an Estonian word and generate output in Estonian accordingly.

3.1.2 Generating without XML

During one of the conversations with supervisors, the topic of whether GPT-4 should produce output in XML format arose. The rationale behind this consideration was that generating both Wordnet parts and XML could potentially increase the likelihood of errors. Therefore, it was ultimately determined that XML generation would be excluded from GPT-4's capabilities and instead accomplished through a Python script to convert GPT-4 output into XML.

```
[Given word]
[adjectives/adverbs/conjunctions/determiners/nouns/prepositions/
    pronouns/verbs]
[Meaning of the word]
[An example sentence with given word]
...
```

Figure 7. Reworked output for getting word definitions.

```
[First relation word]
[Second relation word]
[Third relation word]
...
```

Figure 8. Initial output for getting word definitions.

Figures 7 and 8 display the updated GPT-4 outputs for retrieving word definitions and relation words, respectively. The script had to be changed to convert the GPT-4 output into XML format. The final Python script has the following procedures:

1. Reading in a text file and processing it line by line, where each line contains a unique word.
2. Creating a root for the XML file.
3. Obtaining all meanings of the word using GPT-4 in Figure 7 format.
4. Iterating over each meaning.
5. Creating a new string in XML format with all necessary parts filled in.
6. Iterating over each relation type and generating relations using GPT-4 in Figure 8 format.
7. Appending the generated relations to the string created in step 5 in XML format.

8. Finding the corresponding word in the actual Wordnet and appending it to the string.

9. Parsing the string and appending it to the root XML.

10. After all words are processed, save the XML file.

Regarding the Estonian prompts, it was decided to fully translate the prompts from English to Estonian. The underlying logic is identical to that of the English version. The decision to translate the Estonian prompts came from an experiment where two prompts were taken - one English prompt but specified to output Estonian and another prompt translated to Estonian. These two prompts were used to generate Wordnet for a couple of words and it came out that Estonian prompts yielded better results in generating Estonian Wordnet. Since only the prompts were different between the English and Estonian Wordnet generations, approximately 95% of the code is shared between the two generations.

3.2 Evaluation

After creating the script, the author encountered a challenge in testing the Wordnet generated by GPT-4. During the standard meetup between the author and supervisors, two solutions were proposed: asking GPT-4 to evaluate the results or using the Natural Language Toolkit (NLTK)[14] Wordnet to verify the generated Wordnet. Ultimately, the second option was chosen due to its faster and more cost-effective nature, as the first option would require checking every generated meaning and its relations individually, which would become rather expensive quickly. Additionally, there would not be any ground truth if GPT-4 would evaluate itself, which would require additional evaluation of the generated results. However, the first choice also presented a new problem: how to match the generated Wordnet words with their actual Wordnet counterparts. The idea of asking GPT-4 for assistance arose once again, but it was decided not to use it because of the same reasoning as previously mentioned.

Instead, the author decided to compare the generated meaning for a word with the actual Wordnet meanings for that word. This was accomplished by taking the intersection between GPT-4 generated meaning and Princeton WordNet [Uni10] or EstWN [est] meanings. The underlying concept was that similar meanings would have more common words, leading to the conclusion that the word with the highest count of common words was the correct one. However, there are some limitations to the use of the common word solution. Firstly, while it is unlikely to generate spelling errors in English, it may do so for languages with smaller training data, such as Estonian. Additionally, though the grammar may be correct, GPT-4 may generate text in different grammatical cases for

[14]Natural Language Toolkit. https://www.nltk.org/. Accessed: 2024-03-10

specific words, causing the intersection not to recognize them as the same word. One possible solution to this issue is to take the lemma of the word, which would ensure that words with differing grammatical cases are recognized as having the same meaning by the intersection.

In summary, a two-part process will be employed to evaluate the Wordnets generated by GPT-4. Firstly, a Python script will automatically match the generated words with the actual Wordnet words and synsets, making it easy to calculate the accuracy of the model. The number of matched words, over-generated words and not-generated words can be determined on a word-by-word basis or for the entire generated Wordnet. However, it is important to note that this approach has limitations, as matching words solely based on their description may not always be accurate, resulting in mismatched words. Additionally, GPT-4 may generate mistakes. The second part of the evaluation process involves a manual review, where a subset of the Wordnet is assessed by the supervisors and the author, who provide their opinions on the generated GPT-4. While this human review method is not perfect, as humans can also make mistakes, it offers valuable insights into the accuracy and effectiveness of the Wordnets generated by GPT-4.

3.3 Technical challenges

Throughout the script's development, the author faced challenges with the output generated from GPT-4. There were multiple instances of error handling, including issues with the OpenAI library, such as *ServiceUnavailableError*, *APIError*, *InvalidRequestError*, *RateLimitError* or the content filter being activated. However, the most significant obstacles arose from GPT-4's generated output due to GPT-4 being flexible and unrestricted in its generated output.

While using the section 3.1.1 prompts to generate the Wordnet with XML, the author encountered issues with incorrectly generated or missing XML tags. To address this, a separate function was created to ensure the validity of the XML and the existence of all necessary tags. Consequently, it was decided to switch to using the section 3.1.2 prompts instead, simplifying the checking process to a line count modulo four. However, this approach was not without flaws, as GPT-4 occasionally generated multiple meanings and examples under a single entry for specific words, passing the modulo check. To address this, the author specified in the prompt that each entry should only have one meaning and example.

Although the aforementioned paragraph highlights the most significant aspects of error handling, the author had to make several adjustments to the prompt and closely examine GPT-4's output, particularly when it involved generating XML. To ensure and check GPT-4's generated output, the author even devised two extra files: one for malformed outputs that failed to pass the XML or modulo checks or when GPT-4 produced no output at all and another for logging GPT-4 output exactly as is.

3.4 Experiment setup

Some experiments and comparisons will be conducted to generate Wordnet. The first experiment will compare the correctness of the English Wordnet and the Estonian Wordnet. This experiment will determine the quality of the Estonian Wordnet generated and GPT-4's ability to use Estonian. The second experiment involves comparing GPT-4 generated Wordnet by altering the prompt. This experiment will showcase the vast differences in the output generated by the GPT-4. Furthermore, by changing the prompt, it is possible to instruct GPT-4 to refer to the NLTK Wordnet and generate content based on the actual one. This experiment will help determine whether GPT-4 has seen the Wordnet or not, which can be inferred from the differences between specifically instructed and non-instructed generations.

4 Results

In this section, the outcomes are delved into and the Wordnets generated by GPT-4 are examined. Initially, a correctness analysis is conducted, providing an overview of how well the generated Wordnet synsets match actual Wordnet synsets. Secondly, it is explored whether instructing GPT-4 to seek out actual Wordnet and generating from that improves the results. Lastly, the performance of GPT-4 concerning resource-rich and resource-poor languages is examined. Are they comparable in any way? To make it more concise and readable, the following English words were selected for generating English Wordnet: *crane*, *computer*, *university*, *tongue*, *carrot*, *jumping*, *bark*, *headphones*, *frog* and *flowers*. For Estonian Wordnet, the following words were selected: *keel (language, tongue)*, *kurg (bird crane)*, *lill (flower)*, *pall (ball)*, *tihane (bird tit)*, *kapsas (cabbage)*, *ülikool (university)*, *guugeldamine (googling)* and *arvuti (computer)*.

Please note that in the upcoming and previously mentioned sections, there will be a reference to the *PWN* and *EstWN*. These refer to the true Wordnets, Princeton WordNet [Uni10] and Estonian Wordnet [est], respectively. Estonian Wordnet can be found in the EstNLTK Python library and the Princeton WordNet is available in the NLTK library.

4.1 Statistics

The paper's title, *Sparks of Artificial General Intelligence: Early experiments with GPT-4* [BCE+23] by OpenAI, provides a good summary of the thesis results. While some synsets generated through the process are pretty good, others may not be up to the mark.

Table 1. Statistics for differently generated meanings.

Generation method	**Generated**	**Actual**
English Wordnet	36	60
Estonian Wordnet	20	21
Eng Wordnet (new prompt)	35	60
Est Wordnet (new prompt)	25	21

Table 2. Statistics for differently generated relations.

Generation method	**Generated**	**Actual**	**Overlap**	**Over gen**	**Under gen**
English Wordnet	706	401	40	666	350
Estonian Wordnet	1099	576	15	900	557
Eng Wordnet (new prompt)	1042	385	108	669	265
Est Wordnet (new prompt)	3194	754	21	2877	733

In the tables 1 and 2, it is possible to find statistics regarding the meanings and the relations. The first column in both tables lists the generation method, which corresponds to the sections 4.2.1, 4.2.2, 4.3.1 and 4.3.2 prompts. The second column, *Generated*, shows the total number of meanings or relations generated by GPT-4, while the *Actual* column displays the number of actual meanings or relations. The following three columns are for table 2 only. The *Overlap* column shows the number of generated and actual relations that overlap, while *Over gen* and *Under gen* indicate the number of relations over and under-generated by GPT-4, respectively.

After reviewing table 1, it appears that GPT-4 performed better on Estonian Wordnet than English Wordnet. However, the situation is more nuanced than it initially seems and requires further investigation. Table 2 reveals that GPT-4 tends to over-generate and has limited overlap with actual Wordnets, which may suggest poor results. While the over-generation observation is valid, the situation is more intricate and warrants a deep-dive analysis, similar to the one conducted for generated meanings, to understand GPT-4's generated results fully.

Please note that these results should be taken with a grain of salt as the matching process between the generated and actual meanings was done automatically, as described in the section 3.2 and may produce mismatches. Nonetheless, it provides a helpful overview of the data.

4.2 Correctness

The previous section provided a quick overview of the upcoming examples and their expected results. However, the following sections will delve deeper into the details and provide specific examples. This will help to explain both the positive and negative aspects of the results generated by GPT-4.

4.2.1 English Wordnet

Firstly, the focus will be on the English Wordnet, but before that, it is worth mentioning that all of the generated files can be viewed at the author's GitHub repository[15]. The generation of meanings for a word can be classified into two groups: when a word has one or a few meanings and when a word has multiple meanings. In cases where a word has only a few meanings, GPT-4 is generally accurate in generating the total number of meanings, sometimes being off by just one. A few examples of such words are *crane*, *computer* and *university*. Here are the generated and actual meanings for the word *crane*.

- Generated meanings for *crane*

[15]Repository for code and generated files: https://github.com/EduardNot/Wordnet-with-GPT4

1. *A tall, long-legged, long-necked bird, typically with white or gray plumage and often with tail plumes and patches of bare red skin on the head. Cranes are noted for their elaborate courtship dances.*
2. *Large tall machine that moves heavy objects by suspending them from a projecting arm or beam.*
3. *To stretch out one's body or neck in order to see something.*
4. *To move (a heavy object) with a crane.*

- Actual meanings from PWN for *crane*, excluding names

1. *Large long-necked wading bird of marshes and plains in many parts of the world.*
2. *Lifts and moves heavy objects; lifting tackle is suspended from a pivoted boom that rotates around a vertical axis.*
3. *Stretch (the neck) so as to see better.*

Example for word *computer*.

- Generated meanings for *computer*

1. *An electronic device for storing and processing data, typically in binary form, according to instructions given to it in a variable program.*
2. *A person who makes calculations or computations, especially one employed to do this in an observatory.*

- Actual meanings for *computer*

1. *A machine for performing calculations automatically*.
2. *An expert at calculation (or at operating calculating machines).*

And lastly, for *university*.

- Generated meanings for *university*

1. *An educational institution designed for instruction, examination and both, of students in many branches of advanced learning, conferring degrees in various faculties and often embodying colleges and similar institutions.*
2. *The members of this institution as a societal body, including students and faculty.*

- Actual meanings for *university*

1. *A large and diverse institution of higher learning created to educate for life and for a profession and to grant degrees.*
2. *The body of faculty and students at a university.*
3. *Establishment where a seat of higher learning is housed, including administrative and living quarters as well as facilities for research and teaching.*

The situation can quickly become complicated when examining a group of words with multiple meanings. GPT-4 faces a challenge in accurately distinguishing and generating all possible interpretations. An example of this can be seen with the word *bark*.

- Generated meanings for *bark*

  1. *(Of a dog) make a loud, rough noise.*
  2. *The sharp explosive cry of a dog, fox and seal.*
  3. *Irritably or angrily snap or speak at.*
  4. *The tough protective outer sheath of the trunk, branches and twigs of a tree or woody shrub.*
  5. *Strip the bark from (a tree or piece of wood).*
  6. *A sailing ship, typically with three masts, in which the foremast and mainmast are square-rigged and the mizzenmast is rigged fore-and-aft.*

- Actual meanings for *bark*

  1. *Tough protective covering of the woody stems and roots of trees and other woody plants.*
  2. *A noise resembling the bark of a dog.*
  3. *A sailing ship with 3 (or more) masts.*
  4. *The sound made by a dog.*
  5. *Speak in an unfriendly tone.*
  6. *Cover with bark.*
  7. *Remove the bark of a tree.*
  8. *Make barking sounds.*
  9. *Tan (a skin) with bark tannins.*

The worst example the author saw was with the word *jumping*; GPT-4 generated only three meanings, while the PWN had seventeen.

After examining how meanings were generated, it is reasonable to assume that generating relations would yield similar results. However, if a word has multiple meanings,

GPT-4 may struggle, while fewer meanings result in better performance. Unfortunately, generating relations is more challenging than generating meanings. The author notes that the most significant discrepancy between generated and actual meanings was for the word *jumping* and this issue is present for every relation, but in reverse.

```
<generated>
    <synonyms> ['stork', 'heron', 'egret'] </synonyms>
    <hyponyms> ['common_crane', 'sandhill_crane', 'whooping_crane', '
        brolga', 'sarus_crane', 'demoiselle_crane', 'red-crowned_crane
        ', 'grus', 'black-necked_crane', 'blue_crane', '
        grey_crowned_crane', 'wattled_crane', 'white-naped_crane', '
        hooded_crane', 'siberian_crane', 'black_crowned_crane'] </
        hyponyms>
    <meronyms> ['plumage', 'beak', 'legs', 'wings', 'tail', 'claw']
        </meronyms>
    <antonyms> [] </antonyms>
    <hypernyms> ['bird', 'wading_bird', 'gruidae'] </hypernyms>
    <holonyms> ['gruidae', 'aves', 'chordata', 'animalia', '
        crane_family', 'birds'] </holonyms>
</generated>
<actual>
    <synonyms> ['crane'] </synonyms>
    <hyponyms> ['whooping_crane'] </hyponyms>
    <meronyms> [] </meronyms>
    <antonyms> [] </antonyms>
    <hypernyms> ['wading_bird'] </hypernyms>
    <holonyms> ['gruidae'] </holonyms>
</actual>
```

Figure 9. Crane (bird) generated and actual relations.

The generated and actual relations for bird crane and machine crane are depicted in Figures 9 and 10. As previously stated, GPT-4 generates an excessive amount of relations compared to the actual relations, which are relatively few. That being said, not all generated results are bad. When considering synonyms, there are typically two cases to consider with GPT-4: either it generates suitable synonyms or it generates synonyms that fit better with other relations, such as hyponyms or meronyms. This can be observed in Figure 9, where synonyms often fit better with hyponyms or meronyms. For example, when generating synonyms for *university*, GPT-4 provided several suitable options, including *academy, institute, school, educational institution, college, learning establishment, higher education institution,* and *degree-granting institution*. This trend holds true for all relations: when GPT-4 generates good results for a specific relation, all results tend to be quite good, but when it does not, the results are just bad. Generally, hyponyms and hypernyms were quite good, with few issues beyond occasionally being missing and sometimes labeled as synonyms. Meronyms and holonyms also produced

```
<generated>
    <synonyms> ['derrick', 'hoist', 'mobile_crane', 'tower_crane', '
        jib'] </synonyms>
    <hyponyms> ['tower_crane', 'floating_crane', 'mobile_crane', '
        crawler_crane', 'telescopic_crane', 'rough_terrain_crane', '
        overhead_crane', 'loader_crane'] </hyponyms>
    <meronyms> ['boom', 'cab', 'jib', 'hoist', 'pulley'] </meronyms>
    <antonyms> ['lower', 'drop', 'depress'] </antonyms>
    <hypernyms> ['construction_equipment', 'machine', 'lifting_device
        '] </hypernyms>
    <holonyms> ['construction_site', 'building_equipment'] </holonyms
        >
</generated>
<actual>
    <synonyms> ['crane'] </synonyms>
    <hyponyms> ['davit', 'derrick', 'transporter'] </hyponyms>
    <meronyms> [] </meronyms>
    <antonyms> [] </antonyms>
    <hypernyms> ['lifting_device'] </hypernyms>
    <holonyms> [] </holonyms>
</actual>
```

Figure 10. Crane (machine) generated and actual relations.

good results without any outrageous outcomes. It is difficult to determine antonyms since most were empty, but the ones that were not had positive results. Overall, the results fell into two distinct categories: either good or bad. However, the results were still surprisingly good.

4.2.2 Estonian Wordnet

Considering the unexpected outcomes observed with the implementation of the English Wordnet, it naturally raises curiosity regarding the feasibility and potential benefits of developing an Estonian Wordnet. Firstly, it is not surprising that generating meanings for Estonian words presents similar challenges to those encountered with English words - when a word has multiple meanings, GPT-4 may struggle to generate all of them. In contrast, with words that have few meanings, GPT-4 tends to perform well. For instance, GPT-4 generated accurate meanings for Estonian words like *pall (ball)* and *tihane (bird tit)*, which have only a few meanings. Take the word *pall (ball)* for example.

- Generated meanings for *pall*

    1. *Ümar ja veerev, tavaliselt kummist või muust sarnasest materjalist eset, mida kasutatakse mängimiseks.*

2. *Midagi, mis sarnaneb oma kuju või olemuse poolest palliga.*
3. *Mõõtühik vinna-, klots- või rullmaterjali hulga mõõtmiseks.*

- Actual meanings for *pall*

1. *Ümmargune põrkav mänguasi*.
2. *Pallitaoline moodustis*.
3. *Numbriline hindamisühik kokkuleppelises hindamissüsteemis*.

And an example for *tihane (bird tit)*.

- Generated meanings for *tihane*

1. *Väike põõsastes ja puudel pesitsev laululind, kes sööb põhiliselt putukaid ja seemneid.*

- Actual meanings for *tihane*

1. *Lühikese peenikese nokaga väike lind (Parus) (EKSS).*

Interestingly, taking a look at the word *kurg (throat, bird crane)* and the generated meanings.

- Generated meanings for *kurg*

1. *Linnuliik, millel on pikk kael ja jalad ning suudab lendada pikkadel vahemaadel.*
2. *Filiaalimine(organismis) vahel põhja poole hingetoru kaela taga, ühendades selle suu ja ninaõõnega.*
3. *Emas- või isasloomad oma järglaste sigitamiseks vajaliku kehaorganiga.*

- Actual meanings for *kurg*

1. *Suur, pika noka ja kaela ning kõrgete jalgadega.*
2. *Kõver käepidemega puu (nt vannasadral) või raud (hõlmadral) adra juhtimiseks.*

If someone unfamiliar with Estonian were to examine the meanings produced, they might believe that GPT-4 only missed one definition, which is not so terrible. However, a person who is familiar with the language would regard it as rubbish. Why? Because GPT-4 only accurately generated one definition for *crane* as a bird, which is the initial one. For the second definition, GPT-4 generated *kurk*, which is *throat*. It is difficult to determine why GPT-4 produced *throat*, but it may be due to *kurgu*, which is a grammatical case

for *kurk (throat, cucumber)*. Finally, the third definition generated by GPT-4 is entirely incorrect and nonsensical. It is perplexing as to why GPT-4 generated this definition or where it came from. Taking a look at the words with many meanings like *keel (tongue, language)*, the situation is not better. GPT-4 only generated four meanings out of nine.

As previously noted, there were concerns that GPT-4 may encounter difficulty with words having multiple meanings. However, it remains to be seen whether GPT-4 can successfully generate word relationships in Estonian. As with English Wordnet, synonyms pose similar challenges in that some are apt, while others would benefit from alternative relations. For instance, with the word *keel (tongue, language)*, the suggested synonyms would be more fitting as hyponyms: *murre, dialekt, emakeel, kõnekeel, rahvakeel, sugulaskeel* and *võõrkeel*. Conversely, *ülikool (university)* illustrates a positive example, with suggested synonyms including *kõrgkool, akadeemia, kolledž* and *instituut*. It should be noted that GPT-4 generated more grammatical errors in Estonian, with instances of misspelled words and peculiar grammatical cases. On the whole, hyponyms and hypernyms were both accurate. However, some odd words were included in the former, such as *fin, kil, uim, köü, köis* and *doctor* for *keel (tongue, language)* and in the latter, with examples like *string instrument komponent (instrument component)* and *string vibrateeriv osa (vibrating piece)* for the same word, but different meaning. It is worth noting that GPT-4 included *string*, which is an English word. As with English Wordnet, antonyms were nonexistent. In conclusion, GPT-4 generated Estonian Wordnet with similar mistakes and challenges as those encountered with English Wordnet, albeit with more grammatical errors and overall inaccuracies.

4.3 Contamination

The performance of GPT-4 is generally mixed, with the biggest problem being its inconsistency at times. Initially, the prompt only required the generation of meanings and their relationships, but this has since been refined to match actual Wordnets. These changes allow for a better evaluation of the system's capabilities and its level of reliance on pre-existing knowledge.

4.3.1 Princeton WordNet contamination

Starting by examining the meanings of words, right from the start, there is an intriguing observation. During the experiments, the author did not notice GPT-4 generating any persons or places for *crane* until the prompt specified that it must be generated as it appears in the PWN. At this point, an additional meaning was generated: *United States writer (1871-1900)*. However, two other meanings, *United States poet (1899-1932)* and *a small constellation in the southern hemisphere near Phoenix*, were still missing. While examining the definitions, the author noticed that the generated meanings were identical to those in the PWN. This seems to confirm that GPT-4 is contaminated with

Princeton WordNet. However, it remains perplexing why GPT-4 could not generate all the meanings, including the three odd ones.

Moving on to the word *tongue*, GPT-4 generated five meanings when prompted, down from the previous six. The fascinating part is that the five meanings generated by GPT-4 were word-for-word identical to the actual meanings. For the words *university*, *bark*, *flower* and *carrot*, some meanings were missing, just like *tongue* had missing meanings. Now, taking a look at the three unusual meanings, the first is for *jumping*, where GPT-4 generated *unusually high*. It is difficult to say why it generated this. Still, it may be related to the meaning of *be highly noticeable*, as seen in the example *jumping prices at the grocery store caused concern among the customers*. The next word is *headphones*, where GPT-4 generated the meaning of *a device that holds a pair of small loudspeakers, each in a separate earpiece, that are placed over a user's ears*, which is essentially a duplicate meaning. Lastly, for the word *frog*, GPT-4 generated the meaning of *the triangular elastic part of a horse's foot, helps prevent slipping and assists blood circulation*. Apart from these three odd meanings, all other meanings for *jumping*, *headphones* and *frog* were correct and identical to those in PWN, but of course, some additional meanings were also missing. The only perfect one was *computer*.

Given the new prompt specifications, it may seem that the generated meanings for words are nearly identical word-for-word. Now, it is reasonable to assume this would also apply to relations. Fortunately, GPT-4 has begun generating more relations than actually exists. Furthermore, these relations are not entirely correct at this time either.

Although some of the relationships generated by the AI remained unchanged, such as with the *crane* meanings, improvements were seen with certain words, such as *computer* as a machine. This can be observed in Figure 11, which shows that the generated relations were quite accurate, with the exception of hypernyms where the AI-generated words like *calculator*, *data processor*, *electronic computer* and *information processing system* as synonyms. Additionally, the AI struggled to generate all the meronyms present in the PWN, which was consistent with its difficulty in generating multiple meanings for a single word. A new issue arose with the introduction of the latest prompt, as certain instances revealed that GPT-4 generated insufficient output. This is demonstrated in Figure 12, where not a single hyponym was produced. Despite generating fewer relations overall with the new prompt, the same concerns identified in section 4.2.1 persist.

4.3.2 Estonian Wordnet contamination

Following an investigation into the presence of the Princeton WordNet in GPT-4, attention is shifted to the Estonian side. The focus now is to determine whether GPT-4 has been affected by EstWN. Initial indications suggest that GPT-4 has not been impacted by the Estonian Wordnet, as it did not produce any meanings that closely align with those found in the EstWN. In fact, it appears that GPT-4 generated poorer results when presented with a specific prompt as opposed to an unspecified one, in the author's estimation.

```
<generated>
    <synonyms> ['calculator', 'computing_machine', 'computing_device'
        , 'data_processor', 'electronic_computer', '
        information_processing_system'] </synonyms>
    <hyponyms> ['analog_computer', 'digital_computer', 'home_computer
        ', 'node', 'number_cruncher', 'pari-mutuel_machine', '
        predictor', 'server', 'turing_machine', 'web_site', 'webmaster
        '] </hyponyms>
    <meronyms> ['cpu', 'hardware', 'memory', 'storage', 'peripheral',
         'software'] </meronyms>
    <antonyms> [''] </antonyms>
    <hypernyms> ['machine', 'calculator', 'data_processor', '
        electronic_computer', 'information_processing_system', 'expert
        '] </hypernyms>
    <holonyms> ['circuitry', 'hardware', 'platform', 'system'] </
        holonyms>
</generated>
<actual>
    <synonyms> ['computer', 'computing_machine', 'computing_device',
        'data_processor', 'electronic_computer', '
        information_processing_system'] </synonyms>
    <hyponyms> ['analog_computer', 'digital_computer', 'home_computer
        ', 'node', 'number_cruncher', 'pari-mutuel_machine', '
        predictor', 'server', 'turing_machine', 'web_site'] </hyponyms
        >
    <meronyms> ['busbar', 'cathode-ray_tube', '
        central_processing_unit', 'chip', 'computer_accessory', '
        computer_circuit', 'data_converter', 'disk_cache', 'diskette',
         'hardware', 'keyboard', 'memory', 'monitor', 'peripheral'] </
        meronyms>
    <antonyms> [] </antonyms>
    <hypernyms> ['machine'] </hypernyms>
    <holonyms> [] </holonyms>
</actual>
```

Figure 11. Computer (a computer) generated, with a new prompt and actual relations.

The results are becoming increasingly concerning as GPT-4 generates more incomprehensible meanings and even generates more meanings than necessary. This was a rare occurrence without the specified prompt, but it has now been observed multiple times. Looking at more specific examples, when given the word *keel (tongue, language)*, GPT-4 generated the usual meanings such as an organ, language and instrument string. Previously, it did generate the meaning *mõnes purjetamise või sõudmise kontekstis kasutatav terav piklik osa, mis ulatub allapoole veesõiduki põhja, et parandada stabiilsust*. Instead, it produced two new meanings, *keeleteaduslik tüpoloogia tanüümiks*

```
<generated>
    <synonyms> ['calculator', 'reckoner', 'figurer', 'estimator', '
        computing_machine', 'computing_device', 'data_processor', '
        electronic_computer', 'information_processing_system'] </
        synonyms>
    <hyponyms> [''] </hyponyms>
    <meronyms> [''] </meronyms>
    <antonyms> [''] </antonyms>
    <hypernyms> ['expert'] </hypernyms>
    <holonyms> [''] </holonyms>
</generated>
<actual>
    <synonyms> ['calculator', 'reckoner', 'figurer', 'estimator', '
        computer'] </synonyms>
    <hyponyms> ['adder', 'number_cruncher', 'statistician', '
        subtracter'] </hyponyms>
    <meronyms> [] </meronyms>
    <antonyms> [] </antonyms>
    <hypernyms> ['expert'] </hypernyms>
    <holonyms> [] </holonyms>
</actual>
```

Figure 12. Computer (an expert) generated, with a new prompt and actual relations.

nimetamissüsteem and *keeleteaduse mõiste, mille abil jaotatakse keeli nende struktuuri või päritolu järgi*. Upon closer examination, the first meaning is completely rubbish, but there is enough information to argue that these are duplicates. Furthermore, with the new prompt, duplicates are now being generated by GPT-4, which can be seen in multiple words. This is a departure from the original prompt, where the AI rarely generated duplicates. One significant challenge that GPT-4 encounters is its apparent inability to differentiate between words that involve a letter change when transitioning between different grammatical cases, such as *kurk (throat, cucumber)* and *kurg (bird crane)*. In certain grammatical contexts, *kurk (throat, cucumber)* may appear with a *g* and this can perplex GPT-4, leading it to associate the lemma with *kurg (bird crane)* instead. This can be observed in the meanings generated for *kurg (bird crane)*.

1. *Inimese kaelasopa sees paiknev luu- ja lihaseline moodustis, mis on nikastamise korral inimese hääleaparaat ja võimaldab neelamist.*

2. *Suur veetüüpi lind, kellel on pikk kael ja jalad ning sirge terava otsaga nokk, kütk roosteehitus, eriti rännulind.*

3. *Nõu (eriti pudeli) kitsas suue.*

4. *Aia- ja köögiviljakultuur.*

The correct interpretation for *kurg (bird crane)* is the bird crane, as per the second definition. GPT-4 generated meanings, which are the first and fourth, apply to the word *kurk*, which means throat or cucumber. Meanwhile, the third definition is entirely false as it mistakenly interchanges *kurk* with *kael*, which means neck. Upon examining the sample sentence provided for the third definition, *tema voolas viski otse pudelikurgust kurku*, it becomes apparent that *pudelikurgust (throat of a bottle)* should actually be *pudelikaelast (neck of a bottle)*. It is unclear why GPT-4 confused these two words since there is no such word as *pudelikurk (throat of a bottle)*. Moving on to the word *lill (flower)*, GPT-4 produced an additional meaning compared to the previous attempt. The new definition is *noor, tütarlaps või naine*, which translates to *young girl or woman*, which is once again challenging to comprehend. It is unclear how GPT-4 drew this connection. In section 4.2.2, all of the meanings for the word *pall (ball)* were generated and all of them were correct. However, with the new prompt, GPT-4 only produced two definitions for *pall (ball)*, where one was accurate for a playing ball and the second was *väline, eriti sädelev, joonistus- või värvilise mustriga kaunistus taimel*, which is a description of a Christmas tree ornament. Similar to the old prompt, GPT-4 correctly generated the definition for *tihane (bird tit)*, but when it comes to the word *kapsas (cabbage)*, it falls apart once again. Previously, GPT-4 missed only one definition and generated two out of three meanings. This time, GPT-4 generated three meanings, but two of them are complete nonsense - *raha (money)* and *segadus, sass, jama, keeruline olukord (difficult situation)*. This time around, GPT-4 provided only one definition out of two for *ülikool (university)*, but the definitions are accurate. Although *guugeldamine (Googling)* does not appear in Wordnet, GPT-4 generated the correct definitions on both occasions. However, when examining the word *arvuti (computer)*, GPT-4 generated five definitions that were overly broad and some were incorrect.

1. *Digitaaltehnika saavutuste põhjal loodud seade, mis on võimeline töötama kindla programmi järgi ning tegema teatavaid toiminguid ettenähtud tulemuse saavutamiseks; peaaju; infotehnoloogia.*

2. *Kõik ca arvutit, mida üks isik või organisatsioon omab ning mida ta või see saab tehnoloogia abil tööle panna nii, et tulemused vastavad kindlatele nõuetele.*

3. *Objekt või subjekt, mis loob, säilitab ja töötleb informatsiooni teatud viisil, mis võimaldab sellel teha kindlaid toiminguid ning saavutada ettenähtud tulemusi.*

4. *Disaineri töö, mis seisneb tehnoloogia ja informatsiooni kasutamises, et saavutada teatavaid tulemusi.*

5. *Organisatsioon või isik, kes omab ja kasutab tehnoloogiat ja infotehnoloogiat ning on võimeline saavutama ettenähtud tulemusi.*

Upon reviewing all definitions, it appears that the second part of each definition is a duplicate. Additionally, the fourth definition seems to be industry-specific jargon, while the fifth definition mistakenly refers to a computer as a person or organization. The first three definitions, while broad, do not fully convey the intended definition of the word.

Regarding the relations, the positive news is that they have not deteriorated any further. Unfortunately, the same obstacles persist in generating these relations as previously discussed in sections 4.2.1 and 4.2.2. There is no need to reiterate what has already been stated; the only thing that is worth mentioning is that GPT-4 generated almost three times more relations. This is partially due to AI generating more meanings, but the mistakes are all the same. In total, the updated prompt resulted in worse generated meanings, but the relations remained consistent. It is interesting that GPT-4 has the context of PWN, while of EstWN, it does not.

4.4 Resource-rich versus resource-poor

It is unsurprising that GPT-4 has access to a broader range of English resources, which has led to overall better performance. However, it is surprising that GPT-4 struggled to generate all the meanings with the new specified prompt for a specific word with multiple meanings, even though the generated meanings were identical to those in PWN. On the other hand, it appears that GPT-4 lacks context for actual EstWN, yet it still performed quite well.

The author observed that generating Estonian Wordnet required significantly more time. While generating English Wordnet with the original prompt took around ten to fifteen minutes, generating Estonian Wordnet took up to thirty minutes, but usually around twenty to twenty-five minutes. However, with the new prompt, generating English Wordnet remained the same, but generating Estonian Wordnet took over an hour - specifically, seventy-five minutes. It is unclear why changing the prompt resulted in such a significant slowdown in generating Estonian Wordnet.

To sum up, it is evident that generating the English Wordnet produces superior output at a faster pace. Conversely, the creation of the Estonian Wordnet incurred a significant drawback when GPT-4 was directed to generate content based on the EstWN. While the outcomes are encouraging, it remains the author's opinion that GPT-4's output can be erratic and uncertain.

5 Discussion

The experiment results have provided a mixed outcome. While GPT-4's performance is impressive in some aspects, it falls short in others. The author suggests that one of the issues lies in the difficulty of distinguishing relevant information from irrelevant information for current generative LLMs. Due to their training on vast amounts of data, it is crucial for users to provide appropriate prompts. Providing too little information results in guesswork, while providing too much information leads to hallucinations. Additionally, even minor changes to the prompt can yield vastly different results, presenting the challenge of determining whether the current prompt is optimal or merely a local minimum. This can be observed in sections 4.2 and 4.3. To answer question 4, adding just one extra line can cause GPT-4 output to become word-for-word definitions or produce worse results, even if it takes an additional hour to generate the results.

Regarding the results and to answer research questions 1 and 2, GPT-4 struggled to generate all possible meanings for a word with multiple definitions despite having the context of the PWN and being able to produce accurate, word-for-word meanings similar to those found in the Princeton WordNet. It remains unclear why this was the case, especially since the generated statistics in table 2 indicate that GPT-4 tended to over-generate. Additionally, it is noteworthy that the quality of the Estonian and English Wordnets generated with the first prompt was quite similar, despite GPT-4 having access to a much larger dataset and broader context. Although generative LLMs have made significant progress, they still have a way to go before they can accomplish tasks like the one presented in this thesis topic. In the author's view, it would be more expensive and time-consuming to prompt and analyze GPT-4, correct any errors and discard unnecessary information than to hire a person to construct the Wordnet. This is because a human is still needed to review and correct the output generated by GPT-4.

This all culminates together to answer question 3 that the limitations of GPT-4 and other generative LLMs are apparent when it comes to generating precise results. Take, for instance, the author's work company, which has a license to use GitHub Copilot[16]. After thorough discussions among the author and colleagues, it was agreed that while current LLMs are useful for inspiring creativity, generating more complex code proves challenging. However, when provided with context, such as generating a unit test based on existing code, the AI surprisingly generates good results. However, It should be noted that if the task becomes too complex, such as generating a mock test with a mock API endpoint, the AI's performance degrades significantly. A simple experiment one can undertake is to test the capabilities of generative image AI. Though the AI can produce images of adorable puppies, there may be some errors in the details. However, it is still evident that the image depicts puppies. Conversely, if the AI is instructed to create a blank, white image of nothingness, it will encounter difficulties and errors will be

[16]GitHub Copilot. https://github.com/features/copilot. Accessed: 2024-05-06

noticeable since the image contains something rather than nothing.

Ultimately, it may prove advantageous to utilize an established Wordnet and explore methods for generating a new Wordnet based upon it. Determining the most optimal approach will require extensive experimentation. An alternative possibility could involve organizing the words and tasking GPT-4 with creating a graph similar to a graph problem. There may exist various approaches, but the key takeaway is to leverage existing resources rather than relying solely on GPT-4 to generate from scratch.

6 Conclusion

In this thesis, the effectiveness of GPT-4 was assessed by creating a Wordnet for a resource-rich language, such as English and a resource-poor language, such as Estonian. To accomplish this, a Python script was necessary. This Python script utilized two types of prompts to generate the Wordnets. The first prompt was open-ended, simply requesting the generation of meanings and relations. Conversely, the second prompt specifically instructed the script to generate based on the actual Wordnet. This was implemented to verify whether the existing data had influenced GPT-4. Ultimately, through conducting experiments in this thesis, the research questions proposed in section 1 have been answered and are the following:

1. Research question 1: GPT-4 can generate English Wordnet, but the results are mixed. GPT-4 has difficulties with generating meanings for words that have a lot of meanings, but at the same time, GPT-4 over-generated relations and had a small overlap with the actual relations.

2. Research question 2: GPT-4 can generate Estonian Wordnet, but the results are mixed once again. GPT-4 had similar difficulties as mentioned in research question 1 answer, where GPT-4 struggled to generate meanings for words that had a lot of meanings. Still, the relations were over-generated with a small overlap.

3. Research question 3: The current generation of generative LLMs are not good at generating exact results from nothing, such as this thesis topic required. The current generative LLMs excel at best being creative or producing results based on existing data.

4. Research question 4: Changing prompt can drastically affect the results. When the original prompt in this thesis was changed to generate Wordnet based on actual Wordnets, GPT-4 generated word-for-word exact meanings for English Wordnet. At the same time, the Estonian Wordnet generation suffered from worse results and increased generation time.

In summary, the results themselves were somewhat mixed. In general, GPT-4 performed well when generating multiple meanings for a word but struggled when there were numerous possible meanings. The generated relations were not as accurate as the meanings, with some over-generation and occasional misplacement of relations. Interestingly, while GPT-4 was able to generate word-for-word meanings using English Wordnet, it was not always able to generate all possible meanings for words with many definitions. Conversely, it appears that GPT-4 is not impacted by Estonian Wordnet.

There are various ways to explore the capabilities of future works. A simple approach is to experiment with diverse prompts and even test out new languages. Another possibility is to evaluate different LLMs, including the forthcoming GPT-5. Furthermore, a

supervisor proposed the idea of developing a partially constrained model. This implies that instead of unconstrained output generation, a partially constrained LLM would be limited to producing specific types of output. All of these options are interesting and can yield interesting results.

Overall, the present LLMs require extensive monitoring and are not yet fully developed. The shortcomings of GPT-4 and similar generative LLM technologies become evident when attempting to produce accurate outcomes. The most effective use of current LLMs is within a predefined context, allowing the AI to generate content based on that context. Examples of this include summarizing text or creating unit tests for pre-existing code.

References

[BCE+23] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4. https://arxiv.org/pdf/2303.12712, 2023.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. https://arxiv.org/abs/2005.14165.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. http://arxiv.org/abs/1810.04805.

[dS23] Gilles-Maurice de Schryver. Generative AI and Lexicography: The Current State of the Art Using ChatGPT. *International Journal of Lexicography*, 36(4):355–387, 10 2023. https://doi.org/10.1093/ijl/ecad021.

[est] Estonian Wordnet. https://cl.ut.ee/ressursid/teksaurus/. Accessed: 2024-04-01.

[KRFA17] Mikhail Khodak, Andrej Risteski, Christiane Fellbaum, and Sanjeev Arora. Automated WordNet construction using word embeddings. In Jose Camacho-Collados and Mohammad Taher Pilehvar, editors, *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 12–23, Valencia, Spain, April 2017. Association for Computational Linguistics. https://aclanthology.org/W17-1902.

[LATK14] Khang Nhut Lam, Feras Al Tarouti, and Jugal Kalita. Automatically constructing Wordnet Synsets. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2014. http://dx.doi.org/10.3115/v1/p14-2018.

[Ope24] OpenAI. GPT-4 Technical Report. https://arxiv.org/pdf/2303.08774, 2024.

[OWJ+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018.

[Uni10] Princeton University. About WordNet. https://wordnet.princeton.edu/, 2010.

[Vos02] Piek Vossen. Wordnet, EuroWordNet and Global WordNet. *Revue française de linguistique appliquée*, 2002. https://www.cairn-int.info/journal-revue-francaise-de-linguistique-appliquee-2002-1-page-27.htm.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Eduard Rudi**,
*(*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,

   Extracting Lexical Relations from Large Pre-trained Language Models,
   *(*title of thesis)

   supervised by Mark Fišel and Heili Orav.
   *(*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 3.0, 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.

Eduard Rudi
15/05/2024