UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Sten Marcus Malva

Exploring And Implementing Estonian Text Simplification Using Machine Learning

Bachelor's Thesis (9 ECTS)

Supervisor(s): Eduard Barbu, PhD

Tartu 2024

Exploring And Implementing Estonian Text Simplification Using Machine Learning

Abstract:

The goal of this bachelor's thesis was to explore Estonian text simplification using machine learning and to train such a model with Estonian data. In order to train it, a significant amount of data was gathered, translated, corrected and generated. Then previously implemented algorithms for English were evaluated and one of them was used to take as a base for the Estonian model. Finally a model was trained in three different configurations and evaluated.

Keywords:

Text simplification, Estonian language, machine learning, natural language processing, artificial intelligence

CERCS: P176 Artificial intelligence

Eesti keele lihtsustamise uurimine ja implementeerimine kasutades masinõpet

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärgiks oli uurida Eesti keele lihtsustamise võimalusi masinõppe abil ning treenida selline mudel eestikeelsete andmete peal. Treenimiseks koguti kokku, tõlgiti, parandati ja genereeriti suur hulk andmeid. Seejärel uuriti ja hinnati varem implementeeritud algoritme inglise keele jaoks ning üks neist valiti eestikeelse mudeli aluseks. Viimaks treeniti eestikeelne mudel kolme erineva konfiguratsiooniga, mida hiljem hinnati.

Võtmesõnad:

Keele lihtsustamine, eesti keel, masinõpe, loomuliku keele töötlus, tehisintellekt

CERCS: P176 Tehisintellekt

Table of Contents

Introduction

Some people are unable to access vast amounts of information due to their inability to understand certain languages. Language learning is a time-consuming and some individuals may never fully master a second language. However, there exists an easier way to enhance comprehension of texts. In this thesis, we delve into a process known as text simplification.

Text simplification aids second language learners and individuals with certain cognitive disabilities in comprehending text. Research indicates that, on average, readers need to comprehend 95% of the words in a given text to understand its meaning [1]. This necessity arises because texts are frequently more complex than necessary, whether they are everyday or specialized texts intended for experts in a field.

Text simplification encompasses numerous techniques that vary depending on the language. Some methods focus on lexical simplification by substituting words with less complex alternatives [2], while others aim to simplify text syntactically by altering its structure [3]. Nevertheless, all these approaches share the same objective: to enhance text accessibility for a broader audience while preserving its meaning. Text simplification involves numerous rules and methods to ensure its effectiveness, and there are various criteria to evaluate it.

Text simplification performed by humans is a time-consuming process. Consequently, there is a growing effort to employ computers to enhance the efficiency of this process. Currently, systems have been developed specifically for the English language, utilizing machine learning models that necessitate large amounts of data for training. However, once trained, these models prove to be highly efficient for simplification tasks [4].

For the Estonian language, existing systems primarily rely on algorithmic approaches, which entail defining a set of rules [5] [6]. However, this approach is not always effective because language is not always governed by precise rules, leading these systems to struggle with exceptions. Therefore, this thesis aims to investigate Estonian text simplification using machine learning models.

1 Text Simplification

Text simplification is a process whereby a sentence is revised through various means to create a more readily understandable version. This results in a new sentence that may incorporate different vocabulary, alternative verb conjugations, or a reduction in unnecessary filler words. However, the overarching aim of text simplification remains consistent across contexts: to enhance comprehension for readers and diminish the cognitive effort needed to understand the text [7]. All of the examples to demonstrate different text simplification techniques are taken from our private corpora if not stated otherwise.

1.1 Forms of Simplification

Automatic text simplification is a computational process that utilizes algorithms and natural language processing (NLP) techniques to automatically simplify the content and structure of text. This process typically involves reducing lexical and syntactic complexity, which includes substituting difficult words with simpler alternatives, shortening long sentences, and reorganizing information to enhance comprehensibility while preserving the original meaning and information [8].

Automatic text simplification is related to manual text simplification, wherein a human actively rewrites and simplifies text, in the following ways:

1) Objective: Both manual and automatic simplification techniques aim to enhance the accessibility of text to diverse groups of people. These groups may comprise non-native speakers, children, and individuals with cognitive disabilities [3].
2) Methodology: Manual simplification relies on human intuition, comprehension, and linguistic skills. Automatic simplification, on the other hand, is grounded in programmed rules, machine learning, or a combination of them [3].
3) Efficiency and Scalability: Manual simplification demands time from human experts to generate simplifications. Conversely, automatic simplification can efficiently process large volumes of text, rendering it scalable and efficient compared to manual simplification [9].

Therefore, automatic text simplification aids humans in the text simplification process by harnessing technology to enhance efficiency and accessibility.

This chapter explains three text simplification techniques: lexical simplification, syntactic simplification and conceptual clarification.

1.1.1 Lexical Simplification

Lexical simplification is the most straightforward technique to comprehend. It entails no intricate alterations to the original sentence, such as modifying verb conjugations. The concept is simply to replace original words with new ones that have the same or similar meanings but are easier to understand [1]. According to research in psycholinguistics by Paetzold and Specia [1], English learners typically need to grasp around 95% of the words in a text to achieve a basic understanding of the entire text, even if they don't fully grasp the sentence structure. Moreover, if their intention is to read the text for amusement or as a pastime, they generally need to understand roughly 98% of the words [10]. This underscores the significance of lexical simplification. Merely substituting words can significantly enhance the readability of texts.

To execute lexical simplification effectively, it's essential to discern between words with more complex and simpler meanings. Firstly, we must identify words requiring simplification from the source text. Next, we seek suitable substitutions for these words.

Finally, we determine which substitution best suits the context. Today, this task is typically automated, employing wordnets and trained neural models. However, manually annotated sets of words can also be employed, particularly during model training [1].

To give a better understanding of what lexical simplification looks like, the examplanation from the work of Paetzhold and Specia [1] is used:

The cat perched on the mat. → The cat sat on the mat.

This example illustrates that the word "perched" may be challenging to comprehend. A much simpler alternative in this context is "sat," which is also semantically specific - its meaning cannot be effectively conveyed using other words, thereby lacking a simpler equivalent [11].

Here are some other English lexical simplifications, where the underlined words represent difficult words that are substituted into simpler ones with the same meaning:

- *The government will commence the infrastructure project next month. → The government will start the infrastructure project next month.*
- *The candidate elucidated his position on environmental policies during the debate. → The candidate explained his position on environmental policies during the debate.*

For Estonian, lexical simplification is straightforward using the technique explained above. Here are some examples for Estonian:

- *Põrandal lippab putukas. → Põrandal jookseb putukas.*
- *Raehärra pidas kõnet. → Linnapea pidas kõnet.*
- *Kõik on kriminaalkoodeksi ees võrdsed. → Kõik on seaduse ees võrdsed.*

In these examples we can see that lexical simplification doesn't necessarily only substitute words that are more difficult but also words that are less known. For example "lippab" is more used in everyday speech and so the word "jookseb" has the same meaning but it understood by more people.

1.1.2 Syntactic Simplification

The primary objective of syntactic simplification is to alter the grammatical structure of text to enhance comprehension while retaining its meaning [3]. Typically, this involves breaking down lengthy sentences into shorter and simpler ones. Such simplification renders the text more accessible to individuals with limited familiarity with the language's grammar rules. The extent of syntactic simplification can vary depending on the language. For instance, Chinese lacks verb conjugations and does not distinguish between singular and plural forms, thereby meaning that Chinese is grammatically less complex than English.

Siddharthan outlines three steps in syntactic simplification [3]. Firstly, it involves determining whether a sentence requires syntactic simplification, which can be challenging since certain techniques might seemingly simplify the sentence but could actually make it more difficult to understand. Hence, there's no standardized formula for this task. Once it's established that a sentence can and should be simplified syntactically, the next step is the simplification process itself. This step can take various forms, with several formulas available for the task. However, relying solely on a formulaic approach can be problematic due to language exceptions and variability in word order. Instead, a set of rules is typically employed to minimize errors. Finally, the simplified text is evaluated to ensure that it retains

the original meaning. This evaluation involves calculating scores to assess its grammatical correctness and fidelity to the original meaning.

As mentioned in the paper by Siddharthan [3], a good example for a syntactic simplification, would be:

"I'm highly critical of the decision," said a top aide, Diego Guelar, a former ambassador to the United States. → "I'm highly critical of the decision," said a top aide. This top aide is Diego Guelar, a former ambassador to the United States.

At first, we will notice how the simplified text has more sentences in total. This is because the original sentence had too much information pushed into one sentence. The original sentence talked about a top aide, who said something, and of Diego Guelar, who is a former ambassador of the USA and also the aide who is talked about. We can see, that it can be hard to understand that the top aide is indeed Diego Guelar, because it doesn't specifically explain that he is the top aide in the original sentence. Therefore, we simplify it in a way that makes it clear to the reader that something is first said by the top aide. Then we explain in another sentence that the top aide is a former ambassador of the USA, named Diego Guelar. We can see that the sentence becomes more easier to read and it also

Initially, it's evident that the simplified text comprises more sentences overall. This is due to the original sentence containing excessive information crammed into one sentence. The original sentence discussed a statement made by a top aide and mentioned Diego Guelar, a former ambassador of the USA who is also the aide being referred to. It's apparent that understanding that the top aide is indeed Diego Guelar can be challenging because the original sentence doesn't explicitly clarify that he is the top aide. Therefore, we simplify it by clearly indicating to the reader that a statement is initially made by the top aide. Subsequently, we elaborate in another sentence that the top aide is Diego Guelar, a former ambassador of the USA. This restructuring renders the sentence easier to comprehend while preserving its original meaning.

More examples of rule-based syntactic simplifications:

- Changing passive to active voice: *The book was read by her. → She read the book.*
- *Reducing relative clauses: The man who is talking to John is my uncle. → The man talking to John is my uncle.*
- Dis-embedding relative clauses: *She extracted the cash fine which has outraged him and other clamped motorists. → She extracted the cash fine. It has outraged him and other clamped motorists.*
- Separation of subordinated clauses: *While the law generally supports clampers operating on private land, Mr Agar claims CCS's sign was not prominent enough to be a proper warning to him. → The law generally supports clampers operating on private land. But Mr Agar claims CCS's sign was not prominent enough to be a proper warning to him.*

For Estonian, we have agreed upon our own set of rules. Some examples of syntactic simplification for Estonian are:

- Removal of modal adverbs: *See on üpris võimatu. → See on võimatu.*
- An attribute can be transformed into a relative clause: *Vana puust kääksuv kapp. → Vana puust kapp, mis kääksub.*

- Replacing compound verbs with single verbs: *Ta annab koerale süüa. → Ta söödab koera.*
- Use of nominative case when possible: *Tema ema oli suguluses kirjaniku Charles Read'iga. → Tema ema oli kirjaniku Charles Read'i sugulane.*
- Avoiding the use of essive case: *Louis XIII valitses Prantsusmaa kuningana aastatel 1610-1643. → Louis XIII oli aastatel 1610-1643 Prantsusmaa kuningas.*
- Using finite verbs instead of vat-infinitive: *See taim kasvavat Aasias ja Aafrikas. → See taim kasvab Aasias ja Aafrikas.*

As we can see, Estonian simplification is more difficult to perform, due to the nature of Estonian language

1.1.3 Conceptual Clarification

In natural language processing, conceptual clarification involves providing examples or explanations for words that can have multiple meanings or terms that aren't universally understood [12]. For instance, in English, the phrase "a good car" can be interpreted differently by different individuals. Some might perceive a car as good if it's reliable, while others may consider speed, affordability, or comfort as defining qualities of a good car. To ensure a shared understanding of the term, further clarification is necessary.

Deciding whether further clarification is needed for a word can often be challenging. Typically, this determination is aided by tools such as wordnets and word sense disambiguation. By examining the words surrounding the term in question, we can gain insight into its intended sense. Consequently, additional information from the wordnet can be provided to enhance the overall comprehensibility of the sentence [12].

Taking it all into consideration, let's use this sentence as an example to demonstrate conceptual clarification:

Owls mostly hunt small mammals, insects, and reptiles. → Owls mostly hunt small mammals (such as mice, rats, and rabbits), insects (such as crickets) and reptiles (such as lizards).

We observe that the sentence remains unchanged lexically or syntactically; instead, it has been extended with explanations for certain words. For instance, "small mammals" is elaborated as "mice, rats, and rabbits." This additional information aids readers unfamiliar with owls in understanding which specific animals are being referred to. Without this clarification, readers might mistakenly interpret "small animals" as including creatures like cats, potentially leading to misconceptions about owls preying on domestic pets. Similarly, terms like "insects" and "reptiles" are broad and encompass a wide range of species. By providing conceptual clarification, readers gain a clearer understanding of which insects and reptiles are part of an owl's diet, mitigating potential confusion.

For Estonian, we follow the same principle, although it rarely occurs in use. Some examples include:

- *Põhjapoolusel elavad loomad on sageli kohastunud vees elama. → Põhjapoolusel elavad loomad (nagu jääkarud ja morsad) on sageli kohastunud vees elama.*
- *Endise F1 maailmameistri jaoks on Lexus LFA väga hea auto. → Endise F1 maailmameistri jaoks on Lexus LFA väga hea auto, sest see on kiire ja hea juhitavusega.*

- *Kalimantani saar on tuntud oma maavarade pärast. → Kalimantani saar on tuntud oma maavarade (nagu kivisüsi, kuld ja vääriskivid) pärast.*

Text simplification systems rarely incorporate conceptual clarification because there are too few instances of them in the training data. It is also difficult to perform algorithmically.

1.2 Previous Work for Estonian Text Simplification

Text simplification for Estonian is not a novel endeavor. Prior efforts have been undertaken, but none of them incorporate a neural network model, as presented in this thesis. Previous systems have relied solely on an algorithmic approach, which is founded on a predefined set of manually crafted rules as explained below.

In previous thesis by Martin Peedosk [5], an algorithmic approach was employed to replace difficult words with simpler variants. This system utilized the Estonian wordnet to identify more commonly used words with simpler meanings. Additionally, it measured the cosine distance between the vectors of these words to ensure that they were not significantly dissimilar. However, this approach was limited to lexical simplifications, as it did not alter the sentence structure. While it improved the readability of texts, it may not have been beneficial for individuals struggling with understanding lengthy sentences. One notable challenge was the system's occasional failure to select the correct word with similar meanings and higher frequency of use, resulting in unintended changes in meaning [5].

Another study by Stiivo Siider [6] focused on Estonian syntactic simplification. However, it also relied entirely on an algorithmic approach, drawing from existing simplification principles developed for English. While this approach effectively reduced sentence complexity, it did not incorporate any lexical simplification. Similar to the previously mentioned thesis, this approach encountered challenges in practice due to exceptions that could not be accommodated by an algorithmic approach [6].

1.3 Text Simplification Necessity

Text simplification serves various beneficial purposes. There is evidence suggesting that it can aid individuals with certain medical conditions. For instance, lexical simplification has been shown to benefit people with aphasia and dyslexia [1]. Furthermore, Paetzold and Specia conducted multiple tests involving speakers whose native language is not English. Their findings indicated that words used more frequently, as determined by frequency analysis of large national corpora, are much easier to comprehend for English learners. Therefore, text simplification has the potential to make information more accessible to individuals who are still in the process of acquiring proficiency in a language.

It has been observed that simplifying specific academic texts can ignite interest among individuals who may otherwise feel discouraged from pursuing them due to difficulties in comprehension. Without proper understanding, individuals may perceive certain fields as overly complex, deterring them from further exploration. However, by simplifying the text initially, it becomes more accessible, thereby potentially increasing interest in those fields. Consequently, this opens up more career options for individuals who may have initially been hesitant to pursue them due to the perceived complexity of the subject matter. Additionally, simplifying texts can enable existing specialists to enhance their skills more rapidly, as they require less cognitive processing power to grasp the content [13].

1.4 Text Simplification Systems

There are two distinct types of text simplification systems. One type relies on predefined rules, which can be taught to computers and subsequently applied automatically. The other type utilizes machine learning techniques and does not necessitate predefined rules; instead, it relies on extensive amounts of data for training purposes.

1.4.1 Rule Based Systems

Rule-based systems for text simplification rely on a set of rules defined by humans, many of which have been discussed previously. Teaching these rules to computers often involves the use of parsers. Parsers are algorithms that take text as input, break it down into smaller components known as tokens, and then make decisions based on predefined rules. The output of the parser is typically a parse tree (or syntactic tree in the context of language processing), which can then be further processed into text or any other desired format. Parsers specifically designed for text simplification have been explored, such as the Briscoe and Carroll parser [14].

Syntactic trees serve as the basis for performing transformations, which are operations that alter the structure of a text while retaining its original meaning. When utilizing parsers, transformations often involve less word deletion and more synonym replacement, reflecting the nature of parsing algorithms. Once transformation rules are established, the syntactic tree is examined to determine if it is in a suitable form for transformation. Subsequently, the syntactic tree undergoes modification through transformations that alter its composition. In some cases, transformations may be applied iteratively. Following this process, a transformed sentence is generated [3]. In summary, this process is entirely rule-based, with computers executing operations defined by humans.

1.4.2 Machine Learning Systems

All models discussed in this thesis utilize neural-network-based learning models. Neural models, also called machine learning models, function by employing algorithms inspired by biological brains to establish and strengthen connections between pieces of information. Through machine learning on datasets, these models are developed. They learn to make intricate decisions based on datasets used for training. Consequently, leveraging knowledge gained from the training set, these models can make decisions regarding information not encountered during training. This process is commonly referred to as prediction, or in the context of text simplification models, as translations, where complex sentences are essentially translated into simplified language [15].

With a sufficiently large and diverse dataset for training, it is theoretically possible to create models that outperform current algorithmic approaches. Neural network models have the capacity to learn from vast amounts of data and can capture language-specific nuances and exceptions that may be challenging to encode manually in rule-based systems. This ability to learn from data allows neural network models to adapt and make more informed decisions, potentially surpassing the performance of rule-based approaches, particularly in handling language-specific complexities and exceptions [16].

2 Datasets

The datasets used in this thesis consist of parallel corpora, each containing pairs of source and target sentences. The source sentences are in the original sentences, and the corresponding target sentences are simplified versions of the source sentences. These datasets are segmented into training, validation, and test sets for structured evaluation.

2.1 English Data

English data is the most readily available dataset for text simplification. Presently, systems predominantly utilize open-source data. However, such data is not always created by professionals and may occasionally exhibit lower quality. In this chapter, we will introduce some readily available datasets for use in English text simplification systems.

2.1.1 WikiSmall dataset

WikiSmall[1] is an openly available corpus containing sentences scraped from Wikipedia articles. It comprises 89,042 sentence pairs for training, 205 for validation, and 100 for testing. For the English language, many articles have been transformed into simpler versions. The purpose was to make Wikipedia articles accessible for English learners who might struggle with the, often scientific, language in regular articles. Thus, volunteers have been simplifying articles, and these simplified versions have been incorporated into the WikiSmall corpus. However, this approach presents a problem in the dataset. Essentially, anyone can attempt to simplify articles without any linguistic background. Consequently, many sentences in the corpus are not considered to be of good quality in terms of simplification. Let's examine an example from the dataset:

Similar appendages that are flat and do not come to a sharp point are called nails instead. Mammals A nail is similar to a claw but it is flatter and has a curved edge instead of a point.
→ Similar appendages which are flat and do not come to a sharp point are called nails instead.

We can observe from this example that the simplified sentence is shorter but lacks any significant simplification. For instance, the word "that" has been replaced with "which". While this substitution can be considered lexical simplification in theory, its correctness is debatable. Upon consulting the English word frequency list, we find that "that" ranks as the 9th most frequent word, whereas "which" ranks as the 232nd most frequent word. Hence, it's evident that "which" cannot be considered simpler than "that". Including such sentences in the training set could potentially confuse the model. Therefore, certain modifications should be made to the WikiSmall dataset to enhance its quality.

Nevertheless, WikiSmall has served as a standard dataset for evaluating the effectiveness of simplification models. This thesis will use this dataset only to train our machine learning English models.

2.1.2 Turk Corpus

The Turk corpus[2] consists of 2350 sentences. These sentences were sourced from the WikiSmall dataset, but this time, they underwent refinement by eight professional annotators from Amazon Mechanical Turk to meet the criteria of a good simplification. Their primary objective was to preserve the meaning of the sentences, and they refrained from including

[1] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2447
[2] https://paperswithcode.com/dataset/turkcorpus

any sentences that required splitting. Afterwards, the annotators' work was reviewed, and instances of poor quality were removed [17]. The resulting Turk corpus is widely employed in nearly every text simplification model. Due to its high quality, it serves various purposes. Some researchers have utilized it for validation or testing to achieve more accurate results.

2.1.3 Newsela Corpus

The Newsela corpus[3] comprises 1130 news articles, with each article having four simpler versions tailored for children in different grades. These simplified versions were created by language experts and editors at Newsela[4], a company specializing in developing reading materials for children. Consequently, we can trust the adequacy of their data. Several tests conducted by Wei Xu and others have assessed the quality of the Newsela corpus, revealing that Newsela offers significantly superior simplifications compared to WikiSmall [18]. However, one drawback of using the Newsela corpus is that accessing the data requires obtaining a license since Newsela does not permit public sharing of its content.

2.2 Estonian Data

For our Estonian experiments, we have compiled a dataset that integrates all of the aforementioned datasets along with artificially generated data. This section will elucidate the Estonian dataset utilized in this thesis.

2.2.1 Machine Translated WikiSmall

For larger training set, the entire WikiSmall dataset has been translated into Estonian using University of Tartu's Neurotõlge[5]. The entire translation took approximately 12 hours. Then, few bad instances, where translation failed, were removed or improved by hand. Together there are 88,893 sentence pairs as some sentences were lost in translation due to failure of translation. This is the biggest set in our corpus but it lacks in quality.

2.2.2 Machine Translated Turk Corpus

In addition the entire Turk corpus has been translated into Estonian. But unlike the WikiSmall dataset, the Estonian Turk corpus has been reviewed by our annotators and corrected by hand. Together there are 1912 Estonian sentence pairs that are used in the training set.

2.2.3 WikiLarge Dataset

WikiLarge[6] is a dataset similar to the WikiSmall dataset, only that it has much more sentence pairs. This isn't necessarily good, because it means that there are also more poor simplifications and pairs without any value. For our Estonian dataset it was entirely translated to Estonian and our annotators have picked some good-quality sentences that have been hand-corrected. Together there are 1414 sentence pairs from WikiLarge that are used for training.

2.2.4 AI Generated Data

Newsela and Turk corpuses alone do not provide sufficient data to train a highly accurate model. While WikiSmall contains ample data for training a satisfactory model, hand-evalu-

[3] https://newsela.com/data
[4] https://newsela.com/
[5] https://neurotolge.ee/
[6] https://paperswithcode.com/dataset/wikilarge

ating the dataset is extremely time-consuming. Therefore, to obtain quality data, a portion of it was artificially generated with the assistance of OpenAI's[7] GPT-4[8].

A prompt was engineered to take an Estonian article as input and process its sentences to produce simplifications. The initial batch of sentences was human-evaluated by our annotators, who agreed that they can be used as training data for our Estonian model. Some sentences are manually reviewed to verify it's precision.

Together we got 31,899 sentence pairs from 2000 Estonian Wikipedia[9] articles that have been simplified using GPT-4. This makes the biggest composition of our own dataset.

2.2.5 Total Composition

In total, there are 124,188 sentence pairs (Figure 1) that are used in Estonian dataset. 100 sentences are randomly removed from the set without the translated WikiSmall for most accurate evaluation.

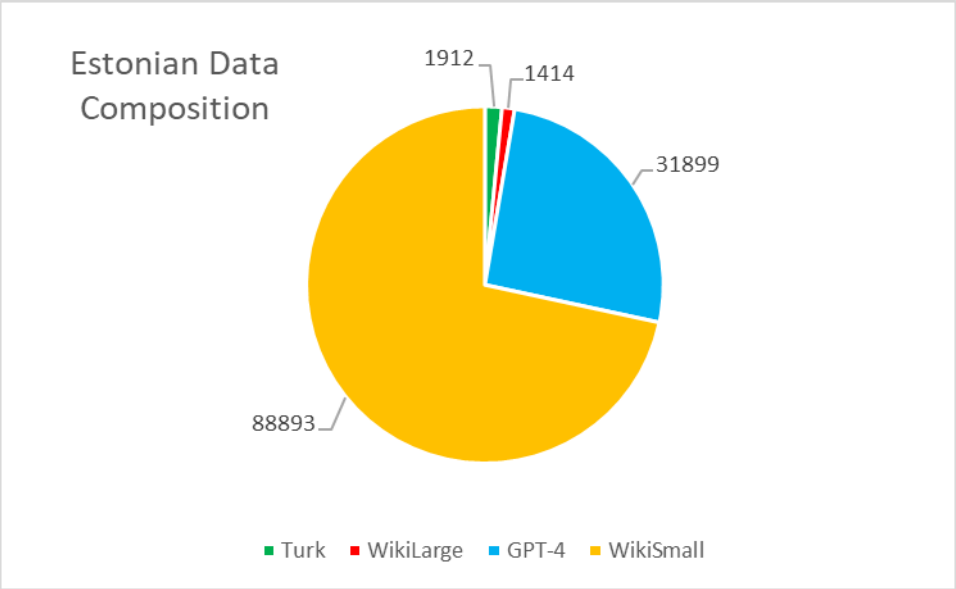

Figure 1: *Graph of Estonian Data Composition*

[7] https://openai.com/
[8] https://openai.com/index/gpt-4/
[9] https://et.wikipedia.org/wiki/Vikipeedia

3 Evaluation

The simplicity of text can be evaluated through various metrics. However, the best results are often obtained through human evaluation, as humans possess the greatest understanding of language simplicity thus far [3]. This thesis will discuss the three most commonly used automatically calculated measures: SARI, BLEU, and FKGL scores.

3.1 SARI score

The SARI (System output Against References and against the Input sentence) score is designed for automatic evaluation, specifically for text simplification. It was initially introduced by Wei Xu in the context of automatically evaluating text simplicity [17].

As described in the paper by Wei Xu and others, the SARI score favours outputs that contain words not present in the input sentence but are found in the reference sentences, which are predominantly evaluated by humans and considered correct. It does not inherently prioritize the removal of words, as doing so could potentially compromise the readability of the sentence. However, it may reward such removal if the output is ultimately more readable. A higher SARI score indicates a better simplification [17].

The calculation of the SARI score involves creating token n-grams from the output sentences and comparing them with the reference sentences. Token n-grams are sequences of n tokens taken from a set of tokens. For example, consider the sentence "sun is shining." After tokenization, the set of tokens would be ["sun", "is", "shining"].

- Unigrams consist of all individual tokens, so there are three unigrams: ["sun"], ["is"], and ["shining"].
- Bigrams are sets of two neighboring tokens, resulting in three bigrams: ["sun", "is"] and ["is", "shining"].
- Trigrams are sets of three neighboring tokens. In this short example, there is only one trigram: ["sun", "is", "shining"].

These n-grams are then compared between the output and reference sentences to compute the SARI score.

Before we can calculate SARI, we must first define the n-gram precision (1) and recall (2) functions for addition operator:

$$p_{add}(n) = \frac{\sum_{g \in O} min\left(\#_g(O \cap \bar{I}), \#_g(R)\right)}{\sum_{g \in O} \#_g(O \cap \bar{I})} \qquad (1)$$

$$r_{add}(n) = \frac{\sum_{g \in O} min\left(\#_g(O \cap \bar{I}), \#_g(R)\right)}{\sum_{g \in O} \#_g(R \cap \bar{I})} \qquad (2)$$

Where O is the simplified text, I is the input text, R is the reference sentence set, p is n-gram precision, r is n-gram recall and g is a specific n-gram itself. # is a binary indicator of n-gram occurrence in a given set. It is defined as follows (3):

$$\#_g(O \cap \bar{I}) = max(\#_g(O) - \#_g(I), 0) \qquad (3)$$

$$\#_g(R \cap \bar{I}) = max(\#_g(R) - \#_g(I), 0)$$

We also need to define n-gram precision (4) and recall (5) for keep operator:

$$p_{keep}(n) = \frac{\sum_{g \in I} min\left(\#_g(I \cap O), \#_g(I \cap R')\right)}{\sum_{g \in I} \#_g(I \cap O)} \quad (4)$$

$$r_{keep}(n) = \frac{\sum_{g \in I} min\left(\#_g(I \cap O), \#_g(I \cap R')\right)}{\sum_{g \in I} \#_g(I \cap R')} \quad (5)$$

Here R' is the n-gram occurrence out of total reference count and the binary indicator for these sets are defined as (6):

$$\#_g(I \cap O) = min\left(\#_g(I), \#_g(O)\right) \quad (6)$$

$$\#_g(I \cap R') = min\left(\#_g(I), \#_g(R)/r\right)$$

Here r is the number of references. For deletion, only precision function is defined because too much deletion has a bigger negative effect to readability as keeping. It is defined as (7):

$$p_{del}(n) = \frac{\sum_{g \in I} min\left(\#_g\left(I \cap \overline{O}\right), \#_g\left(I \cap \overline{R'}\right)\right)}{\sum_{g \in I} \#_g\left(I \cap \overline{O}\right)} \quad (7)$$

Here the binary indicator is defined as (8):

$$\#_g\left(I \cap \overline{O}\right) = max\left(\#_g(I) - \#_g(O), 0\right) \quad (8)$$

$$\#_g\left(I \cap \overline{R'}\right) = max\left(\#_g(I) - \#_g(R)/r, 0\right)$$

Together, the SARI score is defined as (9):

$$SARI = \frac{F_{add} + F_{keep} + P_{del}}{3} \quad (9)$$

$$P_{operation} = \frac{1}{k} \sum_{n=[1,\ldots,k]} p_{operation}(n)$$

$$R_{operation} = \frac{1}{k} \sum_{n=[1,\ldots,k]} r_{operation}(n)$$

$$F_{operation} = \frac{2 \cdot P_{operation} \cdot R_{operation}}{P_{operation} + R_{operation}}$$

$$operation \in [del, keep, add]$$

And k is the highest n-gram order [17]. We can immediately see that this score is difficult to calculate by hand. Let's use it in the following example:

- Input: "The car is currently accelerating."
- Reference1: "The car is currently going forward."
- Reference2: "The car is now accelerating."

- Reference3: “Car is now going forward.”
- Output1: “Car drive now.”
- Output2: “The car is now speeding.”
- Output3: “The car is currently speeding.”

Let’s calculate SARI score for all of the outputs. We get the following results (calculated with python easse[10] library):

- Output1: SARI=34.282, it is the worst result because there are many words that are not included in reference sentences.
- Output2: SARI=55.562, gives the best result since “now” is in 2 out of 3 references.
- Output3: SARI=47.044, gives worse result than Output2 because currently is in only one reference sentence.

SARI is taken as the most precise measurement for text simplification [19].

3.2 BLEU Score

BLEU (BiLingual Evaluation Understudy) was first introduced to measure the quality of machine translation. Since text simplification can be considered similar to machine translation when we take the original language as the source and the simplified language as the target, BLEU can be applied in this context. Similar to SARI, BLEU compares generated outputs to reference sentences; however, BLEU does not consider how consistent the output text is. It calculates the score solely by checking if words exist in the output n-grams and if they are also present in the references. More specifically, it measures how many generated n-grams of tokens match the n-grams of the reference sentences. Apart from SARI, BLEU assigns a lower score to outputs that are too short because in translation, shorter sentences are considered to be of lower quality [20].

Let’s see the formula for BLEU. First, let’s define the precision score (10):

$$p_n = \frac{\sum_{g_n \in I} Count_{clip}(g_n)}{\sum_{g_n' \in I'} Count(g_n')} \qquad (10)$$

$$Count_{clip}(n) = min\big(Count(n), Max\_Ref\_Count(n)\big)$$

Where I is the input token set and g is n-gram of a specific order. The clip count function means that each n-gram is clipped to the maximum number of n-grams in the reference set. Basically meaning that the count of words cannot be higher than that inside the references.

We also need to define the Brevity penalty (11), which penalizes translations that are too short compared to the given references:

$$BP = \begin{cases} 1 & , \ if\ i > r \\ e^{(1-r/i)} & , \ if\ \leq r \end{cases} \qquad (11)$$

Here, i is the length of input and r is the adjusted length of a word of the same type inside the reference set.

Now we can define the BLEU metric as (12):

[10] https://github.com/feralvam/easse

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{k} w_n log p_n\right) \qquad (12)$$

Where k is the highest n-gram order and w is the geometric mean of the n-gram precisions [20].

Let's look at BLEU scores with the same set of sentences we used before:

- Input: *"The car is currently accelerating."*
- Reference1: *"The car is currently going forward."*
- Reference2: *"The car is now accelerating."*
- Reference3: *"Car is now going forward."*
- Output1: *"Car drive now."*
- Output2: *"The car is now speeding."*
- Output3: *"The car is currently speeding."*

For these outputs, we get the following results (calculated with python easse library):

- Output1: BLEU=12.751
- Output2: BLEU=53.728
- Output3: BLEU=53.728

From these results, it becomes clear why BLEU can sometimes be seen as a poor metric for measuring simplicity. We can observe that the scores are the same for the second and third outputs. This is because BLEU does not consider which word is more frequent within the references; it calculates based entirely on existence.

Recently, BLEU has been argued to be an inadequate measure of text simplification and therefore should not be used [19]. In this thesis, it is still used for variation and extra informative evaluation.

The disadvantages compared to SARI are:

- Balanced evaluation: SARI offers a more balanced evaluation. It penalizes over-simplification and under-simplification. It takes into account that the overall meaning should not change. While BLEU penalizes removal of words, it doesn't take into account the whole compositional meaning.
- Focus on modification: SARI specifically focuses on the modifications made to the input text. This means that it evaluates if the changes are appropriate or not.
- Independence from Reference: SARI doesn't depend too much on reference set used for evaluating. Therefore, it is less biased on the reference style. This makes SARI evaluation especially valuable.

3.3 FKGL Score

FKGL (Flesch-Kincaid Grade Level) is designed to measure readability. Originally, its purpose was to estimate the educational grade level required to comprehend the measured text. For instance, a FKGL score of 6 would indicate that Grade 6 students would be capable of understanding the text, based on the standards of the education system in the USA. Therefore, the lower the FKGL score, the more readable the measured text is considered to

be. Occasionally, for short texts, the FKGL score can be negative. In such cases, it is typically rounded to 0, as it would be unconventional to associate negative scores with grade levels. Consequently, 0 can be interpreted as preschool level. Initially, the FKGL score was intended for students to assess their text comprehension skills [21]. It is calculated relatively simplistically compared to BLEU and SARI scores.

The formula to calculate the FKGL score is (13):

$$FKGL = 0.39\frac{N_{words}}{N_{sentences}} + 11.8\frac{N_{syllables}}{N_{words}} - 15.59 \qquad (13)$$

Before the introduction of SARI, FKGL was a commonly used metric to evaluate machine translation. Several years after its introduction, the formula underwent updates, resulting in its current form. However, it has recently gained criticism as a flawed measurement [21].

It's evident from the formula that FKGL rewards texts with a higher number of sentences and words. Simultaneously, it favors texts with fewer words per sentence and fewer syllables per word. However, the score does not assess the readability or grammatical difficulty of the text, making it possible to achieve a high score even if the sentences are challenging to understand but contain fewer words each.

Let's use the same set again to demonstrate FKGL scoring. This time we only need the output sentences:

- Output1: *"Car drive now."*
- Output2: *"The car is now speeding."*
- Output3: *"The car is currently speeding."*

Now let's see the FKGL scores (calculated by hand, using the formula).

- Output1: FKGL=-2.62=0, since negative scores are rounded to 0
- Output2: FKGL=0.52
- Output3: FKGL=5.24

We can see that FKGL score doesn't consider reference sentences at all. Thus it only tells simplicity by the sentence itself, even when sentences might not be grammatically correct.

In this thesis, we will continue to utilize the FKGL score as a measurement. When used in conjunction with other metrics, it can provide valuable insights into text simplicity by analyzing the average number of syllables and words in output sentences. For instance, when comparing two sets of outputs with equal SARI scores, the set with a superior FKGL score should theoretically be easier to comprehend, as it likely contains fewer words and syllables.

4 Experiments

In order to evaluate and compare our models, English models had to be trained. Instructions provided by accompanied research papers with state of the art models have been used to retrain the models. In this thesis, we will cover 3 models for English that offer very good results in their accompanied papers. Estonian models have been reengineered to fit our purpose. All of the models mentioned in this thesis have been trained using Nvidia V100 16 gigabytes GPU[11] backend with 16 gigabytes of CPU memory. All of the models that are hereby mentioned, together with the training datasets are uploaded and found on our GitHub[12]. They are organized by the following model names.

4.1 English Models

All of the models are sourced from the internet. Usually these models have caught our eye by offering good results accompanied with the feasibility to train such a model. While we used more models in the testing phase, only 3 of them will be mentioned in this thesis. They have been originally trained on different datasets. To compare them with each other more precisely, they are all trained on the same English WikiSmall dataset. It is the same experiment also done in the work of Xingxing Zhang and Lapata [22].

4.1.1 DRESS Algorithm

Since the experiments are essentially the same as in the research paper accompanied with this algorithm, we will use it as a standard for comparison. The model is available publicly and can be downloaded from GitHub[13].

The model is a reinforcement learning neural network. It means that the model periodically updates itself to maximize simplification quality. It also has it's own pretrained encoder-decoder model.

Let's mark the input sentence as X and the output sentence as Y. The encoder turns the input sentence into a set of hidden states, which are information that the computer uses in making connections and can't be seen or understood by humans. Let the hidden states set be $[h_1^S, h_2^S, \ldots, h_n^S]$. The decoder takes this set as input and generates a token y_i from the ouput sentence. It continues the generation, taking into account all of the previous generated tokens, until all of the tokens are generated. It takes context information, which is previously put into a vector with specified length, called the context vector c_t. It can be calculated as follows (14):

$$c_t = \sum_{i=1}^{|X|} \alpha_{ti} h_i^S \qquad \alpha_{ti} = \frac{\exp(h_t^T \cdot h_t^S)}{\sum_i \exp(h_t^T \cdot h_t^S)} \qquad (14)$$

The h_t^S represents hidden state of all the tokens already generated. The context vector is used for generating next token. Therefore the generalized formula for generation is (15):

$$P(X|Y) = \prod_{t=1}^{|Y|} P(y_t|y_{1:t-1}, X) \qquad (15)$$

[11] https://www.nvidia.com/en-us/data-center/v100/
[12] https://github.com/mahlane-arbuus/neural_estonian_simplification
[13] https://github.com/XingxingZhang/dress

$$P(y_{t+1}|y_{1:t}, X) = softmax\big(g(h_t^T, c_t)\big)$$

In the last equation, the g function represents a parametrized one-hidden-layer neural network. The generation stops when it produces an ending token. Then the generated ouput is given a reward. The reward is calculated in a way that maximizes simplicity, relevance and fluency of the output sentence. It is calculated by mainly by using SARI to calculate simplicity, cosine similarity of the input and output vectors for relevance, and BLEU for fluency. They are weighted to ensure that one metric doesn't shadow the others.

Then the reinforcement algorithm is finally used for training. The training loss is taken by calculating the reward and then making it negative. Then at first, the model is trained in a way that first predicts the end of the output sentence. This means that the beginning is given and the algorithm only generates simplifications so that the beginning of the outcome is already known. Then after every two epochs (cycles over the whole dataset), the model is allowed to predict even more of the output sentence. This process goes on until finally the model learns to predict on the whole output sentence.

The algorithm uses 2 layers, 256 hidden units and a dropout of 0.2. It is trained for approximately 8 epochs with Adam optimizer and a learning rate of 0.001 [22].

4.1.2 OpenNMT Algorithm

The original algorithm by Sergiu Nisioi[14] used a framework called OpenNMT[15] (Neural Machine Translation) along with Lua[16] training scripts and old version of Torch[17]. Since current systems no longer easily support this setting, we updated and trained the model with a newer version called OpenNMT-py[18]. This version doesn't require any Lua support and can be run entirely with Python[19] and it's packages. Most important of them is PyTorch[20], which is the framework used for training. It is also found that using this newer version doesn't result in worse evaluation scores by a GitHub user[21]. Although it's credibility can be argued since there is no research paper attached. The package itself is used mainly for machine translation.

The algorithm uses 2 layers, 500 hidden units and 0.3 dropout. It is trained for 15 epochs with SGD optimizer and a learning rate of 0.1. The vocabulary is set to 50,000. (Sergiu Nisioi, 2017). The vocabulary is a file that counts words from a set of data and stores them followingly. Two vocabulary files are created for input and output sentences. This is how the model learns to make connection between words. It uses vocabulary files as reference and stores the connections inside the neural network.

First the context vector is calculated. It is calculated similarly to the previous algorithm. It is basically a weighted average of hidden states h_t and alignment weights α_t (used for determining which tokens are the most relevant for generating the output) and can be calculated as follows (16):

[14] https://github.com/senisioi/NeuralTextSimplification/tree/master
[15] https://opennmt.net/
[16] https://www.lua.org/
[17] http://torch.ch/
[18] https://opennmt.net/OpenNMT-py/
[19] https://www.python.org/
[20] https://pytorch.org/
[21] https://github.com/jind11/NeuralTextSimplification-Pytorch

$$c_t = \sum_{i=1}^{|X|} \alpha_{ti} h_i \qquad (16)$$

Here |X| is the length of the input token set. The next hidden states and alignment weights can be calculated using the following formula (17):

$$h_{t+1} = tanh\big(W(c_t; h_t)\big) \qquad (17)$$

$$\alpha_t(s) = \frac{exp\big(h_t^T W(h_{s+1})\big)}{\sum_{s'} exp\big(h_t^T W(h_{s'+1})\big)}$$

Where W is the weight matrix. The model trains the matrix itself and chooses it's dimension based on the sizes hidden state and context vector. It basically predicts the next hidden state by taking into account the current output and the context vector. And s is the current time step in the training process. Time step marks the current token in the input sequence. Here s' means the length of the input token set [9].

The training process stops when a number (15 in this case) of epochs has been reached. It is then measured which model has the best scores, to ensure that it hasn't been overfitted (made too dependent on the training set). The whole training process was finished in 1 hour and 33 minutes.

4.1.3 Fine-tuned T5

The last model for English data features fine-tuning of the T5 model. T5 is a pre-trained large language model that has gained attention recently for being open-source and relatively competitive with other current state of the art models. It can be downloaded from HuggingFace[22] without a license and used freely. In this algorithm, T5 is being fine-tuned for text simplification.

Fine-tuning is a process, where an already trained model is fitted with new data for it to perform a new task or perform an already learned task better. This is usually done with very large models. Fine-tuning is very effective because it saves time and offers a big set of data. When fine-tuning for a specific task, the model also learns to make decisions based on the dataset it was trained with before [23].

The algorithm has it's own tokenizer and tokenizes sentences prior to training. It uses default values that were used to train the T5 model. Only set parameters are that it is trained for 5 epochs with Adam optimizer [24]. The whole training took around 10 hours.

There is also a preprocessor included. It basically tokenizes the text and adds control tokens. Control tokens are there to even further improve generating outputs. It takes them into account in the training process but excludes them when generating outputs. For example, some of them are character length ratio between input and output and Levenshtein distance between input and output. Together there are 5 such tokens [24].

After the preprocessing, the T5 is fed all of the data with "simplify" instruction to mark that it is a simplification process [24].

[22] https://huggingface.co/docs/transformers/en/model_doc/t5

4.2 Estonian Experiments

For Estonian three models were trained. They were all trained using the OpenNMT-py library due to how fast it performed on English dataset and how easy it was to execute. The parameters were left the same as when trained with English data. Only difference is that the vocabulary size was upped to 60,000 and vocabulary sharing was enabled. Vocabulary sharing allows to use a shared vocabulary file. This way, tokens that appear in input set but not in the output set, will get handled better. Throughout the training, 12 checkpoints were saved. Later these checkpoints were all evaluated to ensure that the models were not over-fitted and the best performing model was chosen. The three models trained differed only by their training set. The three models are as follows:

- Quality: This model is identified as the quality model. The word "quality" is used to emphasize that it was trained using our checked data by annotators and our AI-generated data to ensure that the model was fed with the best possible data.
- Translated: This model is identified as the translated model. It is named so because it was trained using only the translated data from WikiSmall. Since there are more words in the training set, the vocabulary size was set to 80,000.
- Combined: This time both of the above sets were used for training. This includes our whole corpus. This time, the vocabulary size was set to 120,000 in accordance to our available resources, since a bigger vocabulary size requires more memory.

In total, it took about 5.5 hours to train all of the models explained above.

5 Results

Our models were evaluated automatically using a Python package named easse. The package already included built in implementation of SARI, BLEU and FKGL scores.

5.1 English Results

The three models, that were described earlier, compared relatively differently to each other. The DRESS model was used as a standard for comparison and it got the following scores: SARI of 13.61, BLEU of 47.93 and FKGL of 11.35. The OpenNMT-py trained very fast with good results. It finished training in only about 1.5 hours and produced the following scores: SARI of 35.82, BLEU of 44.61 and FKGL of 9.97. Last up was fine-tuned T5 with following scores: SARI of 41.21, BLEU of 30.88 and FKGL of 7.23. The results are available to see in table (Table 1) and diagram (Figure 2) formats.

Table 1: *Evaluation of English Models*

Model	BLEU	SARI	FKGL
DRESS	**47.93**	13.61	11.35
OpenNMT	44.61	35.82	9.97
T5	30.88	**41.21**	**7.23**

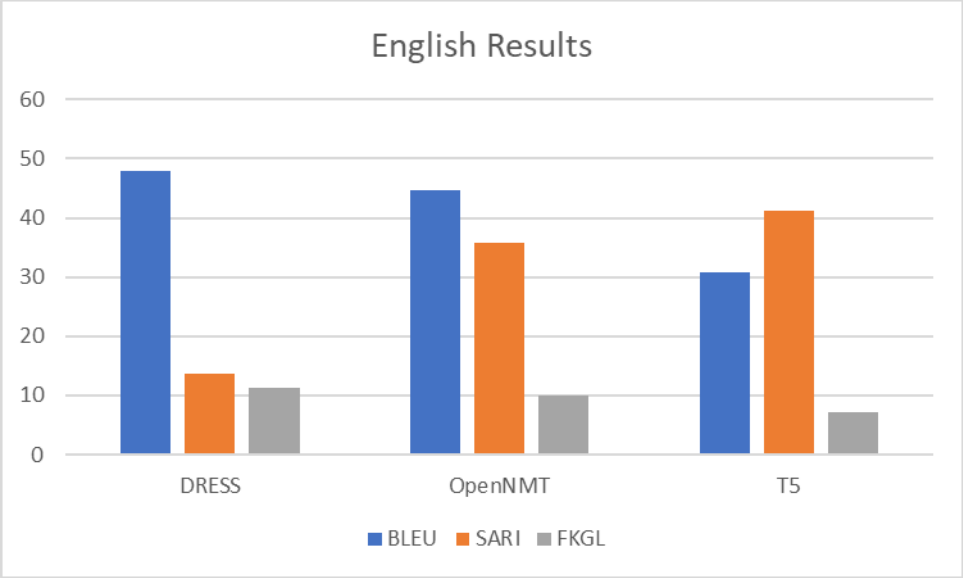

Figure 2: *Diagram of English Results*

T5 got the best results in all of the metrics beside BLEU. The DRESS algorithm got the worst SARI and FKGL scores but the best BLEU score. Upon closer inspection, it was seen that it was mainly copying the input with few instances of lexical simplification. This is why the SARI score is so low. FKGL score hints that there was not much sentence splitting, as the amount of sentences and words was high. OpenNMT model got a better BLEU score than T5. This could indicate that the OpenNMT simplifications have sentences that are more close to the references. This doesn't mean that T5 is worse. Since it's SARI score is so high, it means that it probably used more sentence splitting that wasn't used so much in the reference sentences. Some outputs can be seen at the appendix section (33).

Overall, it seems that fine-tuning pretrained large language models offers far better overall performance but OpenNMT-py can also be considered competitive taking into consideration of how easy and fast it is to train.

5.2 Estonian Results

This time, the results were a little surprising. Overall, the "quality" model got the following scores: BLEU of 14.5, SARI of 47.59 and FKGL of 6.12. For some reason, it trained the longest amount of time of 1 hour and 47 minutes. The "translated" model trained the fastest with 1 hour and 30 minutes and got the following scores: BLEU of 19.12, SARI of 43.72 and FKGL of 6.22. The "combined" model trained slightly slower with 1 hour and 35 minutes and got the following scores: BLEU of 23.86, SARI of 48.85 and FKGL of 7.00. The results can be viewed in a table (Table 2) and diagram (Figure 3) formats.

Table 2: *Evaluation of Estonian Models*

Model	BLEU	SARI	FKGL
quality	14.50	47.59	**6.12**
translated	19.12	43.72	6.22
combined	**23.89**	**48.85**	7.00

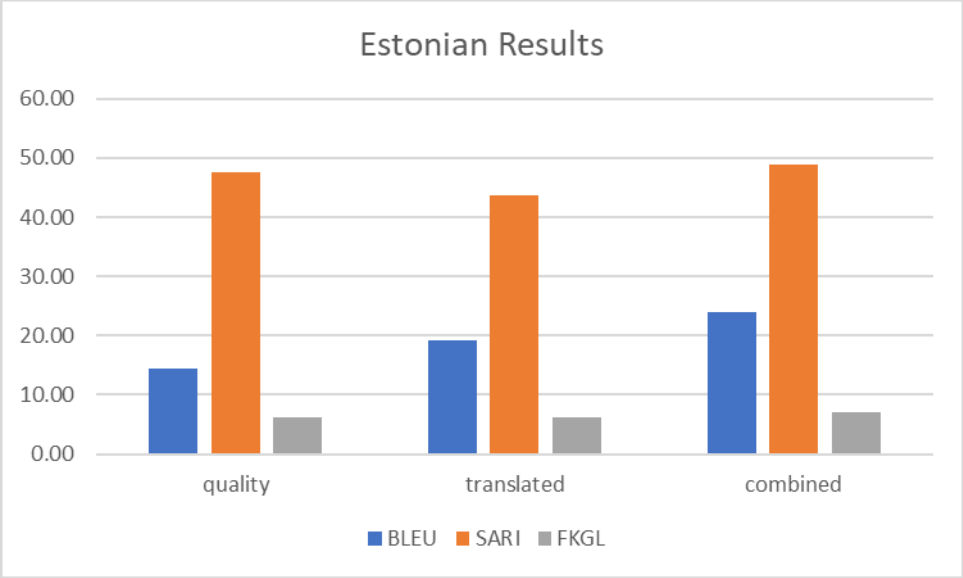

Figure 3: *Diagram of Estonian Results*

We can see that the "combined" model offered the best overall results. The "quality" model was not far behind. It got the worst BLEU score of all of them. Further inspection revealed that it is because the model was trained with data that involved a lot of data splitting. Sometimes it splits sentences even when it wasn't supposed to, creating bad BLEU scores, because BLEU penalizes too short sentences. This is also the reason why it's FKGL score was so low.

As expected, the worst performance was with the "translated" model. Upon inspection, it was revealed that in many cases, it only took the first part of a sentence, usually split where there was a conjunction (for example "et" or "aga") and kept only the first part. Such a sentence loses a lot of it's original meaning. That is also the reason why SARI score was the lowest.

Upon further inspection of the "combined" model, it showed that it produced the best quality simplifications. It knew how to split sentences very well. It sometimes failed to choose a correct beginning after the splitting. For example, let's look at these examples in English and Estonian.

The man, who is very strong, lifted a heavy stone. → The man is strong. He lifted a heavy stone.

Mees, kes on väga tugev, tõstis üles raske kivi. → Mees on väga tugev. Ta tõstis üles raske kivi.

Here we can see how the correct way of splitting works. In our case, sometimes the word "it" ("see") was used instead of "he" ("ta"). While we can still understand the meaning, it becomes grammatically incorrect. This is due to insufficient amount of training data for the model to make such connections. Some outputs can also be seen in the appendix section (34)

In conclusion, there were instances where the best model failed to produce good quality simplifications. This was mainly because it still had too few training data and couldn't learn how to perform simplifications on certain sentence patterns. But it can still be considered as success in terms of overall results.

5.3 Comparison

Overall we can consider the experiments in this thesis a success. This is the first implementation of a machine learning approach to Estonian text simplification, and our models perform comparably well relative to their existing state-of-the-art English counterparts.

There was a difference in training data. The WikiSmall set that was used to train the English models was smaller than our dataset used in the training process for our best model. There were 34,841 sentence pairs more in our training set. But even if we don't take the number into consideration, our "quality" model still had better results than the "translated" variant. Knowing this, we can say that adding even a small high-quality dataset to the open-source datasets available can drastically improve the results.

One interesting measure observed was that the "translated" model turned out better than the original OpenNMT model (Table 3). Another test was performed on the same test set that was used to evaluate the English models. It was translated into Estonian. This time, the "translated" model got the following scores: BLEU of 15.55, SARI of 38.38 and FKGL of 7.76. For comparison, the same metrics for the original OpenNMT model were: BLEU of 44.61, SARI of 35.82 and FKGL of 9.97.

Table 3: *Comparison Between the same English and Estonian models*

Model	BLEU	SARI	FKGL
OpenNMT	**44.61**	35.82	9.97
translated	15.55	**38.38**	**7.76**

This result was very surprising as we expected them to be lower. The BLEU score was the only metric that was worse, indicating that the simplifications differed much from the references. But both the SARI and FKGL scores were better, indicating a better simplification quality and more sentence-splitting or deletion operations. It was confirmed upon closer inspection. Some outputs can be seen in the appendix senction (35)

This could be due to the fact that the translation process actually improved the dataset. Sometimes when we translate a sentence that isn't grammatically correct, the neural network behind the model actually makes sense of the whole sentence and translates it into something that could become grammatically correct. This could be proven further by translating the English set into another language and then again into English to evaluate metrics based on the new set.

5.4 Further Work

Further on, it would benefit if simplifications will get a human evaluation. Overall it can be a good base to continue developing an Estonian text simplification model using machine learning. There is definitely lack of data for a good model and more data should be produced. Fine-tuning a large pre-trained model like Llammas[23] would also be beneficial.

[23] https://huggingface.co/tartuNLP/Llammas-base

Conclusion

In this thesis, our aim was to explore Estonian text simplification by machine learning.

Initially, we gathered and compiled data from diverse sources. The bulk of our data originated from an open-source dataset called WikiSmall, which underwent machine translation into Estonian. However, this dataset lacked in quality, prompting us to supplement it with additional data from various other publicly available sources. Subsequently, we translated and meticulously hand-checked these additional sources. Finally, we augmented our dataset by artificially generating high-quality data with the assistance of GPT-4.

Subsequently, we identified pre-existing algorithms designed for English text simplification, which we implemented and assessed. Following an analysis of the outcomes, we selected a model to serve as the foundation for developing our Estonian model. Making necessary adjustments to adapt the algorithm for Estonian data, we proceeded to train the models and conducted evaluations similar to those performed with their English counterparts.

Based on our findings, we can confidently assert that the thesis was a success. We effectively trained a machine learning model that rivals existing English variants. While not flawless, our model adeptly simplifies text through diverse techniques. Furthermore, we observed significant enhancements in model performance when utilizing high-quality data. Remarkably, our best-performing model emerged from training with a combination of our own high-quality data and machine-translated data. This improvement may be attributed to enhancements in the dataset facilitated by the translation process.

Overall, this thesis serves as a significant milestone in the pursuit of developing an effective machine learning model for Estonian text simplification. Drawing from our findings in English text simplification, future efforts could focus on fine-tuning a large pre-trained model, which is anticipated to yield even superior results using the dataset provided in this thesis.

Acknowledgements

This thesis was made with the support of EKTB55- Text Simplification for Estonian lead by Eduard Barbu. Special thanks to the annotators Piret Kivi, Mihkel Rünkla, Jaan Vaab and Meery-Ly Muru for generating and putting together good quality data. I want to thank Heili Orav from the NLP group of the University of Tartu for valuable insights. Special acknowledgement must go to Eduard Barbu, the supervisor of this thesis, whose guidance was instrumental throughout this journey.

List of References

[1] Specia L. & Paetzold G. H. A Survey on Lexical Simplification. Journal of Artificial Intelligence Research. 2017.

[2] Truica C.-O. Stan A.-I. & Apostol E.-S. SimpLex: a lexical text simplification architecture. Neural Computing and Applications. 2022.

[3] Siddharthan A. Syntactic Simplification and Text Cohesion. University of Cambridge Computer Laboratory. 2004.

[4] Ondov B. Attal K. & Demner-Fushman D. A survey of automated methods for biomedical text simplification. American Medical Informatics Association. 2022.

[5] Peedosk M. Eesti keele digitaalsete ressursside ja tehnoloogiate rakendamine teksti lihtsustamise programmis. University of Tartu Institute of Computer Science bachelor's thesis. 2017.

[6] Siider S. Süntaksianalüüsil põhinev teksti lihtsustaja. University of Tartu Institute of Computer Science bachelor's thesis. 2019.

[7] Allen D. Crossley S. A. & McNamara D. Text simplification and comprehensible input: A case for an intuitive approach. *SAGE.* 2012.

[8] Saggion H. & Grabar N. Evaluation of Automatic Text Simplification: Where are we now, where should we go from here. Avignon. 2022.

[9] Nisioi S. Stajner S. Ponzetto S. P. & Dinu L. P. Exploring Neural Text Simplification Models. Association for Computational Linguistics. Vancouver. 2017.

[10] Schmitt N. & Jiang X. The Percentage of Words Known in a Text and Reading Comprehension. The Modern Language Journal. 2011.

[11] Enç M. The Semantics of Specificity. Massachusetts Institute of Technology. 1991.

[12] Bringmann L. F. Elmer T. & Eronen M. I. Back to Basics: The Importance of Conceptual Clarification in Psychological Science. SageJournals. 2022.

[13] Rossetti A. & Waes L. V. It's not just a phase: Investigating text simplification in a second language from a process and product perspective. University of Antwerp. 2022.

[14] Briscoe T. & Carroll J. Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels. ACL/SIGPARSE. Prague. 1995.

[15] Keijsers N. Neural Networks. Encyclopedia of Movement Disorders. 2010.

[16] Zhou M. Duan N. Liu S. & Shum H.-Y. Progress in Neural NLP: Modeling, Learning, and Reasoning. Microsoft Research Asia. Beijing. 2020.

[17] Xu W. Napoles C. Pavlick E. Chen Q. & Callison-Burch C. Optimizing Statistical Machine Translation for Text Simplification. University of Pennsylvania. 2016.

[18] Xu W. Callison-Burch C. & Napoles C. Problems in Current Text Simplification Research. University of Pennsylvania. 2015.

[19] Sulem E. Abend O. & Rappoport A. BLEU is Not Suitable for the Evaluation of Text Simplification. The Hebrew University of Jerusalem. 2018.

[20] Papineni K. Roukos S. Ward T. & Zhu W.-J. BLEU: a Method for Automatic Evaluation of Machine Translation. IBM T. J. Watson Research Center, New York, 2002.

[21] Tanprasert T. & Kauchak D. Flesch-Kincaid is Not a Text Simplification Evaluation Metric. Pomona College. Claremont. 2021.

[22] Zhang X. & Lapata M. Sentence Simplification with Deep Reinforcement Learning. University of Edinburgh. 2017.

[23] Lalor J. Wu H. & YuImproving H. Machine Learning Ability with Fine-Tuning. ResearchGate. Amherst/Chestnut Hill. 2017.

[24] Sheang K. C. & Saggion H. Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer. Association for Computational Linguistics. Aberdeen, Scotland. 2021.

Appendix

I License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Sten Marcus Malva**,

1. 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 expiry of the term of copyright, my thesis
Exploring And Implementing Estonian Text Simplification Using Machine Learning,
supervised by Eduard Barbu

2. I 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 DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

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

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

Sten Marcus Malva
15/05/2024

II English Model Output Examples

Original: *Genetic engineering has expanded the genes available to breeders to utilize in creating desired germlines for new crops.*

Reference: *New plants were created with genetic engineering.*

- DRESS: *Genetic engineering has expanded the genes available to breeders to make creating germlines for new crops.*
- OpenNMT: *Genetic engineering has expanded the genes available to breeders to develop using new crops.*
- T5: *Genetic engineering has increased the genes available to breeders. This allows them to create germlines for new crops.*

Original: *Restoration of independence In 1991, the Soviet Union broke apart and Armenia re-established its independence.*

Reference: *Armenia received its independence from the Soviet Union in 1991.*

- DRESS: *In 1991, the Soviet Union broke apart and Armenia.*
- OpenNMT: *Restoration of independence In 1991, the Soviet Union broke apart and Armenia.*
- T5: *Restoration of independence In 1991, the Soviet Union broke up and Armenia became independent again.*

Original: *An electron microscope is a type of microscope that uses a particle beam of electrons to illuminate a specimen and create a highly-magnified image.*

Reference: *An electron microscope is a microscope that uses electrons.*

- DRESS: *An electron microscope is a type of microscope that uses a particle beam of electrons to illuminate a specimen.*
- OpenNMT: *An electron microscope is a type of microscope that uses a particle of electrons.*
- T5: *An electron microscope is a type of microscope that uses a beam of electrons to specimen and make an image that is very big.*

Original: *Much of the Enlightenment occurred in French intellectual circles, and major scientific breakthroughs were achieved by French scientists in the 18th century.*

Reference: *French scientists made big scientific discoveries in the 18th century.*

- DRESS: *Much of the Enlightenment happened in French political circles.*
- OpenNMT: *Much of the Enlightenment occurred in French serial circles, and major scientific breakthroughs were achieved by French scientists in the 18th century.*
- T5: *Much of the Enlightenment happened in French circles. Big scientific discoveries were made by French scientists in the 18th century.*

Original: *December is the twelfth and last month of the year in the Gregorian Calendar and one of seven Gregorian months with the length of 31 days.*

Reference: *December is the twelfth and last month of the year, with 31 days.*

- DRESS: *December is the twelfth and last month of the year.*
- OpenNMT: *December is the twelfth and last month of the year in the Gregorian Calendar and one of seven Gregorian months with the length of 31 days.*
- T5: *December is the twelfth and last month of the year. It is one of the seven Gregorian months. It has 31 days in the year.*

III Estonian Model Output Examples

Original: *Esimest tüüpi Wolframi sündroom on põhjustatud mutatsioonidest WFS1 geenis, kuid teist tüüpi sündroom on põhjustatud mutatsioonidest CSID2 geenis.*

Reference: *Esimest tüüpi Wolframi sündroom tekib WFS1 geeni mutatsioonide tõttu. Teist tüüpi sündroom tekib CSID2 geeni mutatsioonide tõttu.*

- "quality": *Esimest tüüpi Wolframi rada on põhjustatud geeni mutatsioonid. Aga teise tüüpi ioonid on see, mis muutub teise tüüpi geeni.*
- "translated": *Esimest tüüpi Wolframi sündroom tekib WFS1 geeni mutatsioonide tõttu. Teist tüüpi sündroom tekib CSID2 geeni mutatsioonide tõttu.*
- "combined": *Esimest tüüpi Wolframi sündroom tekib WFS1 geeni mutatsioonide tõttu. Teist tüüpi sündroom tekib CSID2 geeni mutatsioonide tõttu.*

Original: *Rooma prominentne ooperipublik oli olnud alati uue teose suhtes kahtlev ja mõneti tõrjuv ega kiirustanud piletit ostma.*

Reference: *Rooma ooperipublik on uute teoste suhtes kahtlev.*

- "quality": *Rooma prominentne ooperipublik oli alati olnud uue teose suhtes. Nad ei olnud kahtlev, tõrjuv ega kiirustanud, mida ta vaatab.*
- "translated": *Rooma prominentne ooperipublik oli olnud alati uue teose suhtes kahtlev.*
- "combined": *Rooma ooperipublik oli alati uue teose suhtes kahtlev.*

Original: *Järgmise nelja aasta jooksul andis ta välja veel kaks albumit, millest viimane, 1993. aastal välja antud "Black Reign" sai kuldplaadi staatuse.*

Reference: *Ta andis järgmise nelja aasta jooksul välja kaks albumit. Viimane album (Black Reign) anti välja 1993. aastal ja see sai kuldplaadi staatuse.*

- "quality": *Ta andis järgmise nelja aasta jooksul välja kaks albumit. Viimane nimi oli "Black Reign", mis ilmus 1993. aastal.*
- "translated": *Järgmise nelja aasta jooksul andis ta välja veel kaks albumit, millest viimane, 1993. aastal välja antud "Black Reign" sai kuldplaadi staatuse.*
- "combined": *Järgmise nelja aasta jooksul andis ta välja veel kaks albumit, millest viimane, 1993. aastal välja antud "Black Reign" sai kuldplaadi staatuse.*

Original: *Seejärel, 21. detsembril 1994, paiskas vulkaan gaasi ja tuhka, mida valdavad tuuled kandsid kuni 25 km (16 miili) kaugusele.*

Reference: *21. detsembril 1994, tuli vulkaanist gaasi ja tuhka, mille tuul kandis kuni 25 km kaugusele.*

- "quality": *21. detsembril 1994, pannes värvitud gaasi ja tuhka. Nad sõidavad seal umbes 25 km ehk 16 miili (16 miili).*
- "translated": *Seejärel, 21. detsembril 1994, saatis vulkaan välja gaasi ja tuhka, mille tuul kandis kuni 25 km (16 miili) kaugusele.*
- "combined": *Seejärel, 21. detsembril 1994, saatis vulkaan gaasi ja tuhka, mida tuul kandis kuni 25 km (16 miili) kaugusele.*

IV Estonian and English Model Output Comparison

Original ENG: *The eruption causes deaths of livestock when they eat contaminated grass and also widespread crop failure.*

Reference ENG: *Eruption causes deaths of livestock when they eat contaminated grass and also widespread crop failure.*

Original EST: *Purse põhjustab saastunud rohu söömisel kariloomade surma ja ka laialdast viljakatkestust.*

Reference EST: *Purse põhjustab saastunud rohu söömisel kariloomade surma ja ka laialdast viljakatkestust.*

- Output ENG: *The eruption causes deaths of livestock when they eat grass and also small crop failure.*
- Output EST: *Purse põhjustab kariloomade surma ja viljakatkestust.*

Original ENG: *Earth travels a tremendous distance in its orbit around the sun, at a speed of around 30 km/s or over 108000 km per hour.*

Reference ENG: *The experiment The Earth travels very quickly (100,000 km per hour) around the Sun.*

Original EST: *Maa läbib oma orbiidil ümber päikese tohutu vahemaa, kiirusega umbes 30 km/s või üle 108000 km/h.*

Reference EST: *Katse maa liigub ümber Päikese väga kiiresti (100,000 km tunnis).*

- Output ENG: *The Earth travels a very large distance in its orbit around the Sun.*
- Output EST: *Maa liigub oma orbiidil ümber Päikese tohutu vahemaa.*

Original ENG: *Earth's interior remains active, with a thick layer of relatively solid mantle, a liquid outer core that generates a magnetic field, and a solid iron inner core.*

Reference ENG: *Everything that lives on Earth is on top of the crust. Below that is a layer of thick, liquid rock called the mantle. Under that is a thin liquid layer called the outer core and then the solid iron inner core.*

Original EST: *Maa sisemus jääb aktiivseks, suhteliselt tahke vahevöö paksu kihiga, magnetvälja tekitava vedela välissüdamikuga ja tahke rauast sisesüdamikuga.*

Reference EST: *Kõik, mis maal elab, on maakoore peal. Selle all on paksu vedela kivimi kiht, mida nimetatakse vahevööks. Selle all on õhuke vedel kiht, mida nimetatakse välissüdamikuks ja seejärel tahkeks rauast sisemiseks südamikuks.*

- Output ENG: *The interior remains active, with a thick layer of relatively solid mantle, a liquid outer core that generates a magnetic field, and a solid iron inner core.*
- Output EST: *Maa naabruses jääb aktiivseks suhteliselt mähitud vahevöö ja tahke kangast.*