

Linköping University | Department of Computer and Information Science

Bachelor's thesis, 18 ECTS | Kognitionsvetenskap

2023 | LIU-IDA/KOGVET-G--23/019--SE

# Context-aware Swedish Lexical Simplification

- Using pre-trained language models to propose contextually fitting synonyms

---

*Kontextmedveten lexikal förenkling på svenska: Användningen av förtränade språkmodeller för att föreslå kontextuellt passande synonymer*

**Emil Graichen**

Supervisor : Arne Jönsson  
Examiner : Lars Ahrenberg

## Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>.

## Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>.

## Abstract

This thesis presents the development and evaluation of context-aware Lexical Simplification (LS) systems for the Swedish language. In total three versions of LS models, *LäsBERT*, *LäsBERT-baseline*, and *LäsGPT*, were created and evaluated on a newly constructed Swedish LS evaluation dataset. The LS systems demonstrated promising potential in aiding audiences with reading difficulties by providing context-aware word replacements. While there were areas for improvement, particularly in complex word identification, the systems showed agreement with human annotators on word replacements.

The effects of fine-tuning a BERT model for substitution generation on easy-to-read texts were explored, indicating no significant difference in the number of replacements between fine-tuned and non-fine-tuned versions. Both versions performed similarly in terms of synonymous and simplifying replacements, although the fine-tuned version exhibited slightly reduced performance compared to the baseline model.

An important contribution of this thesis is the creation of an evaluation dataset for Lexical Simplification in Swedish. The dataset was automatically collected and manually annotated. Evaluators assessed the quality, coverage, and complexity of the dataset. Results showed that the dataset had high quality and a perceived good coverage. Although the complexity of the complex words was perceived to be low, the dataset provides a valuable resource for evaluating LS systems and advancing research in Swedish Lexical Simplification.

Finally, a more transparent and reader-empowering approach to Lexical Simplification is proposed. This new approach embraces the challenges with contextual synonymy and reduces the number of failure points in the conventional LS pipeline, increasing the chances of developing a fully meaning-preserving LS system.

**Keywords:** automatic text simplification, lexical simplification, Swedish, BERT, GPT-3, evaluation dataset, synonymy

# Acknowledgments

I would like to express my sincere gratitude to Arne Jönsson for his guidance and support throughout the process of writing this thesis. Furthermore, I would like to extend my appreciation to the annotators who dedicated their time and efforts to develop the evaluation dataset. Their work resulted in the first evaluation dataset for Swedish Lexical Simplification.

I am also grateful to everyone in my "Språkteknologerna" seminar group for their constructive feedback and stimulating discussions. Their diverse perspectives and thoughtful comments have significantly contributed to the quality and depth of this work. I would also like to thank Evelina Rennes and Daniel Holmer who helped me formulate my research questions in the starting phase of this project. Furthermore, I would like to thank Carlos Palomino for coming up with the clever `LäsBERT` name for one of my lexical simplifiers.

Lastly, I would like to thank all individuals mentioned above and for their invaluable contributions and support throughout this thesis. A special shout-out to student cafe Baljan for providing me with cheap coffee which was essential for the completion of this thesis.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	2
1.2 Research questions . . . . .	3
1.3 Delimitations . . . . .	3
<b>2 Theory</b>	<b>4</b>
2.1 Automatic Text Simplification . . . . .	4
2.2 Word Complexity . . . . .	5
2.3 Synonymy . . . . .	6
2.4 Lexical Simplification . . . . .	6
2.5 Lexical simplification in Swedish . . . . .	9
2.6 Transformer Language Models . . . . .	9
<b>3 Data</b>	<b>12</b>
3.1 Linguistic Resources . . . . .	12
3.2 Creating the evaluation dataset . . . . .	14
<b>4 Method</b>	<b>18</b>
4.1 The algorithm . . . . .	18
4.2 Implementation . . . . .	20
4.3 Evaluation . . . . .	25
<b>5 Results</b>	<b>27</b>
5.1 Performance of the Random Forest Classifier . . . . .	27
5.2 Perplexity of LäsBERT models on easy-to-read texts . . . . .	28
5.3 Performance of LS-systems on the evaluation dataset . . . . .	28
5.4 System-annotator agreement . . . . .	30
<b>6 Discussion</b>	<b>31</b>
6.1 System performance results . . . . .	31
6.2 Method . . . . .	35
6.3 The evaluation dataset . . . . .	36
6.4 The work in a wider context . . . . .	37
6.5 Moving beyond a questionable premise . . . . .	38

<b>7 Conclusion</b>	<b>40</b>
<b>Bibliography</b>	<b>41</b>
<b>A GPT Prompt</b>	<b>45</b>

# List of Figures

2.1	Lexical Simplification Pipeline. Adapted from Shardlow (2014) and G. Paetzold and Specia (2016). . . . .	7
3.1	Structure of evaluation dataset. . . . .	16
4.1	Steps in Complex Word Identification . . . . .	21
4.2	Steps in Substitute Generation. The steps describe the procedure for both LS systems. The steps in <code>LäsBERT</code> is represented by the upper pipeline, and the steps in <code>LäsGPT</code> are described in the pipeline below. . . . .	23
4.3	Substitution generation with BERT. Green represents generated alternatives, and the vertical line represents the delimiter between the cloned sentences. Adapted from Qiang et al. (2021) . . . . .	23
4.4	GPT prompt example. . . . .	24
4.5	Substitution filtering and selection . . . . .	25

# List of Tables

3.1	The distribution of words and their corresponding complexity level (Smolenska, 2018) . . . . .	14
3.2	Algorithm steps to create the evaluation dataset . . . . .	15
3.3	Percent of quadruples annotated with "TRUE" in response to the statements regarding <i>Quality, Coverage, and Complexity</i> . . . . .	17
4.1	Algorithm steps to lexically simplify sentences . . . . .	19
5.1	The precision, recall, and F1-score of the RFC used for CWI. The support column represents the distribution of classes in the test set. . . . .	27
5.2	The accuracy of the RFC used for CWI. . . . .	27
5.3	The perplexity of the models on unseen part of the fine-tuning dataset. . . . .	28
5.4	How many times the systems identified and replaced the complex word in a sentence with another word. The best performance is highlighted using bold font. . . . .	28
5.5	Synonym replacements that resulted in the complex word being exchanged for a synonym in the dataset. The best performance is highlighted using bold font. . . .	29
5.6	Synonym replacements that resulted in the complex word being exchanged for a synonymous word that is more frequent than the original word. The best performance is highlighted using bold font. . . . .	29
5.7	The proportion of words that the LS systems and the annotators marked as complex. The best performance is highlighted using bold font. . . . .	30



# 1 Introduction

Text simplification involves modifying a text to enhance its comprehension while maintaining its original meaning. Traditionally, this task was performed manually by authors to make texts more accessible, particularly for individuals with reading difficulties like dyslexia or intellectual disabilities. In the digital age, where online information dominates, the internet serves as a platform for distributing a wide range of content, including government information, media, and democratic discussions. However, the reliance on written text can pose challenges for those who struggle with decoding and understanding such content. Developing tools to aid users with reading difficulties is therefore more relevant than ever. Automatic Text Simplification (ATS), a subfield of Natural Language Processing (NLP), addresses this issue by leveraging technology to simplify text while preserving its informational content (Rennes, 2022).

ATS offers notable advantages over human-authored simplification, such as speed, scalability, and cost-effectiveness. Every modification made to a text carries the inherent risk of resulting in a deviation from the author's intended meaning. This is because no words mean exactly the same thing and changing the sentence structure can lead to different text styles. Minimizing the meaning-altering effect of ATS is an important objective to preserve the original text's integrity. Two primary approaches to automatic text simplification are *syntactic simplification* and *lexical simplification*. *Syntactic simplification* involves implementing structural modifications like sentence shortening and word order changes. *Lexical simplification* focuses on identifying and substituting complex words and phrases with simpler alternatives (Rennes, 2022). This thesis focuses on the development of lexical simplification (LS) systems and the collection of an evaluation dataset for Swedish LS. This is important due to the lack of developed lexical resources available for other languages than Swedish (G. H. Paetzold & Specia, 2017).

The conventional methodology for LS involves substituting complex words and phrases according to a predefined set of rules (Qiang et al., 2021). The method broadly involves looking up synonyms of target words in a thesaurus, i.e. a collection of words and their synonyms, and ranking them according to their simplicity and appropriacy. Thesaurus approaches face challenges due to the fact that thesaurus-synonyms are not synonymous in all contexts which

makes some substitutions nonsensical. Extracting the sense of the word becomes essential to ensure meaningful and contextually appropriate substitution candidates (De Belder & Moens, 2010).

Recently, with the introduction of large-scale pre-trained transformer language models such as BERT (Bidirectional Encoder Representations of Transformers) (J. Devlin et al., 2018) and GPT-3 (Generative Pre-trained Transformer 3) (Brown et al., 2020) a new chapter of NLP has begun. GPT-3 and BERT perform well on a broad set of downstream NLP tasks (Brown et al., 2020; J. Devlin et al., 2018). This has already resulted in Lexical Simplification systems for English that used BERT to generate substitutions for a given complex word (Qiang et al., 2021). The benefit of using these models, trained on vast amounts of text, over existing thesaurus methods is that they should avoid the challenges of contextual synonymy. By capturing the statistical co-occurrence patterns of words, these models can generate more contextually appropriate substitutions for complex words. The LS systems developed for this thesis can be part of building inclusive text simplification tools for Swedish and provide further insight into the potential of applying pre-trained language models in the field of Swedish lexical simplification.

## 1.1 Contributions

Three versions of a Swedish LS system were developed for this thesis: Two versions of an LS system (called *LäsBERT*) inspired by the approach by Qiang et al. (2021) using two Swedish BERT models for substitution generation. One version contains a fine-tuned BERT model and one uses an out-of-the-box model. Fine-tuning a model involves adapting the model to a specific task or dataset by training it with task-specific data. The two models are developed to investigate how fine-tuning affects the end-to-end performance of the LS systems. Furthermore, a GPT-3 based LS system (called *LäsGPT*) was developed which uses OpenAI:s GPT-3 for generating substitutes. These three systems are evaluated on a collected evaluation dataset.

This is important because it investigates the applicability of advanced language models in the field of lexical simplification. Moreover, this work builds on earlier research, providing further evidence and validation for findings made in previous studies (e.g. implementing Swedish Complex Word Identification (Smolenska, 2018) systems in an end-to-end LS system). The development and research of LS systems contribute to the ongoing pursuit of an effective lexical simplification system, aiming to aid audiences with reading difficulties to participate more equally in a text-based information society. The systems' context awareness enables more accurate and meaningful word replacements, hopefully enhancing the overall quality of the simplified texts.

A challenge for Lexical Simplification is the lack of evaluation datasets for other languages than English (G. H. Paetzold & Specia, 2017). This thesis addresses this by developing the first<sup>1</sup> Swedish LS dataset. Developing an evaluation dataset for Swedish lexical simplification is important because it allows for intrinsic and resource-effective evaluation and comparison of different LS systems. The developed LS system (see Section 3.2) is freely available which hopefully will facilitate the faster development of other Swedish LS systems.

---

<sup>1</sup>To the author's knowledge.

## 1.2 Research questions

1. How effective are the developed context-aware LS systems, LäsBERT and LäsGPT, in generating contextually appropriate word substitutions for Swedish lexical simplification?
2. How does fine-tuning the BERT model affect the end-to-end performance of the LäsBERT lexical simplification system?

## 1.3 Delimitations

This thesis investigates the performance of LS systems implementing pre-trained language models. The LS systems and evaluation datasets developed are limited to the Swedish language, which has substantially fewer lexical simplification resources available for research. This thesis focuses on lexical simplification (LS) and does not address the substitution or shortening of larger text units, which falls under the domain of automatic text summarization (Rennes, 2022). Specifically, the thesis only focuses on single words and does not investigate phrase substitution or syntactical changes to sentences.



## 2 Theory

This chapter will introduce Lexical Simplification (LS) and Automatic Text Simplification in general and describe the literature on Swedish LS specifically. The LS pipeline is broken down into distinct steps to illustrate possible design choices and challenges in the construction of automatic LS systems.

### 2.1 Automatic Text Simplification

Text Simplification refers to the process of transforming a text into a simpler one, whilst preserving the core meaning of the original text (Rennes, 2022). There are guidelines for human authors that aim to outline how language should be used to meet the needs of broad reader audiences. These guidelines involve syntactic, lexical, organizational, and semantic guidelines to write more comprehensible texts (PLAIN, 2011).

*Automatic Text Simplification (ATS)* involves formalizing and implementing the principles of these guidelines computationally, which is faster and cheaper than human-authored text simplification. A text to be simplified can be transformed in several ways. First, the *syntax* of the text can be changed. This involves modifying the word order, shortening sentence lengths, and substituting a passive voice with an active voice to achieve a more comprehensible result (Rennes, 2022). Secondly, *semantic simplification* refers to simplifying and explaining complex concepts. In the Federal Plain Language Guidelines, this is represented by recommending authors to provide readers with examples or structuring information in tables (PLAIN, 2011). The third method, which constitutes the main focus of this thesis, is substituting words and phrases in a text to make it more comprehensible. This is called *lexical simplification* and involves finding complex words or phrases in a body of text and finding both suitable and simpler synonyms to replace them with. Henceforth, Lexical Simplification will refer to Automatic Lexical Simplification, which involves constructing a system that replaces complex words algorithmically.

Although the process of lexical simplification can be outlined in a sentence, there are several questions that need to be answered in order to perform it successfully. What makes a word hard to understand? How can we identify it automatically? Why are some words complex

for some audiences, and not for other audiences? Does synonymy exist in the sense that two words mean exactly the same thing? How do we make sure that the alternatives we propose to complex words aren't actually making the text harder to understand or changing the meaning of the text? What can be done to develop ATS systems in languages with considerably fewer resources available than English? The answers to these questions are still topics of research. The following sections are going to outline some of the current findings in the literature and lay the foundation for the implementation of the Automatic Lexical simplification systems that have been developed for this thesis.

## 2.2 Word Complexity

What makes a word complex? Most people would agree that words that refer to the same concept can be variably difficult to understand. Many would probably agree that the word *erroneous* is more complex than *wrong*. But what properties of the word *erroneous* make it more difficult for readers to understand? This question has sparked research, outlining which properties that seem to make some words more difficult than others. There is no broadly accepted definition of what a complex word is; complexity is a subjective matter and varies between reader audiences and individuals. What is easy to read for a competent native language speaker isn't necessarily easy to understand for a novice second-language learner, a child, or someone with aphasia (Knollman-Porter et al., 2015). To discuss word complexity in a meaningful way, it is necessary to declare the assumptions that underlie the discussion. For whom are these words difficult? In which way are the words hard to understand? What are appropriate ways to simplify readers' interaction with texts? Identifying complex words can be a challenging task as there are no solutions that apply to all audiences and no single property that constitutes word complexity. However, the following word properties are usually mentioned when describing complex words:

*Word length* has been included in a broad set of readability formulas and is hypothesized to be correlated to a text's overall complexity (DuBay, 2004). Using metrics such as the number of syllables, number of characters, or number of vowels are used to represent word length (Gooding et al., 2021; Yimam et al., 2017). If a word is more frequently encountered in a language, it is more likely that readers experience them as less complex (De Belder & Moens, 2010). This indicates that *word frequency* is an important factor when determining the complexity of a word. Furthermore, complexity and perceived familiarity of a word are negatively correlated, indicating that more frequent words are easier for readers to understand (Rudell, 1993). Furthermore, when tasked to develop methods to identify complex words in texts word length and frequency were shown to be good predictors of word complexity. The results seem to generalize over several languages (Bingel & Bjerva, 2018; Yimam et al., 2017). Features such as *concreteness* and *imageability* could also have some effects on how comprehensible words are (Begg & Paivio, 1969).

Carroll et al. (1998) applied the assumption that frequency constitutes complexity to lexically simplify newspaper articles that used WordNet (Miller, 1995), a large lexical database, to find synonyms for infrequent words. The frequency of the words were obtained by querying the Oxford Psycholinguistic Database (Quinlan, 1992). If a word in the newspaper article had synonyms in WordNet with a higher frequency in the Oxford Psycholinguistic Database the words were replaced. Biran et al. (2011) defined word complexity as a product of *corpus complexity*, representing word frequency in a corpus and *lexical complexity*, the number of characters of a word. Smolenska (2018) developed and evaluated Complex Word Identification systems for Swedish and found that frequency features were the most informative when classifying the complexity of words. She also showed that classification scores could be maintained by only using frequency features. The source of the data significantly influenced the

system performance. Certain corpora exhibited a higher degree of informativeness in terms of word frequency compared to others. Corpora containing everyday language such as the language used in blogs and in tweets were more informative than niche corpora of texts from sources such as medical texts or movie subtitles (Smolenska, 2018).

## 2.3 Synonymy

Synonyms are words or expressions that have the same or nearly the same meaning in some or all senses (Wei et al., 2009). Does synonymy exist in the strict sense? Rennes (2022) exemplified that synonyms for the word *woman* could be words such as *girl*, *female*, *lass*, *gal*, and *lady*. Although these words represent the same concept, they all carry different connotations. G. Paetzold and Specia (2016) called the idea that words are irreplaceable the *Robbins-Sturgeon hypothesis*. The hypothesis was named after authors Tom Robbins and Theodore Sturgeon who expressed that there are no words that are perfectly interchangeable. Slight shifts in meaning between two "synonymous" words can alter how appropriately a text fits certain messengers and receivers. After all, it isn't appropriate for government web pages to refer to *women* as "*gals*". This indicates that looking up synonyms for a target word in a thesaurus might not be enough to account for all contextual factors influencing the fitness of a given synonym.

Thesaurus-synonyms are also not necessarily synonymous in all contexts, which further complicates the task of lexical simplification. One example of this is the word *critical* which could both refer to someone being inclined to *criticize* others, as well as referring to the *vitality* and *importance* of a component in a system. Which word sense should be selected given a specific context? To native human readers this task is usually performed without difficulty, but translating these *word sense disambiguation* strategies into a machine-friendly format has been an ongoing challenge since the birth of Natural Language Processing (Stevenson & Wilks, 2003). Accurately representing word senses is crucial to develop precise and well-performing LS systems.

## 2.4 Lexical Simplification

Shardlow (2014) and G. Paetzold and Specia (2016) described the general pipeline of Lexical Simplification (LS) (see Figure 2.1). The first step is *Complex Word Identification* (CWI) which aims to find candidates in need of simplification. *Substitution Generation* describes the process of generating alternative words to the identified complex words. To preserve the meaning of the text that is being simplified it's important that these words are synonymous with the complex word. The generated alternatives that are the most synonymous are selected in the next step conveniently named *Substitution Selection*. Finally, to improve the comprehensibility of the processed text it's important that the remaining words are ranked according to simplicity, where the simplest word is chosen as the final word substitute. This final step is therefore called *Substitution ranking*. There is no consensus in the literature on the best ways to perform any of these subtasks. Consequently, this section can't be said to represent all possible approaches to the different LS subtasks. The aim of this section is to illustrate the considerations behind the LS systems presented in this thesis.

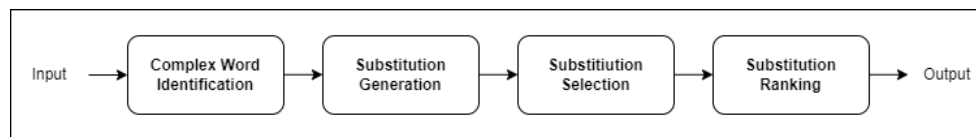


Figure 2.1: Lexical Simplification Pipeline. Adapted from Shardlow (2014) and G. Paetzold and Specia (2016).

### Complex Word Identification

Shardlow (2013) showed that the performance of an overall LS system is dependent on how well the identification of complex words works. If too many words are identified as complex, the system ends up making unnecessary substitutions which might alter the meaning of the sentences too much. If too few words are selected, the output text is not simplified enough.

Smolenska (2018) developed and evaluated systems for Swedish CWI, along with a dataset for training models on this task. It was found that a Random Forest Classifier (RFC) (Breiman, 2001) trained on fifteen (15) features concerning the frequency and syntactic function of the word performed best at the task of classifying complex words. It was concluded that using only the frequency features in the training of the classifier could maintain the scores of the classifier.

### Word Features

A word feature in this case refers to different linguistic features that words possess. Smolenska (2018) classified these features into five sub-categories: *morpho-syntactical*, *contextual*, *syntactical*, *conceptual*, and *frequency* features.

1. *Morpho-Syntactic* features refer to the form and structure of a word, such as the part-of-speech tag, the ratio between vowels and consonants, and the number of characters in the word.
2. *Contextual* features represent features that describe the surroundings of a word. Examples of such features are the mean sentence length, the mean word length in a sentence, and the mean number of punctuation marks.
3. *Syntactic* features broadly represent the syntactic function that the words have in relation to the root words, e.g. nominals, clauses, modifiers, and function words.
4. *Conceptual* features represent information about the semantic content of the words. One feature of this category could be how many senses a word has.
5. *Frequency* features were the most informative and refer to how often a word occurs in relation to other words. The source of the words had an impact on the informativeness of the frequency, where a low frequency in everyday language corpora (e.g. blogs and online communication) was positively correlated with word complexity (Smolenska, 2018).

### Substitution generation

When the complex words are identified within the input text, the LS system needs to generate alternatives to the found words. There are several approaches to *substitution generation* present in the literature. The goal is to generate suitable substitutes for the input complex word. The words generated should preserve the meaning of the text and if possible be substitutes that simplify the text.

Keskisärkkä (2012), E. Abrahamsson et al. (2014), and P. Abrahamsson (2011) used the Swedish synonym dictionary SynLex (Kann & Rosell, 2006) (see Section 2.5 for a more detailed description) to find appropriate synonyms for target words. This approach is based on using established dictionaries to generate alternatives and is also commonly found in the literature for English LS (De Belder & Moens, 2010; S. Devlin, 1998; Gooding & Kochmar, 2019). A more recent method to generating alternative words to an input word is by comparing the word embeddings of the input to other semantically similar words (Glavaš & Štajner, 2015; G. Paetzold & Specia, 2016; Rennes, 2022). Both these methods usually operate on a word level when generating substitution candidates. The possible drawback of analyzing words without their context is that it might result in generating synonyms that aren't synonyms in the specific context in which they are found (see the discussion in Section 2.3).

Using pre-trained encoders such as BERT to reformulate the task of substitute generation into a Masked Language Modelling (MLM) task (see Section 2.6 for a thorough description) has been done to avoid the problem of disregarding context in LS tasks (Pimienta Castillo, 2021; Qiang et al., 2021). The method of reformulating the substitution generation task into an MLM task works by obscuring the complex words in an input text with [MASK] tokens and letting the BERT model generate a probability distribution over suitable alternatives that fit into the slot of the obscured word. To avoid losing the semantic effect the hidden word has on the overall meaning of the sentence, the sentence was cloned, and fed pairwise into the model. One sentence was fed into the model without the complex word masked out and one sentence with the complex word hidden (Qiang et al., 2021). The model then generates words that should fit into the [MASK] slot which hopefully are simpler than the original word.

Substitute generation has also been reformulated as a language generation task (Lee et al., 2021). To generate suitable alternatives to specific words in a short paragraph they utilized the in-context learning abilities (see Section 2.6) of GPT-3 (Brown et al., 2020) to generate substitutions.

### Substitution Selection and Ranking

The goal of *Substitution selection and ranking* is to remove all generated alternatives that aren't grammatical, remove candidates that alter the meaning of the sentence too much, and order the remaining words after simplicity.

Gooding and Kochmar (2019) filtered and ranked the generated substitutes based on three factors: contextual simplicity, contextual semantic equivalence, and grammaticality. Contextual simplicity was calculated by reusing the sequential CWI model used earlier in their pipeline to check if a given substitution generated a simpler sentence than the original word. Contextual semantic equivalence utilized ELMo embeddings (Peters et al., 2018) to encode the sentences and to calculate the cosine distance between the substitutes and the original word in the context of the sentence that was to be simplified. To check whether or not a generated word was grammatical in a sentence, the occurrence of bigrams (a lexical unit consisting of two words) in a corpus was evaluated. If the replacement word together with its right or left neighbor formed a bigram that didn't occur once in the corpus it was assumed that the bigram was ungrammatical, and thus removed (Gooding & Kochmar, 2019).

Others have used the probability distribution of words that BERT returns in the MLM task to determine the likelihood of a generated substitution being a "relevant" substitute (Qiang et al., 2021). Frequency features of words are usually one component of the ranking system, where words that are more frequent are preferred over less frequent words (Keskisärkkä, 2012; Qiang et al., 2021). Ranking synonyms exclusively based on the number of characters

in a word has also been proposed, but this approach has some considerable limitations (P. Abrahamsson, 2011).

## 2.5 Lexical simplification in Swedish

Decker (2003) developed and formalized ATS for Swedish. The approach was based on rules that either added or deleted linguistic elements and altered the overall structure of the texts. The rules were extracted by formalizing the differences between manually simplified texts and their corresponding original text. Rybing and Smith (2009) implemented these rules into a text rewriting tool, COGFLUX, which P. Abrahamsson (2011) expanded with a synonym replacement module.

Keskisärkkä (2012) evaluated synonym replacement methods to lexically simplify texts and evaluated the results with readability measures, the percentage of long words, word length, and replacement error ratio. The approach was to exchange difficult words with easier ones based on different word properties. The properties that were evaluated were word frequency, word length, and level of synonymy with the original word. Lexical resources such as the synonym dictionary SynLex (Kann & Rosell, 2006) (see Section 3.1) were used to find words and their corresponding synonyms. E. Abrahamsson et al. (2014) used a similar approach as Keskisärkkä (2012) to lexically simplify medical texts.

Rennes (2022) developed STILLETT, an automatic text simplification tool, which included both a syntactic simplifier and a lexical simplifier. To find more comprehensible synonyms two methods were evaluated. The first method was inspired by Lin et al. (2003) and was based on the assumptions that words that share translations often are synonyms, that words appearing in similar contexts have a similar meaning (i.e the *distributional hypothesis*) (Harris, 1970) and that a collection of Easy Language texts would yield more comprehensible words than a corpus with standard language.

Two synonym replacement methods were developed and evaluated. The **first method** implemented word2vec word embeddings to find the 40 most similar words to an input word. Of these 40 words only those occurring in a corpus with Easy Language were extracted to then be translated into English. If the word alternatives shared a translation with the input word, they were assumed to be synonyms. Of the remaining words the one with the highest semantic similarity was selected as the replacement. In the **second method**, the target word was translated into English, resulting in several English words. Each of these was translated back to Swedish and followed the same filtering and ranking system as in the first method: check for occurrence in the Easy Language corpus and select the most similar remaining candidate. An online survey evaluated the level of synonymy between the words generated using both methods. For both methods, the median response was that the generated words were synonymous with the original word *sometimes*, which seems to reflect the fact that synonymy is context-dependent (see Section 2.3). More participants rated the words generated by the first method as synonymous compared to the second method. When both methods proposed the same word, the perceived level of synonymy was the highest (Rennes, 2022).

## 2.6 Transformer Language Models

The influential Transformer model was introduced by Vaswani et al. (2017) in response to the drawbacks of Recurrent Neural Networks (RNN:s) and Convolutional Neural Networks (CNN:s). RNN:s and CNN:s struggled with sequence modeling tasks (such as language generation) because of their inability to process input sequences in parallel, which is crucial in the processing of larger bodies of text. The Transformer model's attention mechanism allows

it to map all parts of the input sequence to an output sequence, generating better results in downstream NLP tasks. The attention mechanism also allows for faster training because of the possibility of processing the data in parallel.

## BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer model developed by J. Devlin et al. (2018). The novelty of the model was that it allowed the transformer to train bidirectionally, taking both the left and right context of a word into account. Previous models only processed sentences in a unidirectional manner, processing each word in order from left to right. To be easily adaptable to downstream NLP tasks the BERT framework included two steps. *Pre-training* and *fine-tuning*. *Pre-training* involves training the model on a large set of unlabelled data and *fine-tuning* is the process of adapting it to a specific task using labeled data. BERT was pre-trained on the BooksCorpus (Zhu et al., 2015) (800 million words) and English Wikipedia (2,500 million words) (J. Devlin et al., 2018).

The BERT model was pre-trained with the objective of Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM is inspired by the Cloze Procedure (Taylor, 1953), which originated as a procedure to estimate the relative readability between bodies of text. The procedure involves having subjects try to guess masked-out words in sentences based on the context provided by the surrounding words. The rationale for pre-training BERT on this task was to teach the model to consider both the right and left contexts of words, something that was possible with the introduction of the transformer (J. Devlin et al., 2018).

The pre-training procedure was to randomly mask out words in the training data and let the model try to predict what the masked-out words were. This was done to teach the models the relationships between words. To teach the model the relationship between sentences the model was also trained on NSP. NSP is the task of predicting whether or not a sentence follows another sentence. This was done by labeling sentence pairs from the training data as "IsNext" or "NotNext" given the first sentence. The model performed well on several downstream NLP tasks. According to the authors, this indicated that bidirectional pre-training was important to minimize the cost of adapting large language models to downstream NLP tasks as fine-tuning is comparatively inexpensive. Furthermore, it highlighted the importance that the models learned bidirectionally (J. Devlin et al., 2018).

## GPT-3

GPT-3 (Generative Pre-trained Transformer 3) was developed by Brown et al. (2020) and outperformed many other transformer models on several NLP downstream tasks. In contrast to BERT, the authors aimed to avoid having to fine-tune models with datasets of tens of thousands of examples to accomplish tasks. Humans can learn new, specific tasks by being shown examples of how a task is performed. The authors presented the example that giving instructions to humans such as: "*here are two examples of people acting brave; please give a third example of bravery*" which usually results in good answers. The process of specifying a task to the model in this way was called "*in-context learning*" and can be done by giving the model zero, one, or multiple examples of the required task.

The authors argued against the *pre-train* and *fine-tune* paradigm. The arguments were that the practicality of being able to give instructions to a language model by interacting with it naturally would outweigh the benefits of fine-tuning, the practicality to accomplish specific tasks without large datasets, and finally: problems could arise when a powerful model that easily picks up on irrelevant correlations (i.e overfitting) is fine-tuned on data possibly containing spurious correlations (Brown et al., 2020).

Brown et al., 2020 trained an autoregressive transformer model with 175 billion parameters on several large datasets. The datasets were a combination of parts of the Common Crawl dataset (Raffel et al., 2020)<sup>1</sup> and reference corpora sourced from books, web texts, and Wikipedia. The model’s in-context learning abilities were evaluated in three conditions. The conditions were based on the conditioning of the model, *i.e.* how many demonstrations the model was given before inference time. In the *one-shot setting*, one demonstration of the task was presented to the model along with the natural language instructions. In the *few-shot setting*, the task was demonstrated multiple times and in the *zero-shot setting*, only the natural language instruction was given to the model without any demonstrations (see Figure 4.4 for an example prompt). GPT-3 performed well in the zero-shot, one-shot, and few-shot settings in a number of NLP tasks such as the Cloze-task inspired by (Taylor, 1953), translation, general knowledge tasks, and finding the most appropriate ending to a story. GPT3’s performance of the language understanding benchmark superGLUE (Wang et al., 2019) was increased with the size of the model and the number of examples provided in the context (Brown et al., 2020).

---

<sup>1</sup>410 billion out of the 1 trillion tokens were used



## 3 Data

This chapter will describe the data collected from various sources and what the data was used for. The data was used to train the Random Forest Classifier (RFC) for Complex Word Identification, fine-tune BERT to generate easier substitutes, and construct the first evaluation dataset for the Swedish LS systems.

### 3.1 Linguistic Resources

In this section, the features of the linguistic resources employed in the study are going to be described. To facilitate replicability and improve the transparency of the developed LS systems and evaluation dataset the sources for the resources are included.

#### Resources from Språkbanken

Språkbanken, or *the Swedish Language Bank*, is a research e-infrastructure that aims to support research on language data. Språkbanken develops, publishes, and maintains language technology resources and tools for the Swedish language. It was established in 1975 and is located at the University of Gothenburg. Språkbankens resources include corpora of written and spoken text, lexical databases, and language technology tools. These resources are made available to researchers from different backgrounds, cultural heritage institutions, educators, and businesses to accelerate the development of language technology and the study of the Swedish language (Språkbanken, n.d.). The following linguistic resources are collected from språkbanken <sup>1</sup>:

#### Stockholm Umeå Corpus

The Stockholm-Umeå Corpus (SUC) is a balanced corpus with annotated Part-of-speech (POS) tags, morphological features, and lemmas. The texts of the corpus were collected in the nineties and originate from a variety of sources. A belonging collection of the token frequencies of the corpus is published on the Språkbanken webpage <sup>2</sup> which was used in this

---

<sup>1</sup><https://spraakbanken.gu.se/resurser>

<sup>2</sup><https://spraakbanken.gu.se/resurser/sucx3>

thesis for training the RFC. The version used in this thesis is SUCX 3.0. The suffix "X" denotes a shuffled version of the corpus that is free to use without a license (Ejerhed et al., 2006).

### BloggMix

Språkbanken hosts a collection of 21 corpora with blog texts<sup>3</sup>. The blogs were collected from 1998 to 2017 by collecting blogs appearing on the top lists of *bloggportalen.se*<sup>4</sup>, a Swedish homepage that hosted a platform for users to find blogs with various topics. This thesis used the frequency datasheet for the "*Bloggmix okänt datum*" corpus which consists of blog texts collected with an unknown publishing date. Token frequency, lemma, and POS tag are included in the frequency datasheet for each word. Smolenska (2018) found that word frequencies in blog corpora constitute the most informative features when predicting complex words. This corpus was therefore the primary source of word frequencies throughout this thesis (used for both training the RFC and the evaluation dataset).

### TwitterMix

TwitterMix is a corpus published by Språkbanken that contains frequency data from selected Swedish Twitter users.<sup>5</sup> The dataset is built out of approximately 500 million tokens and 52 million sentences sourced from Twitter. The dataset contains POS tags, lemma, and frequency for each word. The frequencies were extracted to train the RFC.

### 8sidor

8sidor is a Swedish newspaper with easy-to-read texts targeting audiences with different reading difficulties. The newspaper is produced by the Swedish Agency for Accessible Media and is published weekly (Myndigheten för tillgängliga medier, n.d.). A corpus of collected articles from 8sidor is available in Språkbanken<sup>6</sup>. The corpus contains over 420 000 sentences and over 4.5 million tokens. This data was used to fine-tune the BERT model to generate more comprehensible substitutes.

### LäSBarT

LäSBarT is a corpus containing easy-to-read texts sourced from children's books. The corpus found at Språkbanken contains a little over 100,000 sentences and 1 million tokens (Mühlenbock, 2008)<sup>7</sup>. This data was also used to fine-tune a BERT model to generate more comprehensible substitutes.

### Kelly Swedish List

The Kelly Swedish List (Kilgarriff et al., 2014; Volodina & Kokkinakis, 2012) is a lexical resource with over 8,000 Swedish word lemmas annotated with Word-per-million frequencies, word classes, and *Common European Framework of Reference for Language* (CEFR) scores. These scores, taking the values **A1**, **A2**, **B1**, **B2**, **C1**, and **C2**, correspond to language proficiency levels. **A1** represents the language of a basic user and **C2** represents the language of a proficient user (Volodina & Kokkinakis, 2012). The Swedish Kelly list<sup>8</sup> was used to construct the evaluation dataset for this thesis. All words in the Kelly list with the annotated complexity score of **C1** and **C2**, i.e. words that were assumed to be complex, were sourced in the creation of the evaluation dataset.

<sup>3</sup><https://spraakbanken.gu.se/resurser/bloggmix>

<sup>4</sup><https://www.bloggportalen.se>

<sup>5</sup><https://spraakbanken.gu.se/resurser/twitter>

<sup>6</sup><https://spraakbanken.gu.se/resurser/attasidor>

<sup>7</sup><https://spraakbanken.gu.se/resurser/lasbart>

<sup>8</sup><https://spraakbanken.gu.se/resurser/kelly>

### SALDO example sentences

SALDO, *Svenskt Associationslexikon*, is a Swedish lexical-semantic resource developed by Borin et al. (2013) containing word relations and their senses. The resource is developed to facilitate language research. Språkbanken provides a lexicon with example sentences where the words in SALDO are put into a sentence.<sup>9</sup> These example sentences were used together with the complex words in the Kelly Swedish List and synonyms in SynLex to construct the evaluation dataset.

### Other resources

#### Dataset for Swedish Complex Word Identification

Smolenska, 2018 collected a dataset of 4,238 words derived from Rivstart dictionaries (Natur och Kultur, n.d.), a series of textbooks designed for second-language learners of Swedish. The dataset was collected to train and evaluate CWI systems. The books in this series are structured along the progression of CEFR scores. The six categories, **A1** to **C2**, of sourced words, were grouped into three groups. A fourth group was added containing the most complex words. The words in the fourth group were sourced from Ordtestet<sup>10</sup>, a website that targets native Swedish speakers, where users can test their understanding of difficult words. The number of words and their corresponding complexity can be found in table 3.1. The data was used to train the RFC.

Complexity Level:	1	2	3	4
Number of words:	1699 (40.1%)	1056 (24.9%)	978 (23.1%)	505 (11.9%)

Table 3.1: The distribution of words and their corresponding complexity level (Smolenska, 2018)

### SynLex

SynLex (Kann & Rosell, 2006) was constructed by querying users of the Lexin translation service about the perceived level of synonymy between two words. The 82,000 word pairs of the lexicon are annotated with a synonymy score between 0-5 by a distributed user group. 0 represents no synonymy at all, and 5 represents two perfect synonyms. In the dataset used in this project<sup>11</sup> only synonyms that were rated at the synonymy level of 3 or higher were included. The number of synonymy relations in this dataset was over 38, 000 pairs. SynLex was used to find synonyms for the complex words in the evaluation dataset.

## 3.2 Creating the evaluation dataset

An evaluation dataset was created to assess if the LS systems find and substitute complex words. To the author's knowledge, there is no available benchmark or evaluation dataset for Lexical Simplification for Swedish. To be able to evaluate the performance of the systems, the dataset needs to include bodies of text to simplify, e.g. paragraphs or sentences. Several benchmarks for English LS (Kremer et al., 2014; Lee et al., 2021) use sentences to simplify. This dataset used sentences as well. The sentences need to contain one or more complex words for the LS system to find. This dataset just includes one identified complex word per

<sup>9</sup><https://spraakbanken.gu.se/resurser/salldoe>

<sup>10</sup><https://ord.relaynode.info/>

<sup>11</sup><http://folkets-lexikon.csc.kth.se/lexikon/synpairs.xml>

sentence. Lastly, suitable substitutions to the target word need to be identified and ranked according to simplicity. This dataset contains annotated substitutions for the complex words and the corpus frequencies of these substitutions. The dataset was collected automatically (Algorithm 3.2 and evaluated using human judgment (Table 3.3). Possible improvements are discussed in Section 6.3.

### Dataset collection

The collection of the data used in the evaluation dataset was performed automatically with the algorithm described in algorithm 3.2. The collection process began with retrieving all C1 and C2 level words in the Kelly Swedish list. These words represent words that are used by proficient users and were therefore regarded as complex words. The corpus frequency of these words in the BloggMix corpus was found (see row (4) in algorithm 3.2). Thereafter, available synonyms to the complex words in the SynLex thesaurus were found and saved in a Python dictionary (see row (5)). The corpus frequencies of these synonyms were also looked up and saved to the Synlex dictionary (see rows (6) and (7)). The final collection step was to find an example sentence in SALDO where the complex word occurred (row (8)). If at least one synonym for the complex word and one example sentence were found the row was written to the evaluation dataset CSV file (rows (9) to (12)).

Algorithm 3.2: Algorithm steps to create the evaluation dataset

---



---

```

1. Result: evaluation dataset
2.
3. for word in KellyComplexWords do:
4.     wordFreq ← findFreq(BloggMix, word)
5.     synDict ← findSynonyms(SynLex, word)
6.     for synonym in synDict do:
7.         synDict ← findFreq(BloggMix, synonym)
8.     exampleSent ← findExampleSentence(SALDO, word)
9.     if exampleSent == [ ] or synDict == { } do:
10.        NEXT(word)
11.     else do:
12.        WRITE(word, wordFreq, synDict, exampleSent)

```

---

### Structure of the Evaluation dataset

The structure of the evaluation dataset is illustrated in Figure 3.1. Each row in the dataset CSV file consisted of a quadruple containing a complex word found in the Kelly dataset, its frequency in the BloggMix corpus, a dictionary with synonyms and their corresponding frequencies, and an example sentence containing the complex word sourced from SALDO. If the complex word missed synonyms or example sentences it was not written to the evaluation dataset. 185 quadruples were found using the method outlined in Algorithm 3.2. All rows were manually annotated resulting in a dataset of 150 quadruples to evaluate Lexical Simplification systems on.

The structure of this dataset is resemblant to the structure of the SWORDS (Lee et al., 2021) benchmark for English Lexical simplification. SWORDS contains triplets of a sentence, a complex word, and one alternative to the complex word. This evaluation dataset and SWORDS differ structurally in two distinct ways. First, it includes all found and fitting synonyms to the target word in the same dataset quadruple. Secondly, the raw corpus frequencies from the BloggMix corpus are included both for the target word and for the synonyms. This can be

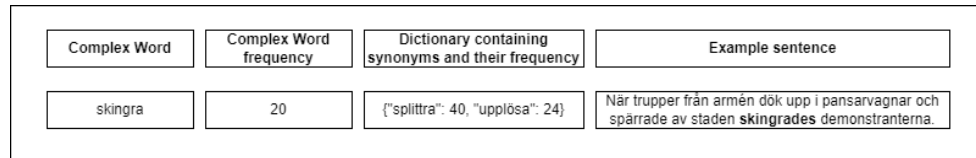


Figure 3.1: Structure of evaluation dataset.

used to evaluate if a substitution  $W \rightarrow W'$  substituted an infrequent word for a more frequent one.

### Annotating and evaluating the dataset

The annotation process involved manually checking all quadruples. All synonyms that did not "fit" into the context of the example sentence were removed. The fitness of a word refers to whether the synonyms were synonymous in the context of the example sentence. As discussed in Section 2.3, a synonymy relation is dependent on the context in which the words occur. To query a dictionary for all possible synonyms for a word  $W$  was therefore expected to yield some synonyms that didn't fit into the context of the example sentence. The Swedish word "*pendla*" both refers to *commuting* to work and a swinging, oscillating movement. The word "*oscillerade*" (eng: *oscillated*) therefore is not synonymous with the word "*pendla*" in the sentence "*Jag **pendlade** 1200 kilometer i veckan.*" (eng: *I **commuted** 1200 kilometers a week*). If all synonyms in the quadruple were removed the quadruple was deleted. A total of 35 quadruples were deleted through the manual annotation.

Three native Swedish student annotators were enlisted to annotate the dataset used in this study. The annotators assessed the *quality*, *coverage*, and *complexity* of the dataset. *Quality* refers to if the alternatives were synonymous with the complex word in the context of the example sentence. *Coverage* refers to if all possible synonyms were listed in the dataset. *Complexity* refers to the perceived complexity of the complex word. The student annotators were recruited from the Cognitive Science Bachelors program at Linköping University. To assess their word knowledge the annotator took two online versions of the vocabulary test in the Swedish academic aptitude test, *Högskoleprovet* (Universitets och högskolerådet, 2023). The maximum score for the two tests combined was 40. The annotators scored, 37, 33, and 35 on the tests respectively, indicating that all had good lexical proficiency.

Each annotator got 50 separate quadruples to evaluate to cover the whole dataset of 150 sentences. The annotators answered three questions corresponding to the *quality*, *coverage*, and *complexity* of the quadruples (see Table 3.3 for the evaluation results). They were evaluated by responding with True (represented by the number 1) or False (represented by the number 0) to the following statements:

- *Quality: All synonyms proposed could be exchanged with the complex word in the example sentence without changing the meaning of the sentence.*
- *Coverage: There are no missing possible substitutes I can think of that could substitute the complex word in the example sentence without changing the meaning of the sentence.*
- *Complexity: The complex word is actually a complex word.*

	Quality	Coverage	Complexity
% marked as TRUE :	86.6%	72 %	28.6%
# marked as TRUE :	130/150	108/150	43/150

Table 3.3: Percent of quadruples annotated with "TRUE" in response to the statements regarding *Quality*, *Coverage*, and *Complexity*.

The results show that the annotators in general agree that the synonyms proposed in the dataset fit in the context of the example sentence (86.6% of the quadruples). In 72% of the quadruples, the annotators thought that there were no omitted synonyms that could replace the complex word in the sentence. However, as discussed by Lee et al. (2021), humans generally don't recall all possible substitutions for a given word when working from memory. The *perceived* coverage of the dataset is therefore probably much higher than the *actual* coverage. The annotators did generally not think that the words sourced from the Kelly Swedish List were complex. However, since the annotators were native Swedish speakers with a university education, the perception of what constitutes a complex word might not generalize well to audiences with reading difficulties. The dataset is freely available at <https://github.com/emilgraichen/SwedishLSdataset>.



## 4 Method

This chapter will describe the implementation and evaluation of three lexical simplification (LS) systems, two versions of an LS system named LäsBERT, and one version of an LS system named LäsGPT. The structure of the chapter is based on the general pipeline of other LS systems described in Section 2.4. First, the overall LS algorithm and each step are explained. The LS systems are identical except for the *Substitution generation* subtask. Both LäsBERT versions use `KBLab/bert-base-swedish-cased` model developed by Malmsten et al. (2020). The versions differ in that one utilizes a BERT model that has been fine-tuned on two corpora of easy text. The other, the baseline model, uses the out-of-the-box `KBLab/bert-base-swedish-cased` model to generate substitutions. These systems generate substitutions as a Masked Language Modelling task. LäsGPT uses `OpenAI:text-davinci-003` GPT model (Brown et al., 2020) to generate substitutes as a language generation task (see Section 2.4). The chapter ends with an explanation of how all three LS systems were evaluated. The LäsBERT baseline version of the algorithm is freely accessible at <https://github.com/emilgraichen/SwedishLexicalSimplifier>.

### 4.1 The algorithm

The algorithm to lexically simplify sentences can be found in Algorithm 4.1. The algorithm describes the overarching logic and structure of the systems to illustrate how the systems are designed. Therefore, some details of the algorithm differ between the description in Algorithm 4.1 and the actual implementation. The algorithm works by looping through every word in a sentence. A Random Forest Classifier (RFC) scores the complexity of each word. All words that are scored with a complexity score of 3 or 4 are regarded as complex. For each complex word alternatives are generated with BERT or GPT. These are filtered and selected to preserve the meaning of the sentence as much as possible. The selected words are then ranked according to simplicity utilizing the same RFC used to identify the complex words. If one of the generated alternatives is classified as a simpler word than the complex word, the alternative word  $W'$  replaces the original complex word  $W$ .

Algorithm 4.1: Algorithm steps to lexically simplify sentences

---



---

```

1. Result: Lexically Simplified Sentence
2. sentence ← inputSentence
3. stopwords ← inputStopwords
4. for word in sentence do:
5.     if word not in stopwords do:
6.         complexWords ← ComplexWordClassifier(word)
7.
8.     for complexWord in complexWords do:
9.         maskedSentence ← maskComplexWord(sentence, complexWord)
10.        wordSubstitutes ← subsGeneration(maskedSentence)
11.        suitableSubs ← FILTER(wordSubstitutes)
12.        topSubs ← RANK(sentenceSimilarity(suitableSubs, complexWord))
13.        for sub in topSubs do:
14.            subsComplexity ← ComplexWordClassifier(sub)
15.            simplestSub ← HEAD(SORT(subsComplexity))
16.            CWComplexity ← ComplexWordClassifier(complexWord)
17.            if simplestSub < CWComplexity do
18.                REPLACE(Sentence, complexWord, simplestSub)

```

---

- 
- (4) (5) Loop over all words in the sentence that aren't stopwords.
- 
- (6) Generate Word features and classify the words as complex or not. Add the complex words to list complexWords
- 
- (9) The complex word in the sentence is replaced with a [MASK] token (LäsBERT) or highlighted with \*\*-symbols (LäsGPT).
- 
- (10) Substitute generation. Generated substitutes are added to list wordSubstitutes.
- 
- (11) Incomplete tokens and words that don't have the same POS tag as the original word are removed.
- 
- (12) Encode and calculate cosine similarity with sentenceBERT. Take the 5 most similar substitutions and add to list topSubs.
- 
- (13) (14) Generate word features for each word, use CWI classifier to get the complexity score for the 5 remaining substitutions.
- 
- (15) (16) Assign simplest substitution according to complexity score to simplestSub. Use CWI classifier to get complexity score for original complex word.
- 
- (17) (18) Replace simplestSub with complexWord if simplestSub has lower complexity score than complexWord.
- 
-

## 4.2 Implementation

This section is structured based on the LS pipeline described in Section 2.4. From now on the two versions of the LäsBERT system (one with a fine-tuned BERT model, and one baseline model without any fine-tuning) will be treated as one. This is because the purpose of developing two LäsBERT versions is to investigate what effect the fine-tuning of the BERT model has on the end performance of the LS system. The two LS systems LäsBERT and LäsGPT were identical except for the *Substitution Generation* subtask. This facilitates comparison between the models.

### Complex Word Identification

As described in Section 2.4, frequency is the main predictor for word complexity. Constructing a Random Forest Classifier (RFC) (Breiman, 2001) to classify word complexity only using frequency features can be built and generate good results (Smolenska, 2018). An RFC was trained using the `ensemble` module in the Python library Scikit-Learn (Pedregosa et al., 2011) utilizing the Swedish complex word dataset (see Section 3.1) developed by Smolenska (2018).

The preprocessing stage involved splitting the input sentence into individual words because the RFC works on a word-by-word basis. Non-alphanumerical characters of the sentence were also removed. The RFC generates a word complexity score between 1-4 based on the features of a word (described in Section 2.4). In this implementation, the features that were used to score the complexity of a word and train the RFC were the word's corpus frequency in the BloggMix, TwitterMix, and SUCX 3.0 (see Chapter 3) corpora together with the length of the word.

Informativeness was the basis for using the frequency datasets of the BloggMix, TwitterMix, and SUCX 3.0 corpora. According to Smolenska (2018), the selected corpora were amongst the most informative for predicting word complexity, which is why the corpora were suitable for this implementation. Earlier work has also established a relationship between word length and its complexity (see Section 2.2). The number of characters in each word was therefore used as the last feature for Complex Word Identification (CWI).

Words scored with "1" or "2" were treated as non-complex without the need for simplification. Words scored by the classifier to have a complexity of "3" or "4" were regarded as complex and were sent further down the pipeline for simplification. To avoid classifying words without semantic content, i.e. stopwords, all Swedish stopwords included in the NLTK resource `nltk.stopwords` (Bird et al., 2009) were removed from the input sentence. See rows (5) and (6) in Algorithm 4.1 and Figure 4.1.

A training and test dataset for the RFC was constructed with the words found in the CWI dataset (see Section 3.1) developed by Smolenska (2018). The number of characters was computed and the frequency of each word was looked up in each corpus and written to the RFC training dataset. The RFC training dataset was split into 90% training data and 10% test data. The performance of the RFC can be found in Section 5.1. The training dataset and RFC training algorithm can be freely accessed at <https://github.com/emilgraichen/SwedishCWI>.

The corpus frequencies were normalized by computing the common logarithm (the logarithm with a base of 10) of the frequency. The result represents how many times more common a word **W1** is to a word **W2**. When the common logarithm frequency of **W1** is equal to 1 and the common logarithm frequency for **W2** is 2 it means that **W2** occurs 10 times more than **W1**.

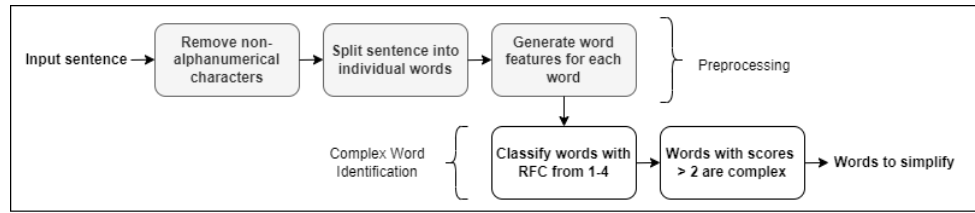


Figure 4.1: Steps in Complex Word Identification

in the corpus. This normalization method yielded the best results in earlier work (Smolenska, 2018).

### The CWI system in practice

The trained RFC in these implementations predicts the word complexity of a word based on four (4) word features. This is considerably fewer features than the optimal CWI system developed by Smolenska (2018) which used (15) word features to rate the word complexity. Generating 15 features for every word isn't practical, and therefore only the most informative features were selected. Although this limits the accuracy that can be achieved by the RFC, it is more practical to implement. Furthermore, since the most informative features already were selected, the law of diminishing returns probably would have applied to further added features. Generating more features would thus not have improved the classifier significantly. The work by Smolenska (2018), identifying the most informative word features for Swedish CWI, is therefore very useful.

### Substitute Generation

As described in Section 2.4 the LS subtask *substitution generation* aims to generate alternatives to a complex word. Both `LäsBERT` versions treat this task as a Masked Language Modelling (MLM) task and `LäsGPT` treats it as a Language generation task. This step is represented in rows (9) and (10) in Algorithm 4.1 and illustrated in Figure 4.2.

### LäsBERT

Two versions of `LäsBERT` were developed. The first version of `LäsBERT` used a fine-tuned BERT model, `KBLab/bert-base-swedish-cased`, developed by the Royal Library of Sweden (Malmsten et al., 2020). It was fine-tuned on easy-to-read texts and used to generate substitutes for the identified complex words. The second version of `LäsBERT` uses the original version of `KBLab/bert-base-swedish-cased`<sup>1</sup> without any fine-tuning. By developing two versions of the LS system, it is possible to investigate whether fine-tuning has any effect on the final performance of the overall LS system.

The idea to reformulate the substitution generation subtask as an MLM task (see Section 2.6) was developed by (Qiang et al., 2021) for English and was adapted in this thesis for Swedish. The idea involves obscuring a complex word with a `[MASK]` token and letting the BERT model predict the obscured word. The prediction consists of words that hopefully can be used as substitutes for the complex word.

To generate substitutes for a complex word the target sentence to be simplified was cloned into a sentence pair "`{S, S'}`". The second sentence `S'` had the identified complex words replaced with the `[MASK]` token and fed into the model. The rationale behind feeding the original sentence into the model twice is that this forces the model to consider the meaning

<sup>1</sup><https://huggingface.co/KBLab/bert-base-swedish-cased>

of the complex word when generating substitutes (see the next paragraph for an example). BERT predicted which word was obscured behind the [MASK] token based on both the right and left context (see Figure 4.3). A probability distribution was returned with substitutes and their corresponding probability. This should avoid the problem that thesaurus-based approaches face in the Substitution Generation subtask: that words generated aren't synonyms in all contexts.

A concrete example of how the substitution generation task is implemented with BERT (see Figure 4.3) is the following: Assume that the word **"complicated"** is identified as a complex word in the sentence: **"The test was complicated"**. The aim is to use BERT to propose alternative words that preserve the meaning of this sentence. First; the sentence is cloned into a sentence pair: **"The test was complicated. The test was complicated."**. Followed by this the complex word is masked out in the second sentence: **"The test was complicated. The test was [MASK]."**. The rationale behind feeding the original sentence into the model twice is that this forces the model to consider the meaning of the word **"complicated"** when generating alternative words. If the sentence isn't cloned and only **"The test was [MASK]"** is fed into the model, the model loses the meaning of the masked-out word, in this case, the word **"complicated"**. This could result in predicting that the sentence should be **"The test was [easy]"** which completely changes the meaning of the original sentence. Feeding a sentence pair into the model lets BERT consider the meaning of the complex word which seems to avoid the meaning-changing substitutions that the model proposes (Qiang et al., 2021). In both LäsBERT implementations, the models were tasked to generate 20 substitutes.

### Fine-tuning BERT

The original `KBLab/bert-base-swedish-cased` Malmsten et al. (2020) is trained on data that is aimed to be representative of the Swedish language. This involves training the model on texts from different sources and time periods to get a model that is trained on a representative sample of the Swedish language. This is, however, not necessarily desirable in the context of LS. The aim is to get the model to generate the easiest words possible to aid tasks downstream in the pipeline. The model should preferably have a bias towards easier words and suppress more difficult words in the MLM task. To accomplish this the `KBLab/bert-base-swedish-cased` model was fine-tuned on the LäsBarT and 8sidor corpora (see Chapter 3) which contains easy-to-read texts. The huggingface tutorial<sup>2</sup> (Huggingface, n.d.) to adapt masked language models to domain-specific data was used with an adaptation for LS and the `KBLab/bert-base-swedish-cased` model.

The fine-tuning of the BERT model in one of the LäsBERT versions began with creating a fine-tuning dataset. It involved reading the words from the 8sidor and LäsBarT corpora and concatenating the words into sentences. The sentences from both corpora were written to a text file and a random split into training and test sets was performed. 10 % of the dataset was used for testing and 90 % for training. The test set can be used to test the perplexity of the model, which is a measure of the model's (un)certainly in predicting a masked-out word. This in turn reflects the model's estimated word error rate when predicting a word (Chen et al., 1998). The perplexity of the models on unseen easy-to-read text can be found in Table 5.3.

The Python libraries TensorFlow (Martín Abadi et al., 2015) and Transformers (Wolf et al., 2020) were used for this implementation. The sentences were preprocessed through tokenization using the `KBLab/bert-base-swedish-cased` tokenizer and chunked together to a length of 128 tokens. Using the Transformers class `DataCollatorForLanguageModeling`

<sup>2</sup><https://huggingface.co/learn/nlp-course/chapter7/3?fw=tf>

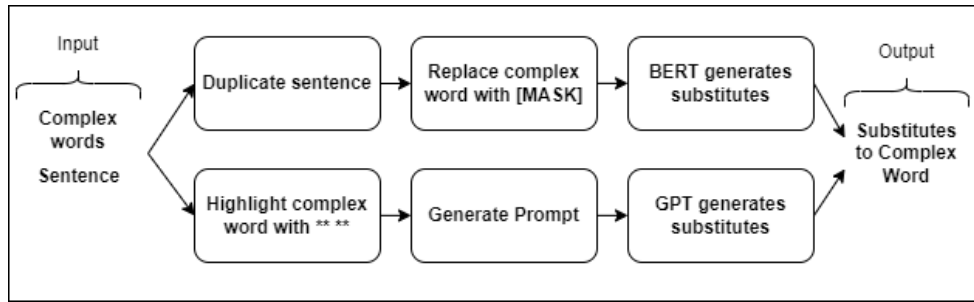


Figure 4.2: Steps in Substitute Generation. The steps describe the procedure for both LS systems. The steps in LäsBERT is represented by the upper pipeline, and the steps in LäsGPT are described in the pipeline below.

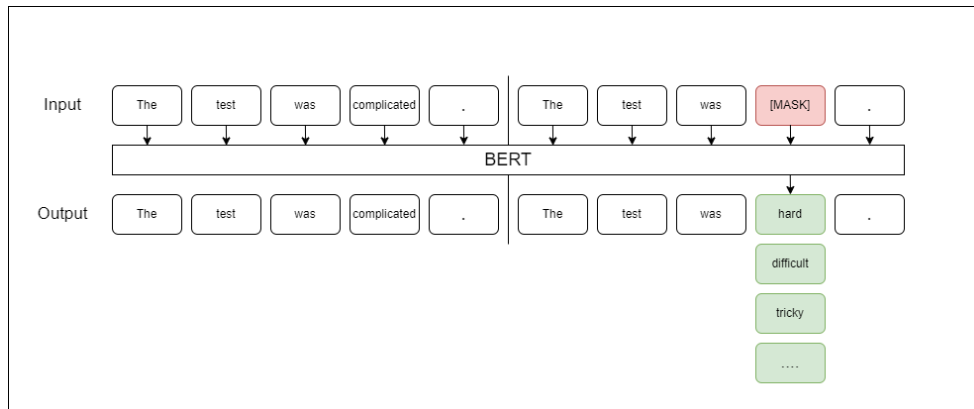


Figure 4.3: Substitution generation with BERT. Green represents generated alternatives, and the vertical line represents the delimiter between the cloned sentences. Adapted from Qiang et al. (2021)

15 % of the words were replaced with [MASK] tokens. The model was compiled with the following optimizer parameters: `init_lr=2e-5`, `num_warmup_steps=1000`, `num_train_steps=len(training_data)`, `weight_decay_rate=0.01`. The procedure to fine-tune the model originates from Huggingface (n.d.). The model was fine-tuned using the available GPU provided by a Google Colab Pro subscription.

### LäsGPT

OpenAI:s GPT 3.5 `text-davinci-003` model<sup>3</sup> was used to generate substitutes as a language generation task. To generate substitutes for the complex word reliably and in a predictable format the model needed to be prompted in an appropriate way. Brown et al. (2020) showed that conditioning the model with several examples of the task, i.e. few-shot learning, generally yielded the best results for several tasks. The prompt format and parameters used by Lee et al. (2021) to generate substitutes for English complex words were used. Since this thesis is about Swedish Lexical simplification, the prompts were adapted to Swedish. An example of the structure of the prompt can be seen in Figure 4.4.

Except for the `max_token` parameter, the same parameters used in Lee et al. (2021) were used in this implementation. `Temperature= 0`, `max_tokens = 100`, `top_p = 1`, `frequency_penalty = 0`, `presence_penalty = 0.5`. All other parameters were set to default.

<sup>3</sup><https://platform.openai.com/docs/models/gpt-3-5>

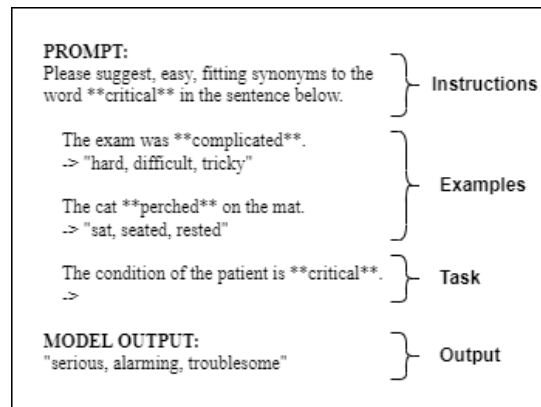


Figure 4.4: GPT prompt example.

An overview of the substitution generation task with GPT can be found in Figure 4.2. As illustrated in Figure 4.4 is that all complex words are highlighted with asterisks, which is one of many possible ways of emphasizing the complex word. Highlighting the complex word is the first step, followed by generating the prompt, to finally use the model to generate substitutions for the complex word. The prompt was constructed using the Python library LangChain<sup>4</sup>. The prompt was fed to the `text-davinci-003` model through the OpenAI API<sup>5</sup>. The exact prompt used in this thesis can be found in Appendix A.

### Substitute Filtering and Selection

This subtask aims to narrow down the pool of generated substitutions by removing unfit words and words that alter the meaning of the sentence too much. In Algorithm 4.1 these steps are described on rows (11) and (12).

#### Filtering substitutes

The generated words for both LS systems needed to be filtered to remove words that weren't good substitutions. A basic criterion for synonymy is that two words have the same Part-of-speech tag. Therefore, the POS tag for both the generated alternatives and the complex word was found in the SUCX 3.0 corpus. If the POS tags did not match the alternative was removed. If the generated token was empty or began with "##" (indicating that the generated substitute was an incomplete word), the words were removed as well.

#### Selecting substitutes

The Royal Library of Sweden developed a BERT model, `KBLab/sentence-bert-swedish-cased` that maps sentences to a 768-dimensional vector space (Rekathati, 2021). This facilitates comparison between sentences by calculating the cosine distance between their vector representations. The cosine distance represents a similarity score, ranging from 1 to 0. A score of 1 represents *total* similarity between the sentences and 0 represents *no* similarity between the sentences.

To select which of the generated substitutes preserve the meaning of the sentence as much as possible, new sentences were constructed replacing the complex word with each of the generated substitutes in the original sentence. By examining the similarity of sentences rather than comparing individual words using a thesaurus, the issue of contextual synonymy can be

<sup>4</sup>[https://python.langchain.com/en/latest/modules/prompts/prompt\\_templates/getting\\_started.html](https://python.langchain.com/en/latest/modules/prompts/prompt_templates/getting_started.html)

<sup>5</sup><https://platform.openai.com/docs/api-reference>

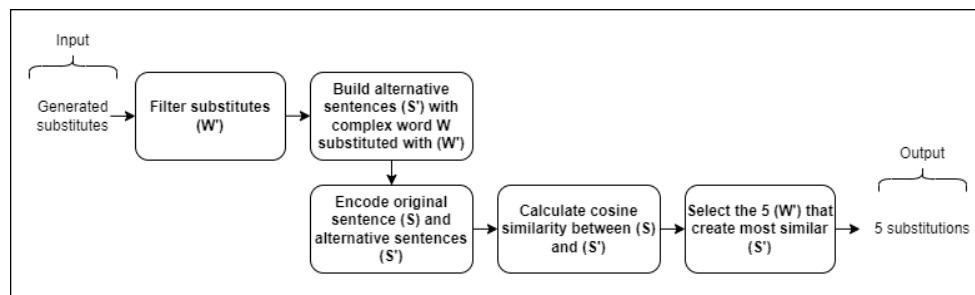


Figure 4.5: Substitution filtering and selection

avoided. This is because the meaning of a word varies depending on its context, as discussed in Section 2.3. The alternative sentences were encoded using the Sentence-BERT model and the cosine distance between the sentence vector representations of the original sentence and the alternative sentences were calculated. The five substitutes that created the most similar sentences were selected as the most meaning-preserving substitutes. See Figure 4.5 for an illustration of the steps taken in the Substitution selection subtask.

### Substitute Ranking

The five substitutes selected in the substitution selection task were words that preserve the meaning of the original sentence as much as possible. Assumptions regarding these words are that they are synonymous with the original word and that they fit into the context of the original sentence. The next step is to rank the selected substitutes according to simplicity to simplify the text as much as possible.

Word features (see Section 2.4) were generated for the selected substitutes and the original complex word. The RFC used in the CWI subtask (see Section 4.2) was used to rate the complexity of the selected substitutes and the original word. The easiest word was used as a replacement for the complex word. If the complex word was easier than all generated alternatives, no substitution was made. This step is important to minimize substitutions that replace the complex word with more difficult words. Replacing a complex word with a word with the same difficulty should be avoided. The more words that are replaced in a sentence, the more the meaning of the sentence is altered. If there is no obvious increase in readability when replacing a word with another, a simplification algorithm should be designed to be conservative, which is the case for this implementation. However, this conservatism negatively affects the recall of the system (see the next section). The substitute ranking steps are represented by rows (13) to (18) in Algorithm 4.1.

## 4.3 Evaluation

The goal of a Lexical Simplifier is to propose and exchange difficult words with easier ones with the purpose of reducing the lexical complexity of the overall text. To test how well an LS system replaces difficult words, an evaluation dataset is needed. To the authors' knowledge, no benchmark or evaluation dataset exists for Swedish lexical simplification. To be able to evaluate LS systems intrinsically the first Swedish LS evaluation dataset was therefore created (see Section 3.2). The dataset is meant to be representative of natural language and provide samples of texts where difficult words occur. The performance of the LS system can be evaluated by measuring the number of times it replaces complex words in these samples with replacement words that are both synonymous with and easier than the target word. The performance can also be split up into different performance measures that reflect different aspects of the overall performance, which can guide developers to which aspects of

the LS systems need to be attended to. It is important to note that the methods of evaluating the performance of LS systems vary in the literature and that the development of the evaluation dataset was not the primary area of interest for this thesis. Possible improvements are discussed in Section 6.3.

### **Performance Measures**

This section explicates the performance measures used for evaluating the Lexical Simplifier developed for this thesis. The main performance measures are Recall, Synonymous replacements, and Synonymous replacements with higher corpus frequency.

#### **Recall**

One way to assess how well an LS system simplifies a given text is by measuring the number of complex words it identifies and replaces. The proportion of identified and replaced words indicates how confident a user can be that the end-to-end LS system effectively simplifies complex words. This measure is called recall, and it represents the ratio of the identified and replaced complex words in a text to all the complex words in the text. A high recall indicates that the system identifies and replaces most of the complex words. However, a high recall could be due to the system simplifying everything in the text, which is not necessarily desirable in the context of LS. Every substitution comes with a slight shift in meaning, which is why conservative lexical simplification approaches should be preferred. To ensure that the system correctly replaces the complex words, precision usually is used in conjunction with recall. Precision reflects the proportion of correctly replaced words to the total number of complex words. Unfortunately, the method of collecting and evaluating the evaluation dataset for this thesis does not facilitate a precision score. This is because the total count of complex words in the sentences is unknown; only the presence of a particular complex word is identified.

#### **Synonymous replacements**

When an LS system replaces a word for another, it is important that the meaning of the original sentence is preserved. To be able to check whether the systems replacements are suitable in the context of the original sentence the evaluation dataset contains human-annotated synonyms that are good substitutions for the complex word. If a complex word substitution results in replacing a word with one of these synonyms, the system has successfully preserved the meaning of the text. The share of synonymous replacements indicates how well the system is at finding synonymous words given a target word. The dictionary form, or word lemma, was used in all word comparisons.

#### **Synonymous replacements with higher corpus frequency**

In addition to preserving the meaning of the text, it is also desirable for the LS system to propose words that are more comprehensible than the original word. This measure aims to evaluate the LS system's ability to replace complex words with synonymous words that are more frequent in everyday language. The assumption is, as discussed in Section 2.2, that corpus frequency is a good estimate of word complexity. Corpus frequency has also the benefit of being very easy to implement as a word complexity estimate, which is why frequency represents simplicity in this dataset. The measure is calculated by determining how many of the complex words are substituted with a word that is both synonymous *and* more frequent than the original word. This measure indicates how well the system is able to select appropriate replacements that both are synonyms and easier to understand.

## 5 Results

This chapter will present the performance of the developed LS systems and the performance of the Random Forest Classifier used for CWI and substitution ranking. As described in Chapter 4 the LäsBERT system was developed in two versions. The first version uses a BERT model fine-tuned with easy-to-read text used for Substitution generation (see Section 4.2). The other LäsBERT system utilized the BERT model developed by Malmsten et al. (2020) without any fine-tuning. This version serves as a baseline to evaluate the effects of the fine-tuning of the substitution generation model on the end performance of the system.

### 5.1 Performance of the Random Forest Classifier

Table 5.1: The precision, recall, and F1-score of the RFC used for CWI. The support column represents the distribution of classes in the test set.

Class	Precision	Recall	F1-score	Support
<b>1</b>	0.63	0.73	0.67	154
<b>2</b>	0.35	0.28	0.31	107
<b>3</b>	0.59	0.65	0.62	103
<b>4</b>	0.54	0.42	0.47	60
<b>Weighted Avg:</b>	0.54	0.55	0.54	$\Sigma = 424$

Table 5.2: The accuracy of the RFC used for CWI.

Accuracy	Baseline (Most Frequent Class)
<b>0.55</b>	0.36

The RFC used for Complex word identification (CWI) and substitution ranking was tested on 424 out of 4238 words in the CWI dataset (see Section 3.1). Precision is the proportion of true positive samples out of all samples classified as a class **C**. An example from Table 5.1 is that the precision for class **1** is 0.63. This means that of all the words that the model classified as class **1**, 63% should be classified as **1**. Recall represents the proportion of the classified true positive samples out of all true positive samples of a class. An example from Table 5.1

is that class 2 had a recall score of 0.28. This means that out of all the samples belonging to class 2, 28% of the samples were correctly classified as class 2. The F1 score represents the harmonic mean of precision and recall. Accuracy (see Table 5.2) is the proportion of all correctly classified classes in the dataset, which in the case of this classifier was 0.55.

## 5.2 Perplexity of LäsBERT models on easy-to-read texts

In this Section, the perplexity scores of the masked language models in both versions of the LäsBERT LS system.

Table 5.3: The perplexity of the models on unseen part of the fine-tuning dataset.

	<b>Fine-tuned</b> KBLab/bert-base-swedish-cased	<b>Baseline</b> KBLab/bert-base-swedish-cased
Perplexity on unseen easy-to-read text	<b>4.58</b>	18.88

Table 5.3 represents the perplexity before and after fine-tuning the BERT models on the corpus with easy-to-read texts, indicating a significant decrease in perplexity and improved performance of the language model. As described in Section 4.2 perplexity represents the confidence in the model's predictions of a masked-out word.

## 5.3 Performance of LS-systems on the evaluation dataset

This section presents the performance of the developed Lexical Simplification systems. The evaluation measures used to evaluate the lexical simplifiers are explained in Section 4.3.

Table 5.4: How many times the systems identified and replaced the complex word in a sentence with another word. The best performance is highlighted using bold font.

	<b>LäsBERT</b> (baseline, not fine-tuned)	<b>LäsBERT</b> (fine-tuned)	<b>LäsGPT</b>
<b>Recall</b> (% of the complex words substituted for any word)	<b>53/150</b> <b>(35.3%)</b>	<b>53/150</b> <b>(35.3%)</b>	49/150 (32.7%)

Table 5.4 shows that the LäsBERT baseline system that hadn't been fine-tuned found and exchanged as many complex words as the fine-tuned LäsBERT system (35.3% of the complex words). They both found and replaced slightly more complex words than the LäsGPT system (32.7% of the complex words).

Table 5.5: Synonym replacements that resulted in the complex word being exchanged for a synonym in the dataset. The best performance is highlighted using bold font.

	<b>LäsBERT</b> (baseline, not fine-tuned)	<b>LäsBERT</b> (fine-tuned)	<b>LäsGPT</b>
<b>Synonymous replacements</b> (% of the <u>total</u> complex words resulting in a synonymous replacement)	14/150 (9.33%)	12/150 (8%)	<b>16/150</b> <b>(10.6%)</b>
<b>Synonymous replacements</b> (% of the <u>replaced</u> complex words resulting in a synonymous replacement)	14/53 (26.4%)	12/53 (22.6%)	<b>16/49</b> <b>(32.7%)</b>

Table 5.5 shows that the LäsBERT baseline system that hadn't been fine-tuned replaced complex words with words that were found in the dataset 9,33% of the time. The fine-tuned LäsBERT system replaced 8% of the complex words with a synonym included in a dataset. The LäsGPT system replaced 10.6% of the complex words with a synonym included in the dataset.

Table 5.6: Synonym replacements that resulted in the complex word being exchanged for a synonymous word that is more frequent than the original word. The best performance is highlighted using bold font.

	<b>LäsBERT</b> (baseline, not fine-tuned)	<b>LäsBERT</b> (fine-tuned)	<b>LäsGPT</b>
<b>Synonymous replacements with higher corpus frequency</b> (% of the <u>total</u> complex words that were replaced with a synonymous <i>and</i> more frequent word)	13/150 (8.7%)	11/150 (7.33%)	<b>15/150</b> <b>(10%)</b>
<b>Synonymous replacements with higher corpus frequency</b> (% of the <u>synonymous</u> replacements that resulted in a more frequent word)	13/14 (92.9%)	11/12 (91.7%)	<b>15/16</b> <b>(93.8%)</b>

Table 5.6 shows that the LäsBERT baseline system replaced complex words with synonyms found in the dataset that also were more frequent than the complex word 8.7% of the time. The fine-tuned LäsBERT system replaced 7.33% of the complex words with a more frequent synonym. The LäsGPT system replaced 10% of the complex words with a synonym in the dataset that was more frequent than the original word.

## 5.4 System-annotator agreement

The complexity of the evaluation dataset was assessed by humans (see Section 3.2). The complexity of each word was assessed by letting the annotators respond with TRUE or FALSE to the statement: *The complex word is actually a complex word.* As described in Table 3.3: 28.6% (43/150) of the words were marked as TRUE. This complexity assessment can be used to check if the systems replace the words that the human annotators regarded as complex. The overlap between the words that the human annotators marked as complex and the words that the system replaced indicates how good the systems are at replacing the words that humans perceive as complex.

Table 5.7: The proportion of words that the LS systems and the annotators marked as complex. The best performance is highlighted using bold font.

	<b>LäsBERT</b> (baseline, not fine-tuned)	<b>LäsBERT</b> (fine-tuned)	<b>LäsGPT</b>
<b>True Positives</b> (% of the complex words that were annotated as complex <i>and</i> replaced by the LS system)	<b>26/43</b> ( <b>60.5%</b> )	<b>26/43</b> ( <b>60.5%</b> )	22/43 (51.2%)
<b>True Negatives</b> (% of the complex words that were annotated as non-complex <i>and not</i> replaced by the LS system)	<b>80/107</b> ( <b>74.7%</b> )	79/107 (73.4%)	<b>80/107</b> ( <b>74.7%</b> )
<b>Total agreement</b>	<b>106/150</b> ( <b>70.1%</b> )	105/150 (70%)	102/150 (68%)

Table 5.7 describes that both LäsBERT versions replaced 60.5% of the words that were annotated as complex by the humans. LäsGPT scored lower and replaced 51.2% of the words annotated by humans as complex. LäsGPT and the baseline version of LäsBERT both agreed with the annotators on 74.4% of the words that were annotated as non-complex. The baseline version of LäsBERT had the highest overall agreement with the human annotators with 70.1% of the words being aligned with the human annotators.



## 6 Discussion

This chapter will mainly discuss the results presented in Chapter 5 and the method presented in Chapter 4. The chapter will also include a discussion of the collected evaluation dataset and its validity. The underlying premise of this thesis and a lot of other LS research will also be questioned: the premise that it is desirable to design systems that preemptively replace words in texts, rather than supporting the interaction between readers and texts in a more transparent manner.

### 6.1 System performance results

This section will discuss the results in Chapter 5. First, the performance of the Random Forest Classifier followed by the system performance on the evaluation dataset. The last part will discuss the effects of fine-tuning the BERT model on the end-to-end performance of the LäsBERT systems.

#### Random Forest Classifier performance

The performance of the Random Forest Classifier (RFC) used for CWI and substitution ranking is illustrated in Table 5.1 and 5.2. The average F1 score of 0.54 indicates a moderate performance of the RFC. The RFC developed by Smolenska (2018) with 39 word features had an F1 score of 0.769 and her RFC using 10 frequency features scored 0.773. The accuracy (the proportion of correctly classified words) of the RFC developed in this thesis was 0.55. The baseline for the accuracy measurement is classifying everything as the most frequent class, which in this case would have resulted in an accuracy score of 0.36 (see Table 5.2). This result also indicates moderate performance and does not quite reach the results of the RFC developed by Smolenska (2018) with a score of 0.747. The RFC in this thesis used 4 features (see Section 4.2), which is considerably fewer than the RFC:s used by Smolenska (2018) which could account for the poorer performance of the RFC system. The moderate performance of the RFC impacted the overall system performance, which will be highlighted in the following sections.

## LS System Performance

The results of the performance measure described in Section 4.3 are found in Section 5.3. The following subsections are going to discuss and summarize these results.

### Recall and System-annotator agreement

As shown in Table 5.4, 35.3% of the complex words were replaced by both LäsBERT systems, and 32.7% of the complex words are replaced by the LäsGPT system. The assumption is that all the words in the evaluation dataset are complex. Thus, identifying and replacing only one-third of complex words indicates that the systems could be in need of improvement. This highlights the significance of the CWI component in the LS pipeline, as it plays an important role in avoiding the omission of words that require simplification.

The systems are expected to have comparable recall rates since they are identical apart from the substitution generation subtask. The LäsGPT system did perform worse than the BERT-based systems. The LäsGPT system seems to generate substitutes that don't make it to the later stages of the LS pipeline. It is difficult to conclude if this is due to GPT generating more difficult or fewer substitutes. The BERT-based systems, with their instruction to generate 20 substitutes, had an inherent advantage in producing substitutes over the LäsGPT system which was conditioned to generate around 5 to 6 substitutes (see Appendix A). Generating more substitutions increases the likelihood that some substitutions successfully progress through the later stages of the pipeline. This advantage could explain their slightly higher recall performance compared to the LäsGPT system.

The human assessment of the evaluation dataset showed that just 28.6% of the words were actually regarded as complex. The overlap between the words annotated as complex in the human assessment and the complex words exchanged by the LS systems indicate how useful the systems would be upon implementation. As shown in Table 5.7 the LS systems agreed with the annotators between 68% (LäsGPT) and 70.1% (LäsBERT baseline) of the time. As described in Section 5.4 system-annotator agreement is constituted by the proportion of words that fulfill one of these conditions:

1. Words that are regarded as complex by the annotators *and* replaced by the LS system.
2. Words that are regarded as non-complex by the annotators *and not* replaced by the LS system.

The LäsBERT versions performed better than the LäsGPT version with an agreement of 60.5% for true positives (i.e. both humans and systems regarded the word as complex). This result is a bit more promising because it implies that the system and the human annotators in general agree on which words need to be replaced and which words are simple enough. The relatively poor performance on the evaluation dataset could possibly be linked to the complex words in the evaluation dataset not being complex enough, resulting in too few substitutions.

### Synonymous replacements

As shown in Table 5.5 LäsGPT performed the best when it came to replacing the complex words with synonyms (10.6%). A synonymous replacement rate of one-tenth out of all complex words must be considered weak. However, when considering only the complex words that were replaced, the rate of synonymous replacements improves to 32.7% in the LäsGPT system. The LäsBERT models demonstrate a lower percentage of synonymous substitutions,

with the baseline and fine-tuned versions yielding rates of 26.4% and 22.6%, respectively. This could be due to the fewer generated substitutes that GPT generates, which results in fewer replacements with higher quality. The architectural differences between the BERT and GPT models (see Section 2.6) might also explain some of the observed differences.

### Synonymous replacements with higher corpus frequencies

Table 5.6 shows that all synonymous replacements except for one resulted in a more frequent word. This was true for all developed LS systems. Higher corpus frequency is treated as analogous to lexical complexity, which means that LäsGPT had the biggest simplification effect on the texts with 10% of the words resulting in a word with higher corpus frequency. The difference between the models is small which limits the possibility of drawing reliable conclusions from the results.

### Summary of the LS system performance

In this section, the overall performance of the LS systems were analyzed, focusing on recall, synonymous replacements, replacements with higher corpus frequency, and system-annotator agreement. The performance of the RFC was also analyzed and compared to previous research done by Smolenska (2018).

The results revealed that both the LäsBERT and LäsGPT systems had relatively low recall rates, replacing only about one-third of the complex words in the evaluation dataset. This indicates the need for improvement in the systems' ability to identify and replace complex words accurately. The CWI component of the LS pipeline was highlighted as an area for future improvement. Regarding system-annotator agreement, the LS systems showed agreement with human annotators between 68% (LäsGPT) and 70.1% (LäsBERT baseline) of the time. The LäsBERT versions performed slightly better, with an agreement of 60.5% for true positives, indicating that the systems and human annotators generally agreed on which words needed to be replaced.

When it comes to synonymous replacements, LäsGPT performed the best, with a rate of 10.6% of complex words replaced by synonyms. However, when considering only the replaced complex words, the synonymous replacement rate improved to 32.7% for LäsGPT. The LäsBERT models demonstrated lower percentages of synonymous substitutions.

Furthermore, almost all synonymous replacements resulted in words with higher corpus frequencies, indicating a simplification effect. LäsGPT had a slightly bigger impact on text simplification, with 10% of the words resulting in a word with higher corpus frequency. While there is still potential for improvement, the relatively low perceived complexity of the complex words in the dataset and the more promising system-annotator agreement suggests that some issues are attributable to the dataset itself rather than to the LS systems.

### The effects of fine-tuning the BERT model

Both LäsBERT versions treated the substitution generation subtask as a Masked Language Modelling task using the Swedish BERT model developed by Malmsten et al. (2020). The difference between both versions was that one version used a BERT model fine-tuned on easy-to-read text, and the other used the out-of-the-box model from the Royal Library of Sweden huggingface repository <sup>1</sup>. The fine-tuned model was expected to be biased toward easier words and consequently generate easier substitutes. The decrease in perplexity from 18.88 to 4.58 (see Table 5.3) indicates that the model has improved in its ability to predict

<sup>1</sup><https://huggingface.co/KBLab/bert-base-swedish-cased>

words from the easy-to-read text. The underlying assumption is that this also leads to the model generating substitutes that are more readable.

The purpose of developing these two models was to investigate whether fine-tuning had any effect on the end-to-end performance of the Lexical Simplification system. As shown in Tables 5.4 and 5.7 the fine-tuning did not have any considerable effect on the number of words that were replaced by the model. Both versions identified and replaced 35.3% of the complex words and agreed 70.1% and 70% respectively of the time with the human annotators on which words were in need of simplification.

This result is somewhat surprising. The recall is expected to be similar in both versions which are identical except for the substitution generation subtask. However, the fact that they are exactly the same is surprising. The assumption is that the fine-tuned BERT model generates easier substitutes than the non-fine-tuned model. This should result in a lower average complexity level of the generated substitutes at the end of the LS pipeline, where the substitution ranking subtask ranks the remaining substitutes and the original complex word according to their simplicity. A word substitution is made if any of the ranked substitutes are *easier* than the original complex word (see rows (15) to (18) in Algorithm 4.1). The fine-tuned version that is supposed to produce easier words should fulfill this condition more often than the non-fine-tuned version. Therefore, some difference between the number of replacements is expected in favor of the fine-tuned model, which was not found in the evaluation. This result could be an effect of the relatively small size of the evaluation dataset, which limits the space where small effects can express themselves. To summarize: the quantity of the replacements wasn't affected by the fine-tuning.

If the quantity of the replacement wasn't affected by fine-tuning the BERT model, what about the quality of the replacements? There are two factors that determine the quality of a replacement: the level of synonymy and the simplicity of the replacement word. To determine these factors the evaluation dataset contained human-annotated synonyms and their corresponding corpus frequency. The corpus frequencies are treated as analogous to the word complexity. The number of synonymous replacements is similar for both versions: 14/150 for the baseline version and 12/150 for the fine-tuned version. Of all these synonymous replacements the new word had higher corpus frequencies 13/150 and 11/150 times (as seen in Table 5.5 and 5.6). In other words: almost every synonymous replacement also resulted in a simplifying replacement in both versions. Overall, the baseline model replaced the complex words with both synonymous words and simpler words more often than the fine-tuned model, even if the difference is small.

It was expected that the LS systems would occasionally replace the complex word with a replacement with higher corpus frequency than the original word, resulting in a replacement that made the text more difficult. This was expected to occur more often in the baseline version compared to the fine-tuned version, which was not the case. Both versions only made such an erroneous substitution once. There seems to be no real benefit of fine-tuning the BERT model for this implementation since the performance is worsened compared to the out-of-the-box model. The reason behind the reduced performance in the fine-tuned version remains unclear. A possibility is that the fine-tuning process had a detrimental impact on the model's overall language comprehension. The properties of the sentences (sentence lengths, passive/active voice, etc.) in the easy-to-read corpora might differ from sentences in the evaluation dataset too much, resulting in poor performance of the lexical simplification system on "ordinary" texts.

## Summary of the effects of fine-tuning

To summarize the found effects of fine-tuning the language model for substitution generation: fine-tuning did not affect the number of words replaced by the model on this evaluation dataset. Both versions performed similarly, identifying and replacing 35.3% of complex words and agreeing with human annotators 70.1% and 70% of the time, respectively. This lack of difference is hypothesized to be attributed to the small size of the evaluation dataset, limiting the expression of subtle effects. The evaluation also revealed that both versions had a very similar number of synonymous replacements, with next to all of these replacements also resulting in words with higher corpus frequency. Interestingly, the baseline version tended to make more synonymous and simpler replacements than the fine-tuned version. This indicates that it's not worth the effort to fine-tune the language model since it seems to have a detrimental rather than beneficial effect on the end-to-end performance. The reason behind the reduced performance of the fine-tuned version remains unclear. A possible explanation is that fine-tuning had an adverse impact on the model's overall language comprehension.

## 6.2 Method

There are almost a limitless number of combinations of different approaches, methods, sources of data, and models that could be implemented in the different subtasks of the Lexical Simplification pipeline. The LS systems developed in this thesis vary from each other in one distinct way: in the substitution generation subtask. The implementation of the LS systems builds upon a combination of earlier research, which has resulted in CWI systems for Swedish (Smolenska, 2018), resulted in language models that perform well on natural language understanding and generation tasks (Brown et al., 2020; J. Devlin et al., 2018; Malmsten et al., 2020; Rekathati, 2021) and resulted in adaptations of these for the specific purpose of LS (Qiang et al., 2021). The aim was to create an end-to-end lexical simplification system for Swedish<sup>2</sup>. The implementations consider the context of a complex word to generate replacement candidates that fit into the sentence of the complex word. The implementations take a conservative approach to LS, only substituting complex words when there is a measured simplification effect, resulting in fewer meaning-altering word substitutions without any reduction of complexity. This, in turn, affects the measurable recall negatively, since the systems are designed to avoid substituting words unnecessarily. However, the systems come with some limitations and room for improvements that could be implemented to develop the systems further.

### Limitations and possible improvements

Three factors are identified as areas of improvement and described in the subsections below:

#### Computation/context tradeoff

One tradeoff that affected the method used in this thesis is a computation/context tradeoff. Context in this case refers to the words surrounding a target complex words, which are used by the models to suggest fitting substitutes. Large transformer models such as BERT and GPT are computationally expensive to run, and a longer context involves more computation. While it would be preferable to feed whole texts or paragraphs into the language models to give them the best chance of suggesting fitting substitutes, this is not practically feasible with the available computation resources in this thesis. The LS systems developed in this thesis operate on a rather artificial sentence level, which however is much more preferable than operating on a word level (i.e. checking for synonyms in a thesaurus). The operation on a

<sup>2</sup>Note: This is not the first LS system for Swedish (see Section 2.5)

sentence level is artificial because it does not reflect how readers interact with texts naturally but still provides an approach that is somewhat context-aware.

### Cost and computation time

The cost and computation time is also a factor to consider when developing lexical simplification systems, even if it isn't a research question of this thesis. Running BERT is computationally expensive and therefore expensive if implemented on a large scale. The LS systems in this project were executed on a personal computer equipped with an 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz processor, as no additional computational resources were available for the running of the LS systems. Each LS system took over an hour to process the evaluation dataset (only 150 sentences), even when utilizing the OpenAI API for outsourcing the computations for substitution generation. However, the most computationally expensive step is probably the substitution selection step which utilizes sentence-BERT (Rekathati, 2021) to generate similarity scores between the original sentence and up to 20 alternative sentences. There are other approaches that could utilize more computationally effective methods, such as analyzing words instead of sentences. But as described in Section 4.2, there are benefits of comparing synonyms on a sentence level instead of a word level. Another way to reduce the computation time of the substitution ranking step is to reduce the number of generated alternatives that need to be compared to the original sentence, either by reducing the number of substitutes suggested by the models or filtering the substitutes more strictly.

### Random Forest Classifier performance

One limiting factor in this thesis was the performance of the Random Forest Classifier. As discussed in Section 6.1 the F1 score and accuracy of the RFC developed in this thesis did not match the RFC performance by Smolenska (2018). Since the RFC is implemented both in the CWI and substitution ranking steps of the LS pipeline the performance of the CWI is crucial for the overall system. Poor RFC performance results in the wrong words being simplified and the substitutes being ranked in the wrong order. Generating more word features could help in improving the performance, but as discussed in Section 4.2 this was not feasible for this thesis.

The CWI dataset used to train the RFC (Smolenska, 2018) may have been subject to limitations as well. To evaluate the dataset, only a sample comprising 4.7% of the words was assessed by two annotators. This assessment involved ordering one word from every complexity level and examining how well the human judgment corresponded to the actual complexity level. However, the relatively small evaluation sample and the approach of ordering words based on complexity rather than allowing annotators to assign complexity values freely introduce uncertainties regarding the extent to which the CWI dataset accurately reflects "true" complexity levels. A dataset reflecting more "true" complexity levels could have increased the system-annotator agreement in this thesis (see Table 5.7). The lack of validated data in NLP remains a challenge for smaller languages like Swedish.

## 6.3 The evaluation dataset

One of the key contributions of this thesis is the development of the evaluation dataset that is (to the authors' knowledge) the first of its kind for Swedish lexical simplification. Developing evaluation resources is especially important in smaller languages with limited data available. The collected dataset facilitates both faster development and the possibility for intrinsic evaluation of future Swedish lexical simplification systems. Without the need of evaluating the systems extrinsically, utilizing human evaluation, considerable time and resources could be

saved. Human evaluation of an LS system is still very relevant but can be postponed to later stages of development to maximize the performance of the systems.

Each quadruple in the automatically collected dataset was assessed by two different human annotators, once in the removal of unfit words, and once in the assessment phase. The dataset demonstrates an assessed high quality with 86.6% of the collected and manually checked synonyms in each quadruple being perceived as meaning-preserving substitutes for the complex words in the example sentences (see Table 3.3). The purpose of the double annotation was to create a dataset with a high level of quality and to outline the properties of the dataset (complexity, coverage, and quality). Moreover, the inclusion of word frequencies for both the complex words and the generated synonyms in each quadruple allows LS system developers to check easily whether a synonymous substitution also is a more frequent one.

However, there are also several important limitations to consider. One drawback is the scope of the dataset. Both the small size of the dataset (150 sentences) and the relatively small number of human annotators (1+3) involved in the assessment process are two significant limitations. With only three annotators, each with a university background and good lexical proficiency, there is a risk that what constitutes a "complex word" doesn't generalize well to audiences with reading difficulties. Increasing the number of annotators could provide a more comprehensive and reliable evaluation of the complexity levels. Involving members from target audiences with reading difficulties, e.g. people with Dyslexia and second language learners, would also ensure that the dataset reflects the needs of the intended users of an LS system.

Although the assessment indicated that the dataset has high coverage (72%), it is important to acknowledge that this number might be an overestimation (see Section 3.2). As mentioned in Lee et al. (2021) humans don't recall all possible word substitutes when only working from memory, therefore missing relevant synonyms might be omitted in the human assessment. Utilizing a thesaurus could be a way to expand the number of suitable synonyms and consequently improve the coverage of the evaluation dataset. Furthermore, the complexity level assessment of the dataset showed that only 28.6% of the complex words were actually regarded as complex by the annotators. This suggests that words without the need for simplification are present in the dataset, skewing the results of an LS system negatively.

Lastly, the annotators were tasked to provide binary responses of true or false to the statements listed in Section 3.2. This binary assessment reduced the nuances of complexity and contextual synonymy into two dimensions. An optimal evaluation of the dataset would have allowed for a more comprehensive assessment.

In conclusion, the evaluation dataset for lexical simplification has several advantages, most importantly being the first of its kind in Swedish, having all quadruples in the set being assessed by two human annotators, and containing word frequencies. However, possible improvements include increasing and diversifying the number of human annotators, increasing the coverage and complexity, and allowing for a more nuanced evaluation.

## 6.4 The work in a wider context

Lexical simplification can play an important role in improving the accessibility of texts. A modern information society relies on the written word to facilitate government services, occupations and communicate academic content. Readers with reading difficulties are disadvantaged if texts are overly difficult to comprehend. Therefore, the integration of lexical simplification techniques can contribute to bridging this gap by making information more accessible. By being able to automatically simplify texts across society individuals with read-

ing difficulties can better understand and engage more equally with many aspects of society. In a globalized world with more second-language learners than ever, lexical simplification can be a useful tool to make education, civil rights, and employment opportunities more accessible.

However, there are important risks to consider associated with inappropriately implemented lexical simplification systems. Sacrificing meaning for simplicity is detrimental to scientific research where nuanced, effective, and precise language is required to explain complex phenomena in the world. It is not certain that audiences with reading difficulties benefit from having academic texts simplified for them: an academic paper probably contains words that are more expressive than in other texts. Simplifying difficult texts could impede these audiences from learning the powerful words that allow communication to be efficient and precise. This thesis has built upon the assumption that frequent words are easier to understand, effectively limiting the preferred vocabulary to the most frequent words. Implementing this across society could result in the reduction of the Swedish language, the loss of linguistic variety, and consequently a loss of cultural diversity. The evolution and changeability of language is a natural process, but implementing systems that endorse one kind of language over another is problematic given the biases contained in the data the systems are trained on.

## 6.5 Moving beyond a questionable premise

The main premise of this thesis (and a lot of other LS research) is that the ultimate objective of lexical simplification (LS) research is the designing of a system capable of replacing complex words with simpler synonyms. However, it is worth considering whether the focus on hard word substitutions may be misguided. There is an argument to be made that another research objective is more fundamental: simplifying the interaction between readers and their text. This discussion is significant because it directs future research, including the formulation of the Lexical simplification task itself and the selection of appropriate evaluation metrics.

As discussed in Section 2.3, it's debatable whether strict synonymy exists at all. It is unlikely that even the most proficient human annotators can simplify texts through lexical substitution without altering the meaning of the original sentence. This raises questions about the feasibility of implementing an LS system with such capabilities. The challenge lies in the fact that each replaced word introduces an indisputable alteration of the meaning of the sentence, effectively obscuring the authors' intentions. A more transparent LS approach to aiding readers in understanding difficult texts is to provide the user with the *option* to access contextually relevant synonyms, glossaries, and word explanations when the reader finds some words difficult. The benefit of this approach is that it effectively eliminates the first and last stages in the LS pipeline, reducing the number of failure points in the LS system.

Instead of designing LS systems that replace complex words pre-interaction with substitutes that may or may not be synonymous with the original word, there should be a better approach: When a user experiences difficulties understanding a word, he/she activates the LS system on the difficult word. The LS system analyzes the context of the complex word and represents the sense of the target word. The LS system proceeds to generate suitable synonyms and selects the N most appropriate words that could fit as substitutions and presents them to the user. The user is presented with contextually appropriate synonyms, which hopefully correspond to words they are familiar with. Suggestions are generated, but no meaning-altering substitution is made.

This approach has the benefits of allowing users to actively engage with the LS system when encountering challenging words, rather than relying on pre-interaction replacements. This

eliminates the need for the CWI subtask<sup>3</sup>. An additional advantage is that the approach does not require producing a single replacement word; instead, it can generate a set of closely related words that aid the user in comprehending the complex word. This task should be easier than finding just one replacement. This eliminates the need for substitution ranking subtask in the LS pipeline. The LS task is now reduced and simplified to, a word sense disambiguation task, a substitution generation task, and a substitution filtering task.

Moreover, an opaque approach to word replacements may overlook the importance of developing readers' skills and strategies for navigating complex texts. Simply substituting difficult words with easier alternatives may provide a temporary solution, but it fails to empower readers with the necessary skills to engage with complex texts independently. This approach enhances the user's vocabulary and reading confidence by presenting contextually appropriate synonyms where needed - allowing the users to develop as readers.

This shift in focus towards a more user-empowering LS approach has significant implications for the selection of appropriate evaluation metrics. Traditional metrics, such as precision and recall, will not be as relevant in the proposed shift of focus. Instead, metrics that assess the user's vocabulary size and perceived reading confidence over time with the text should be considered. This is because traditional metrics assume that there is a gold standard for the meaning of a word. The development of the skills that the readers have is the goal of the approach presented, which makes the developed skills the appropriate subject of measurement. This probably applies more to second-language learners and schoolchildren rather than audiences with inherited reading difficulties, such as people with dyslexia and intellectual disabilities.

A skeptic of this view might argue that aiding the interaction between user and text is precisely what LS systems are already doing, suggesting that the proposed perspective view brings nothing new to the table. Replacing difficult words with easier ones already aids the user in their interaction, therefore no misguided premises are operated on. The response to this is that designing systems that alter the meaning of the original text does not aid the users' interaction, it aggravates it in the long term. The reigning premise fails to understand the complexity of synonymy which results in researchers devoting time to finding perfect synonyms that don't exist. Furthermore, it assumes that word complexity is a general property of a word and is not perceived differently by each individual.

Lexical simplification research is still useful, outlining the properties of complex words, developing new ways of generating substitutes, including contextual synonymy in the substitution selection task, etc. The end goal of lexical simplification needs to be altered to allow for a more user-centered lexical simplification. By fostering understanding, developing reading strategies, and preserving the integrity of texts, we can create a more empowering environment for readers, enabling them to engage with complex texts while promoting linguistic diversity and preserving the richness of language.

---

<sup>3</sup>Note: Research on CWI is still very useful for the field of ATS as a whole.



## 7

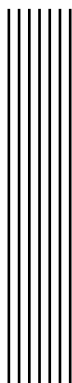
# Conclusion

**RQ1:** *How effective are the developed context-aware LS systems, LäsBERT and LäsGPT, in generating contextually appropriate word substitutions for Swedish lexical simplification?*

The lexical simplifiers developed for this thesis do not differ substantially in their performance from each other. The LäsBERT versions have a slightly higher recall, whilst LäsGPT performs slightly more synonymous replacements that also have a higher corpus frequency. The absolute percentage of the number of substitutions is not very high with around just a third of the complex words in the dataset being replaced by the LS systems. However, the agreement between the systems and annotators on which words should be substituted is relatively high (68% to 70.1%). There is room for improvement in the evaluation dataset: A higher proportion of perceived complex words is needed to more accurately reflect which words need to be simplified.

**RQ2:** *How does fine-tuning the BERT model on easy-to-read text affect the end-to-end performance of the LäsBERT lexical simplification system?*

The fine-tuning process did not have a noteworthy impact on the number of words replaced by the model. Both the fine-tuned and non-fine-tuned versions identified and replaced approximately 35.3% of complex words and had a similar agreement with human annotators. However, the evaluation revealed that the baseline version tended to make slightly more synonymous and simpler replacements compared to the fine-tuned version. This suggests that fine-tuning the model may not be beneficial and could potentially have a detrimental effect on the system's performance. The exact reason for the reduced performance of the fine-tuned version remains unclear, but it is speculated that the fine-tuning process may have negatively affected the model's overall language comprehension.



## Bibliography

- Abrahamsson, E., Forni, T., Skeppstedt, M., & Kvist, M. (2014). Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compound language. *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, 57–65.
- Abrahamsson, P. (2011). *Mer lättläst: Påbyggnad av ett automatiskt omskrivningsverktyg till lätt svenska* (Bachelor's Thesis, Linköpings Universitet).
- Begg, I., & Paivio, A. (1969). Concreteness and imagery in sentence meaning. *Journal of Verbal Learning and Verbal Behavior*, 8(6), 821–827.
- Bingel, J., & Bjerva, J. (2018). Cross-lingual complex word identification with multitask learning. *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, 166–174.
- Biran, O., Brody, S., & Elhadad, N. (2011). Putting it simply: A context-aware approach to lexical simplification. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 496–501.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: Analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."
- Borin, L., Forsberg, M., & Lönnngren, L. (2013). Saldo: A touch of yin to wordnet's yang. *Language resources and evaluation*, 47(4), 1191–1211.
- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5–32.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.
- Carroll, J., Minnen, G., Canning, Y., Devlin, S., & Tait, J. (1998). Practical simplification of english newspaper text to assist aphasic readers. *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, 7–10.
- Chen, S. F., Beeferman, D., & Rosenfeld, R. (1998). Evaluation metrics for language models.
- De Belder, J., & Moens, M.-F. (2010). Text simplification for children. *Proceedings of the SIGIR workshop on accessible search systems*, 19–26.
- Decker, A. (2003). Towards automatic grammatical simplification of swedish text. *Stockholm University Department of Linguistics Computational Linguistics*.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

- Devlin, S. (1998). The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic databases*.
- DuBay, W. H. (2004). The principles of readability. *Online Submission*.
- Ejerhed, E., Källgren, G., & Brodda, B. (2006). Stockholm-umeå corpus version 2.0. *Stockholm University, Dep. of Linguistics and Umeå University, Dep. of Linguistics*.
- Glavaš, G., & Štajner, S. (2015). Simplifying lexical simplification: Do we need simplified corpora? *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 63–68.
- Gooding, S., & Kochmar, E. (2019). Recursive context-aware lexical simplification. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4853–4863.
- Gooding, S., Kochmar, E., Yimam, S. M., & Biemann, C. (2021). Word complexity is in the eye of the beholder. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4439–4449.
- Harris, Z. S. (1970). *Discourse analysis*. Springer.
- Huggingface. (n.d.). Fine-tuning a masked language model [Accessed: 2023-04-24 from <https://huggingface.co/learn/nlp-course/chapter7/3?fw=tf>].
- Kann, V., & Rosell, M. (2006). Free construction of a free swedish dictionary of synonyms. *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA 2005)*, 105–110.
- Keskisärkkä, R. (2012). *Automatic text simplification via synonym replacement* (Bachelor's Thesis, Linköpings Universitet).
- Kilgarriff, A., Charalabopoulou, F., Gavrilidou, M., Johannessen, J. B., Khalil, S., Johansson Kokkinakis, S., Lew, R., Sharoff, S., Vadlapudi, R., & Volodina, E. (2014). Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48, 121–163.
- Knollman-Porter, K., Wallace, S. E., Hux, K., Brown, J., & Long, C. (2015). Reading experiences and use of supports by people with chronic aphasia. *Aphasiology*, 29(12), 1448–1472.
- Kremer, G., Erk, K., Padó, S., & Thater, S. (2014). What substitutes tell us—analysis of an “all-words” lexical substitution corpus. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 540–549.
- Lee, M., Donahue, C., Jia, R., Iyabor, A., & Liang, P. (2021). Swords: A benchmark for lexical substitution with improved data coverage and quality. *arXiv preprint arXiv:2106.04102*.
- Lin, D., Zhao, S., Qin, L., & Zhou, M. (2003). Identifying synonyms among distributionally similar words. *IJCAI*, 3, 1492–1493.
- Malmsten, M., Börjeson, L., & Haffenden, C. (2020). Playing with words at the national library of sweden – making a swedish bert.
- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from [tensorflow.org](https://www.tensorflow.org/)]. <https://www.tensorflow.org/>
- Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11), 39–41.
- Mühlenbock, K. (2008). Readable, legible or plain words—presentation of an easy-to-read swedish corpus. *Multilingualism: Proceedings of the 23rd Scandinavian Conference of Linguistics*, 8, 327–329.
- Myndigheten för tillgängliga medier. (n.d.). Lättläst [Accessed: 2023-04-21, <https://www.mtm.se/var-verksamhet/lattlast/>].

- Natur och Kultur. (n.d.). Rivstart [Accessed: 2023-04-21 from <https://www.nok.se/laromedel/serier/Rivstart/>].
- Paetzold, G., & Specia, L. (2016). Unsupervised lexical simplification for non-native speakers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).
- Paetzold, G. H., & Specia, L. (2017). A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60, 549–593.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. <https://doi.org/10.18653/v1/N18-1202>
- Pimienta Castillo, J. S. (2021). *Multilingual lexical simplification* (Master's Thesis, Universitat Pompeu Fabra).
- PLAIN. (2011). *Federal plain language guidelines*. CreateSpace Independent.
- Qiang, J., Li, Y., Zhu, Y., Yuan, Y., Shi, Y., & Wu, X. (2021). Lsbert: Lexical simplification based on bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 3064–3076.
- Quinlan, P. T. (1992). *The oxford psycholinguistic database*. Oxford University Press.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1), 5485–5551.
- Rekathati, F. (2021). The kblab blog: Introducing a swedish sentence transformer. <https://kblablab.github.io/posts/2021-08-23-a-swedish-sentence-transformer/>
- Rennes, E. (2022). *Automatic adaptation of swedish text for increased inclusion* (Doctoral dissertation). Linköping University Electronic Press.
- Rudell, A. P. (1993). Frequency of word usage and perceived word difficulty: Ratings of kučera and francis words. *Behavior Research Methods, Instruments, & Computers*, 25(4), 455–463.
- Rybing, J., & Smith, C. (2009). *Cogflux: Grunden till ett automatiskt textförenklingssystem för svenska* (Bachelor's Thesis, Linköpings Universitet).
- Shardlow, M. (2013). A comparison of techniques to automatically identify complex words. *51st annual meeting of the association for computational linguistics proceedings of the student research workshop*, 103–109.
- Shardlow, M. (2014). Out in the open: Finding and categorising errors in the lexical simplification pipeline. *LREC*, 1583–1590.
- Smolenska, G. (2018). *Complex word identification for swedish* (Master's Thesis, Uppsala Universitet).
- Språkbanken. (n.d.). Om oss [Accessed: 2023-04-21, Retrieved from: <https://spraakbanken.gu.se/om>].
- Stevenson, M., & Wilks, Y. (2003). Word sense disambiguation. *The Oxford handbook of computational linguistics*, 249, 249.
- Taylor, W. L. (1953). “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4), 415–433.
- Universitets och högskolerådet. (2023). Öva på gamla högskoleprov [Accessed: 2023-04-27 from <https://www.studera.nu/hogskoleprov/infor-hogskoleprovet/ova-pa-gamla-hogskoleprov/>].
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

- Volodina, E., & Kokkinakis, S. J. (2012). Introducing the swedish kelly-list, a new lexical e-resource for swedish. *LREC*, 1040–1046.
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.
- Wei, X., Peng, F., Tseng, H., Lu, Y., & Dumoulin, B. (2009). Context sensitive synonym discovery for web search queries. *Proceedings of the 18th ACM conference on Information and knowledge management*, 1585–1588.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., ... Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
- Yimam, S. M., Štajner, S., Riedl, M., & Biemann, C. (2017). Cwig3g2-complex word identification task across three text genres and two user groups. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 401–407.
- Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Proceedings of the IEEE international conference on computer vision*, 19–27.



## A GPT Prompt

This is the prompt used in the LäsGPT LS system. The instructions, task, and model completion varies with the sentence that is being simplified.

---

<b>Instructions:</b>	Ge förslag på enkla lämpliga substitutioner för ordet <b>**framsynthet**</b>
<b>Example 1:</b>	En integrerad stad är tvärtom blandad med en <b>**heterogen**</b> befolkningssammansättning i alla bostadsområden. alternativ: olikartad,blandad,brokig,oenhetlig,inhomogen,sammansatt
<b>Example 2:</b>	Skatten har lett till hårt motstånd från inflytelserika företrädare för näringslivet och regeringen ville därför <b>**revidera**</b> den. alternativ: ändra,omarbeta,bearbeta,förbättra,korrigera,modernisera,rätta
<b>Example 3:</b>	För det andra gav myten skarp ammunition till nordamerikanska forskare som önskade <b>**blottlägga**</b> sin egen stats smutsiga historia. alternativ: blotta,exponera,avhölja,avkläda,barlägga,synliggöra
<b>Example 4:</b>	Det är ett misstag att presentera föremålen isolerade när deras <b>**signifikans**</b> kan förstås först när man ser helheten. alternativ: betydelse,vikt,betydenhet,värde,angelägenhet,relevans
<b>Example 5:</b>	Sajterna anklagades för att sprida pornografi och annat material som kan <b>**fördärva**</b> ungdomar. alternativ: "förstöra,skada,försämra,korrumpera,sabba
<b>Task</b>	Tack vare hans mod och <b>**framsynthet**</b> får miljoner fler äta sig mätta idag. Alternativ:
<b>Model completion:</b>	visdom,insikt,förutseende,förmåga,skarpsinne,intelligens