Context-aware Swedish Lexical Simplification

Emil Graichen
Department of Computer and Information Science
Linköping University
Linköping, Sweden
emigr679@student.liu.se

Arne Jönsson
Department of Computer and Information Science
Linköping University
Linköping, Sweden
arne.jonsson@liu.se

Abstract

We present results from the development and evaluation of context-aware Lexical simplification (LS) systems for the Swedish language. 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.

1 Introduction

Lexical simplification (LS) is the task of replacing complex words with easier ones. The approaches to this task usually involve replacing words with simpler synonyms found in a linguistic database (Devlin, 1998; Gooding and Kochmar, 2019; Rennes, 2022), implementing rules to "translate" linguistic units into easier ones (Zhu et al., 2010; Coster and Kauchak, 2011), or using word embeddings to generate similar substitution candidates (Glavaš and Štajner, 2015; Gooding and Kochmar, 2019). As mentioned in Qiang et al. (2021) these methods usually fail to take the context of the target word into account, resulting in nonsensical substitutions.

Recently, with the introduction of large-scale pre-trained transformer language models such as BERT (Devlin et al., 2018) and GPT-3 (Brown et al., 2020) a new chapter in the field of Natural Language Processing (NLP) has begun. GPT-3 and BERT perform well on a broad set of downstream NLP tasks (Brown et al., 2020; Devlin et al., 2018). BERT has already been implemented in LS systems for English (Qiang et al., 2021; Li et al., 2022), Portuguese (North et al., 2022), Spanish, and German (Pimienta Castillo, 2021). The benefit of using these models, trained on vast amounts of text, over conventional methods is that these models can generate more contextually appropriate substitutions for complex words, which is reflected in the high performance of these systems (Li et al., 2022; Qiang et al., 2021; Saggion et al., 2022). The TSAR-2022 Shared Task (Saggion et al., 2022) demonstrated that BERT-based LS systems were surpassed only by a GPT-3 based method (Aumiller and Gertz, 2022), highlighting the effectiveness of large-scale pre-trained transformers in the realm of Lexical simplification.

In this paper, three versions of a Swedish LS system are presented: 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 uses a BERT model fine-tuned on easy-to-read-text and one uses an out-of-the-box model to investigate how fine-tuning the BERT model affects the end-to-end performance of the LS system. 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 newly collected evaluation dataset.

2 Lexical Simplification

Shardlow (2014) and Paetzold and Specia (2016) described the general pipeline of Lexical Simplification: *Complex Word Identification* (CWI) which aims to find candidates in need of simplification and *Substitution Generation* (SG) which describes the process of generating alternative words to the identified complex words. The most synonymous generated alternatives are selected in the next step conveniently named *Substitution Selection*. Finally, in the *Substitution ranking* task, the remaining words are ranked according to simplicity, where

Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability associated with RANLP-2023, pages 11–20,
held in Varna Bulgaria, Sept 7, 2023.
https://doi.org/10.26615/978-954-452-086-1_002

the simplest word is chosen as the final word substitute.

2.1 Complex Word Identification

Shardlow (2013) showed that the performance of an overall LS system is dependent on the performance of the CWI component. 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.

2.2 Substitution Generation

There are several approaches to SG 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), Abrahamsson et al. (2014), and Abrahamsson (2011) used the Swedish synonym dictionary SynLex (Kann and Rosell, 2006) 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 (Gooding and Kochmar, 2019; Devlin, 1998; De Belder and Moens, 2010). 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 (Rennes, 2022; Glavaš and Štajner, 2015; Paetzold and Specia, 2016). 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.

Using pre-trained encoders such as BERT to reformulate SG into a Masked Language Modelling (MLM) task has been done to avoid the problem of disregarding context in LS tasks (Qiang et al., 2021; Pimienta Castillo, 2021; North et al., 2022). The method 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. These words are then treated as replacement candidates for the complex word (Qiang et al., 2021).

SG has also been reformulated as a language generation task (Lee et al., 2021; Aumiller and Gertz, 2022). To generate suitable alternatives to specific words in a short paragraph they utilised the in-context learning abilities of GPT-3 (Brown et al., 2020) to generate suitable substitutions.

2.3 Substitution Selection and Ranking

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 utilised 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 in a corpus was evaluated. If the replacement word together with its right or left neighbour formed a bigram that didn't occur once in the corpus it was assumed that the bigram was ungrammatical, and thus removed (Gooding and 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 (Qiang et al., 2021; Keskisärkkä, 2012). Ranking synonyms exclusively based on the number of characters in a word has also been proposed, but this approach has some considerable limitations (Abrahamsson, 2011).

3 Data used in our studies

Various data sources were 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 for RFC training

The following resources were used to train the RFC for CWI:

The Stockholm-Umeå Corpus (SUC) is a balanced corpus collected in the nineties with annotated POS tags, morphological features, and lemmas. The corpus' token frequencies, were used to train the RFC for CWI. The version used in this paper is SUCX 3.0[1] which is free to use without a license (Ejerhed et al., 2006).

Språkbanken hosts corpora from blogs[2] and Twitter[3]. The blogs were selected from the top lists of *bloggportalen.se*[4] a Swedish homepage hosting blogs on various topics, and the Twitter posts were sourced from a selection of Swedish Twitter users. The statistics data sheets for both the BloggMix and the TwitterMix corpora were also used, which included token frequency, lemma, and POS tags for each token. Smolenska (2018) determined that word frequencies in blog corpora are highly informative for predicting complex words. Therefore, the BloggMix corpus served as the main source of word frequencies for training the RFC. However, it was also used to construct the evaluation dataset.

Smolenska (2018) collected a dataset of 4,238 words derived from Rivstart dictionaries (Natur och Kultur), 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 *Common European Framework of Reference for Language* (CEFR) scores. These scores, taking the values **A1** (novice)**, A2, B1, B2, C1**, and **C2** (proficient), correspond to language proficiency levels (Volodina and Kokkinakis, 2012). These six categories, **A1** to **C2**, of sourced words, were grouped into three, and a fourth group was added containing the most complex words. The words in the fourth group were sourced from Ordtestet[5], a website that targets native Swedish speakers, where users can test their understanding of difficult words.

3.2 Linguistic Resources used for fine-tuning

Two corpora containing easy-to-read text were used to fine-tune the BERT model in one of the `LäsBERT` versions. 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). The 8sidor corpus contains over 420 000 sentences and over 4.5 million tokens[6]. LäSBarT is a corpus containing easy-to-read texts sourced from children's books. The corpus contains a little over 100,000 sentences and 1 million tokens (Mühlenbock, 2008)[7].

3.3 Linguistic Resources used for the evaluation dataset

The Kelly Swedish List (Volodina and Kokkinakis, 2012; Kilgarriff et al., 2014) is a lexical resource with over 8,000 Swedish word lemmas annotated with Word frequencies, word classes, and CEFR scores. All **C1** and **C2** words in the Swedish Kelly list[8], i.e. words that were assumed to be complex, were sourced for the evaluation dataset.

SynLex (Kann and 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 were 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[9] only synonyms that were rated at the synonymy level of 3 or higher were included which amounted to 38,000 word-pairs.

SALDO is a Swedish lexical-semantic resource developed by Borin et al. (2013) containing word relations and their senses. The resource includes a lexicon where the words in SALDO are put into

[1] https://spraakbanken.gu.se/resurser/sucx3

[2] https://spraakbanken.gu.se/resurser/bloggmix

[3] https://spraakbanken.gu.se/resurser/twitter

[4] https://www.bloggportalen.se

[5] https://ord.relaynode.info/

[6] https://spraakbanken.gu.se/resurser/attasidor

[7] https://spraakbanken.gu.se/resurser/lasbart

[8] https://spraakbanken.gu.se/resurser/kelly

[9] http://folkets-lexikon.csc.kth.se/lexikon/synpairs.xml

an example sentence[10]. These example sentences, the complex words from the Kelly Swedish List, and the SynLex synonyms were used to create the evaluation dataset.

4 Creating the evaluation dataset

The evaluation dataset was collected automatically and evaluated manually. 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 assumed to be complex words. The corpus frequency of these words in the BloggMix corpus was retrieved. Following this, all available synonyms to the retrieved words were saved from the SynLex dictionary. The corpus frequencies of these synonyms were also saved. The final step was to find an example sentence in SALDO where the complex word occurred, resulting in 185 quadruples consisting of a complex word, its corpus frequency, a dictionary of suitable synonyms, and an example sentence. After a manual annotation process, nonsensical quadruples were removed, leaving a total of 150 quadruples.

Three native Swedish student annotators were enlisted to evaluate the dataset. 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. Student annotators from Linköping University's Cognitive Science Bachelors program were recruited. Two online versions of the Swedish academic aptitude test, *Högskoleprovet*, (Universitets och högskolerådet, 2023) were used to assess their word knowledge. The combined maximum score was 40 and the annotators scored 37, 33, and 35 respectively, indicating their strong lexical proficiency.

Each annotator got 50 separate quadruples to evaluate to ensure that all of the 150 quadruples in the dataset were human-annotated once. The annotators answered three questions regarding the *quality*, *coverage*, and *complexity* with "True" or "False" for each quadruple.

The results (see Table 1) 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). For 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 higher than the *actual* coverage. This has the possible effect of artificially limiting the score that a Lexical Simplification system can achieve on the dataset since valid substitutions could be missing in the set of correct alternatives. The annotators did generally not think that the words sourced from the Kelly Swedish List were complex, with only 28,6% of quadruples being annotated as complex. However, since the annotators were native Swedish speakers with a university education, the perception of what constitutes a complex word might not generalise well to audiences with reading difficulties. The dataset is freely available at `https://github.com/emilgraichen/SwedishLSdataset`.

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

Table 1: Percent of quadruples annotated with "True" in response to the statements regarding *Quality, Coverage*, and *Complexity.*

5 Method

In this section, we will describe the implementation and evaluation of three LS systems, each varying only in the substitution generation subtask. The developed systems are two BERT-based versions of an LS system called `LäsBERT`, and one version of a GPT-3 based LS system called `LäsGPT`. The structure of this section is based on the general pipeline of other LS systems described in Section 2.

5.1 Complex Word Identification

As described in Section 2.1, frequency features can be treated as 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) utilising the Swedish complex word dataset developed

[10] `https://spraakbanken.gu.se/resurser/saldoe`

by Smolenska (2018). The RFC was trained on a dataset containing four (4) word features and outputs a word complexity score between 1-4. In this implementation, the features that the RFC used were the word's corpus frequency in the BloggMix, TwitterMix, and SUCX 3.0 corpora together with the length of the word. The corpus frequencies were normalised by computing the common logarithm of the absolute frequency. This normalisation method yielded the best results in earlier work (Smolenska, 2018). The RFC training dataset was split into 90% training data and 10% test data (see Table 3 for the classifier performance).

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 (Bingel and Bjerva, 2018). The number of characters in each word was therefore used as the last feature for Complex Word Identification (CWI).

To implement the trained RFC in the LS pipeline the first step involved splitting the input sentence into individual words, because the RFC operates on a word-by-word basis. All non-alphanumerical characters in the sentence were also removed. 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.

Every word in the input sentence was then classified by the trained RFC from "1" to "4". Words scored with "1" or "2" were treated as non-complex and scores of "3" or "4" were sent further down the pipeline for simplification.

5.2 Substitute Generation

Two versions of `LäsBERT` were developed. The first version used a fine-tuned KB-BERT model[11], 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 KB-BERT 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 was developed by (Qiang et al., 2021) for English and was in this paper adapted 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 a `[MASK]` token and fed into the model. The rationale behind feeding the original sentence into the model twice is that it forces the model to consider the meaning of the complex word when generating substitutes. A probability distribution was returned with substitutes and their corresponding probability, in this case, the BERT models generated 20 alternatives. These alternatives are generated based on the left and right context of the masked-out word. This should handle the problem that some of the conventional approaches face; that words generated are not synonyms in all contexts.

The out-of-the-box KB-BERT model is trained on text data from different sources and time periods to be representative 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 when predicting masked-out complex words. To accomplish this the KB-BERT model was fine-tuned on the LäsBarT and 8sidor corpora which contain easy-to-read texts. The huggingface tutorial[12] (Huggingface) to adapt masked language models to domain-specific data was adapted to the easy-to-read corpora and the KB-BERT model.

The fine-tuning of the BERT model in one of the `LäsBERT` versions began with creating a fine-tuning dataset with the words from the 8sidor and LäSBarT corpora and concatenating them into sentences. These sentences 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 was used to

[11] https://huggingface.co/KBLab/bert-base-swedish-cased

[12] https://huggingface.co/learn/nlp-course/chapter7/3?fw=tf

test the perplexity of the model, which is a measure of the model's (un)certainty 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 2, indicating a significant decrease in perplexity and improved performance of the language model.

	Fine-tuned KBLab/BERT	**Not Fine-tuned** KBLab/BERT
Perplexity	**4.58**	18.88

Table 2: The perplexity of the models on unseen part of the fine-tuning dataset.

`LäsGPT` utilised OpenAI:s GPT 3.5 `text-davinci-003` model[13] 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 but with an adaptation for Swedish. Except for the max_token parameter, the same parameters used in Lee et al. (2021) were used in this implementation. GPT-3 was prompted to generate around six alternatives for each word.

5.3 Substitute Filtering and Selection

The generated words for all LS systems needed to be filtered to remove substitutions that were not appropriate. A basic criterion for synonymy is that two words have the same Part-of-speech (POS) tag. Therefore, the POS tag for both the generated alternatives and the complex word was retrieved from the SUCX 3.0 corpus. If the POS tags did not match, the alternative was removed. If the generated token was empty or incomplete, it was removed as well.

The KB-SENTENCE-BERT model maps sentences to a 768-dimensional vector space (Rekathati, 2021), in contrast to the KB-BERT models which work on a word level. This facilitates comparison between sentences by calculating the cosine distance between their vector representations. 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 and filtered substitutes in the original sentence. By examining the similarity of sentences rather than comparing individual words using a thesaurus or word embeddings, the meaning-preserving effect on the bigger linguistic unit is taken into account, thus minimising the likelihood of substitutions that are inappropriate in the context.

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.

5.4 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 to 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 were generated for the selected substitutes and the original complex word. The RFC used in the CWI subtask 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 minimise 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. The baseline version of `LäsBERT` is available at `https://github.com/emilgraichen/SwedishLexicalSimplifier`.

[13] `https://platform.openai.com/docs/models/gpt-3-5`

6 Results

The performance of the Random Forest Classifier is presented in Table 3. The RFC used for Complex word identification (CWI) and substitution ranking was tested on 424 out of 4238 words in the CWI dataset.

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

Table 3: The precision, recall, and F1-score of the RFC used for CWI. The support column represents the distribution of classes in the test set.

Accuracy is the proportion of all correctly classified classes in the dataset, which in the case of this classifier was 0.55.

	LäsBERT (baseline)	**LäsBERT** (fine-tuned)	**LäsGPT**
Recall	**53/150 (35.3%)**	**53/150 (35.3%)**	49/150 (32.7%)

Table 4: Number of complex words substituted for any word. Bold font highlights the best performance.

Table 4 shows that the LäsBERT baseline system that had not 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).

Synonymous replacements	**LäsBERT** (baseline)	**LäsBERT** (fine-tuned)	**LäsGPT**
total complex words	14/150 (9.33%)	12/150 (8%)	**16/150 (10.6%)**
replaced complex words	14/53 (26.4%)	12/53 (22.6%)	**16/49 (32.7%)**

Table 5: Replacements that resulted in the complex word being exchanged for a synonym in the dataset. Bold font highlights the best performance.

Table 5 shows that the LäsBERT baseline system that had not 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.

Replacements	**LäsBERT** (baseline)	**LäsBERT** (fine-tuned)	**LäsGPT**
total complex words replaced with a synonymous *and* more frequent word	13/150 (8.7%)	11/150 (7.33%)	**15/150 (10%)**
synonymous that resulted in a more frequent word	13/14 (92.9%)	11/12 (91.7%)	**15/16 (93.8%)**

Table 6: Replacements that exchanged the complex word with a synonymous *and* more frequent word. Bold font highlights the best performance.

Table 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.

	LäsBERT (baseline)	**LäsBERT** (fine-tuned)	**LäsGPT**
True Positive (annotated as complex *and* replaced)	**26/43 (60.5%)**	**26/43 (60.5%)**	22/43 (51.2%)
True Negative (annotated as non-complex *and* **not** replaced)	**80/107 (74.7%)**	79/107 (73.4%)	**80/107 (74.7%)**
Total agreement	**106/150 (70.1%)**	105/150 (70%)	102/150 (68%)

Table 7: The proportion of words that the LS systems and the annotators marked as complex. Bold font highlights the best performance.

The results in Table 7, reflect the system-annotator agreement. If a complex word in the dataset evaluation, see Section 4, was regarded as complex by the annotators *and* replaced by an LS system at test time it counted towards the True Positive score. If annotators marked the words as non-complex *and* the LS systems didn't replace the word with anything it counted towards the True Negative score.

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.

7 Discussion

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 language model for substitution generation 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 assumed 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.

8 Conclusion

The lexical simplifiers presented in this paper 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 of the evaluation dataset. A higher proportion of perceived complex words is needed to more accurately reflect which words need to be simplified.

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 may be that the fine-tuning process have negatively affected the model's

overall language comprehension.

Lay Summary

Lexical simplification is the task of replacing complex words with easier ones. The approaches to this task usually involve replacing words with simpler synonyms found in a linguistic database, implementing rules to ”translate” linguistic units into easier ones, or using language models to generate similar substitution candidates. These methods usually fail to take the context of the target word into account, resulting in nonsensical substitutions.

We present results from the development and evaluation of context-aware Lexical simplification systems for the Swedish language. Three versions of lexical simplification models were created and evaluated on a newly constructed Swedish evaluation dataset. The simplification 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.

References

Emil Abrahamsson, Timothy Forni, Maria Skeppstedt, and Maria Kvist. 2014. Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 57–65.

Peder Abrahamsson. 2011. Mer lättläst: Påbyggnad av ett automatiskt omskrivningsverktyg till lätt svenska. Bachelor's thesis, linköpings universitet, Linköping University.

Dennis Aumiller and Michael Gertz. 2022. UniHD at TSAR-2022 shared task: Is compute all we need for lexical simplification? In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 251–258, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Joachim Bingel and Johannes Bjerva. 2018. Cross-lingual complex word identification with multitask learning. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 166–174.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. ” O'Reilly Media, Inc.”.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. Saldo: a touch of yin to wordnet's yang. *Language resources and evaluation*, 47(4):1191–1211.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models.

Will Coster and David Kauchak. 2011. Learning to simplify sentences using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Prroceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM; New York.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Siobhan Devlin. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic databases*.

Eva Ejerhed, Gunnel Källgren, and Benny Brodda. 2006. Stockholm-umeå corpus version 2.0. *Stockholm University, Dep. of Linguistics and Umeå University, Dep. of Linguistics*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *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)*, pages 63–68.

Sian Gooding and Ekaterina Kochmar. 2019. Recursive context-aware lexical simplification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4853–4863.

Huggingface. Fine-tuning a masked manguage model. https://huggingface.co/learn/nlp-course/chapter7/3?fw=tf. Accessed: 2023-04-24 from https://huggingface.co/learn/nlp-course/chapter7/3?fw=tf.

Viggo Kann and Magnus Rosell. 2006. Free construction of a free swedish dictionary of synonyms. In *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA 2005)*, pages 105–110.

Robin Keskisärkkä. 2012. Automatic text simplification via synonym replacement. Bachelor's thesis, linköpings universitet, Linköping University.

Adam Kilgarriff, Frieda Charalabopoulou, Maria Gavrilidou, Janne Bondi Johannessen, Saussan Khalil, Sofie Johansson Kokkinakis, Robert Lew, Serge Sharoff, Ravikiran Vadlapudi, and Elena Volodina. 2014. Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48:121–163.

Mina Lee, Chris Donahue, Robin Jia, Alexander Iyabor, and Percy Liang. 2021. Swords: A benchmark for lexical substitution with improved data coverage and quality. *arXiv preprint arXiv:2106.04102*.

Xiaofei Li, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. Mantis at tsar-2022 shared task: Improved unsupervised lexical simplification with pretrained encoders. *arXiv preprint arXiv:2212.09855*.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with Words at the National Library of Sweden – Making a Swedish BERT. *arXiv preprint arXiv:2007.01658*.

Katarina Mühlenbock. 2008. Readable, legible or plain words–presentation of an easy-to-read swedish corpus. In *Multilingualism: Proceedings of the 23rd Scandinavian Conference of Linguistics*, volume 8, pages 327–329. Acta Universitatis Upsaliensis Uppsala, Sweden.

Myndigheten för tillgängliga medier. Lättläst. https://www.mtm.se/var-verksamhet/lattlast/. Accessed: 2023-04-21, https://www.mtm.se/var-verksamhet/lattlast/.

Natur och Kultur. Rivstart. https://www.nok.se/laromedel/serier/Rivstart/. Accessed: 2023-04-21 from https://www.nok.se/laromedel/serier/Rivstart/.

Kai North, Marcos Zampieri, and Tharindu Ranasinghe. 2022. Alexsis-pt: A new resource for portuguese lexical simplification. *arXiv preprint arXiv:2209.09034*.

Gustavo Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jorge S Pimienta Castillo. 2021. Multilingual lexical simplification. Master's thesis, Universitat Pompeu Fabra.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, Yang Shi, and Xindong Wu. 2021. Lsbert: Lexical simplification based on bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3064–3076.

Faton Rekathati. 2021. The kblab blog: Introducing a swedish sentence transformer.

Evelina Rennes. 2022. *Automatic Adaptation of Swedish Text for Increased Inclusion*. Ph.D. thesis, Linköping University Electronic Press.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the TSAR-2022 shared task on multilingual lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 271–283, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st annual meeting of the association for computational linguistics proceedings of the student research workshop*, pages 103–109.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *LREC*, pages 1583–1590.

Greta Smolenska. 2018. Complex word identification for swedish. Master's thesis, Uppsala Universitet.

Universitets och högskolerådet. 2023. Öva på gamla högskoleprov. https://www.studera.nu/hogskoleprov/infor-hogskoleprovet/ova-pa-gamla-hogskoleprov/. Accessed: 2023-04-27 from https://www.studera.nu/hogskoleprov/infor-hogskoleprovet/ova-pa-gamla-hogskoleprov/.

Elena Volodina and Sofie Johansson Kokkinakis. 2012. Introducing the swedish kelly-list, a new lexical e-resource for swedish. In *LREC*, pages 1040–1046.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.