Computational Linguistics in the Netherlands Journal 13 (2024) 31-61 Submitted 02/2024; Published 03/2024

# Controllable Sentence Simplification in Dutch

**Theresa Seidl**[*] there.seidl@gmail.com
**Vincent Vandeghinste**[*,**] vincent@ccl.kuleuven.be

[*] *KU Leuven, Belgium*

[**] *Instituut voor de Nederlandse Taal, Leiden, the Netherlands*

## Abstract

Text simplification aims to reduce complexity in vocabulary and syntax, enhancing the readability and comprehension of text. This paper presents a supervised sentence simplification approach for Dutch using a pre-trained large language model (T5). Given the absence of a parallel corpus in Dutch, a synthetic dataset is generated from established parallel corpora. The implementation[1] incorporates a sentence-level discrete parametrization mechanism, enabling control over the simplification features. The model's output can be tailored to different simplification scenarios and target audiences by incorporating control tokens into the training data. The controlled attributes include sentence length, word length, paraphrasing, and lexical and syntactic complexity.

This work contributes a dedicated set of control tokens tailored to the Dutch language. It shows that significant simplification can be achieved using a synthetic dataset with as few as 2000 parallel rows, although optimal performance requires a minimum of 10,000 rows. The fine-tuned model achieves a 36.85 SARI score on the test set, supporting its effectiveness in the simplification process.

This research contributes to the field of sentence simplification by discussing the implementation of a supervised simplification approach for Dutch. The findings highlight the potential of synthetic datasets and control tokens in achieving effective simplification, despite the lack of a parallel corpus in the target language.

## 1. Introduction

Text simplification refers to complexity reduction in vocabulary and sentence structure to make text more readable and understandable while maintaining its information and meaning. From a linguistic viewpoint, simplification comprises the lexical part, where complex words are replaced by simpler versions, and the syntactic aspect, where the sentence structure is adjusted to reduce complexity. Text simplification refers to simplifying texts on a paragraph level for easier reading (Alva-Manchego et al. 2020a). Sentence simplification deals with simplification on a sentence level "where the input of the model is a single source sentence, and the output can be composed of one sentence or split into multiple" (Martin et al. 2020) to produce a simpler output. Frequently, the terms *text simplification* and *sentence simplification* are used interchangeably (Stajner 2021). Papers that regard text simplification specifically on phrase or text level are scarce (Glavaš and Štajner 2013, Laban et al. 2021, Narayan and Gardent 2014), and most work focuses on sentence simplification (Alva-Manchego et al. 2020a, Martin et al. 2022).

Among the targeted audiences for simplification applications are addressees with cognitive impairments (Carroll et al. 1998, Rello et al. 2013), low reading skills (Aluisio et al. 2008, Watanabe et al. 2009), second language learners (Lee and Yeung 2018), or speakers where the target language is not the mother tongue (Paetzold and Specia 2016b). Likewise, simplification is applied as a preprocessing step in parsing (Chandrasekar et al. 1996), question generation (Heilman and Smith 2010), semantic role labeling (Vickrey and Koller 2008), or text augmentation (Sevens

1. Code is released under: `https://github.com/tsei902/simplify_dutch` and `https://huggingface.co/tsei902/simplify_dutch`

et al. 2018). Often, simplification is understood as a text-editing task, that comprises splitting (Narayan and Gardent 2014, Siddharthan 2006), deletion, and sentence compression (Févry and Phang 2018, Filippova and Strube 2008, Ghalandari et al. 2022), as well as paraphrasing (Dehghan et al. 2022, Maddela et al. 2021, Martin et al. 2022, Qiang et al. 2022, Specia 2010, Wubben et al. 2012) to modify sentences toward a specific target.

In this paper, we focus on sentence simplification in Dutch. More specifically, this work aims to clarify the following questions: "*Can a synthetically generated dataset be used in the absence of a Dutch parallel corpus? What is a suitable set of control tokens for Dutch language? To which extent are these control tokens valuable to steer the generation of simplified sentences?*"

To do so, we adapt a sentence-level transfer learning approach based on a text-to-text transformer model (Raffel et al. 2020) trained in a supervised fashion on Dutch textual data. By adding a control mechanism (Martin et al. 2020), the model output is adjustable regarding sentence length, word length, the amount of paraphrasing, and lexical and syntactic complexity. Additionally, we explore the further adaptation of the generated output at decoding time. Our contributions are the following: We develop an end-to-end simplification approach with a model trained on the simplification task, we provide a synthetically generated parallel dataset, and we propose a first set of suitable control token values for sentence simplification in Dutch.

This work is structured as follows: Section 2 gives an overview of related work in sentence simplification. Chapter 3 introduces the data used. Chapter 4 introduces methods, model and the simplification control mechanism. Chapter 5 contains the details on model training, evaluation, hyperparameter search of training parameters, and control tokens, as well as on the conduction of experiments. We discuss the results in chapter 6 and conclude in chapter 7.

## 2. Related Work

Early work on simplification is rule-based (Carroll et al. 1998, Daelemans et al. 2004, Siddharthan et al. 2004) or enhanced with automatically induced rules (Vandeghinste and Pan 2004). Later work defines sentence simplification as a monolingual machine translation (MT) task, either on the phrase-level (Coster and Kauchak 2011, Narayan and Gardent 2014, Wubben et al. 2012) or on the syntax-level (Zhu et al. 2010). Numerous studies have been conducted on sentence simplification in various languages, such as Dutch (Bulté et al. 2018, Sevens et al. 2018), Japanese (Kajiwara and Komachi 2018, Maruyama and Yamamoto 2019), Brazilian Portuguese (Aluisio et al. 2008), Spanish (Drndarević and Saggion 2012, Martin et al. 2020), or French (Cardon and Grabar 2018, Martin et al. 2020).

In section 2.1 we describe work on sentence simplification for Dutch. Section 2.2 describes recent techniques in generic sentence simplification, mainly performed on English. Section 2.3 describes related work with control tokens.

### 2.1 Sentence Simplification in Dutch

Research on text simplification in Dutch is scarce. Early approaches in Dutch text simplification were developed with the goal of syntactic sentence compression for subtitles for hearing-impaired people (Daelemans et al. 2004, Vandeghinste and Pan 2004). Bulté et al. (2018) developed a knowledge-based automated lexical simplification tool for Dutch. Their tool consists of several processing steps such as word sense disambiguation, difficult word estimation, replacement of difficult words by synonyms, reverse lemmatization, and output annotation. The system of Bulté et al. (2018) is intended to keep the simplified sentence as close as possible to the original sentence. Their system successfully replaces complex words like "aanzienlijk" with "groot", or "deal" with "akoord". In the same year, Sevens et al. (2018) proposed a rule-based approach with syntactic parsing and develop a complete simplification tool for Dutch. Their syntactic simplification module is used as a pre-processing step in text-to-pictograph transformation. Vandeghinste et al. (2019) created the

Wablieft corpus, a corpus based on a Belgian easy-to-read newspaper. Vandeghinste and Bulté (2019) compare the Wablieft easy-to-read newspaper corpus with a standard Dutch newspaper (de Standaard) and extract the linguistic features that account for readability in simpler text. Their study shows that syntactic metrics are a better indicator for easy reading than lexical metrics and the most important feature for sentence simplification is the number of words per sentence.

However, the aforementioned simplification approaches have limitations. Some systems, such as those by Sevens et al. (2018), rely heavily on rule-based systems for simplification. While rule-based methods can be effective in certain cases, they often struggle with handling complex linguistic phenomena or adapting to varied contexts. Automated simplification tools, like the one developed by Bulté et al. (2018), may not adequately account for individual user preferences or diverse reading abilities. What constitutes "simplified" language can vary widely depending on factors such as age, education level, and cognitive abilities. Critical evaluation and validation of simplification techniques are crucial to ensuring their utility and reliability in real-world applications. Without the usage of global simplification metrics, scholarly work is not reproducible and not comparable within and across languages and simplification approaches.

### 2.2 Newer Approaches to Sentence Simplification

Nonetheless, English remains the most prevalent language and area of work for sentence simplification. In the realm of simplification literature, which primarily focuses on English as the target language, a distinction is made between pre-neural approaches (Saggion 2017, Stajner 2021), which leverage extensive parallel corpora of aligned sentences (Martin et al. 2020), and data-driven neural approaches that emerged after 2015 (Alva-Manchego et al. 2020a, Stajner 2021).

Neural models are primarily based on a sequence-to-sequence architecture. In 2017, Nisioi et al. (2017) introduced a neural machine translation (NMT) system that used a sequence-to-sequence architecture for text simplification. Using long short-term memory networks (LSTM), Zhang and Lapata (2017) trained their model with reinforcement learning to optimize their model toward grammaticality, simplicity, and adequacy. Zhao et al. (2018) added an external paraphrase database as a source for real-world simplification rules to guide simplification learning based on a transformer architecture (Vaswani et al. 2017).

In their multilingual unsupervised sentence simplification (MUSS), Martin et al. (2022) apply large-scale data mining techniques by searching for simple and complex sentences in CCNet (a snapshot of open source web) (Wenzek et al. 2020) to create sentence pairs for supervised training. They then use these synthetic datasets to fine-tune BART (Lewis et al. 2020) and mBART (Liu et al. 2020) models and add control tokens from a prior study (Martin et al. 2020). Recent neural approaches are mostly complete systems that do not emphasize a target population or one simplification technique (Maddela et al. 2021, Stajner 2021) but have a more technical focus. With some exceptions, most approaches mentioned above require parallel data, that is, aligned complex and single sentence pairs to train their models towards learning the sentence simplification task.

### 2.3 Controllable Sentence Simplification

Controllable sentence simplification differentiates between decoding-based and learning-based approaches. Decoding-based controllable sentence simplification does not modify the training process but modifies "the system during decoding to control a given attribute" (Martin et al. 2020).

In learning-based approaches, on the other hand, model conditioning toward a specific output is done via training (Martin et al. 2020). In line with prior authors, this work uses a learning-based approach to control sentence simplification, much in the style of Martin et al. (2020).

Martin et al. (2020) prepend text control tokens to regulate the amount of compression, paraphrasing, and lexical and syntactic complexity in the target sentence. In this manner, model conditioning toward a specific output is performed via training, and the output generation is additionally

| | Train dataset | | Validation dataset | | Test dataset | |
|---|---|---|---|---|---|---|
| | complex | simple | complex | simple | complex | simple (avg.) |
| Rows | 10,000 | 10,000 | 992 | 992 | 359 | 359 |
| Sentences | 10,875 | 10,210 | 1061 | 1008 | 385 | 463.4 |
| Words | 220,806 | 161,018 | 22,196 | 16,305 | 7292 | 6116 |
| Characters (w. spaces) | 1,412,360 | 1,008,125 | 141,954 | 102,041 | 46,872 | 38,119 |
| Avg. sent. length | 20.30 | 15.77 | 20.92 | 16.17 | 18.94 | 13.20 |

Table 1: Corpus statistics for all datasets.

controlled during decoding. The authors define the following explicit control tokens: *NbChars* to control the compression level, *LevSim* to control similarity between the source and target sentence, *WordRank* to control for word complexity, and *DependencyTreeDepthRatio* to account for syntactic complexity.

Similarly, Sheang and Saggion (2021) first pre-train a transformer model on a parallel corpus for sentence simplification. In a second step, they adopt all control tokens introduced by Martin et al. (2020) and add a control token *Words* measuring the words ratio between the source and target sentence. Finally, Menta and Garcia-Serrano (2022) pre-train a transformer model for sentence simplification in technical domains such as Computer Science and Medicine. With regard to the control tokens used, they replace the *WordRank* token with a *Language Model Fill-Mask (LMFMR)* with masked token prediction, assuming that word rankings of simple words are lower. As a result, the lexical complexity is controllable with a *Language Model Fill-Mask* token that predicts simple words before their complex counterparts. This work uses a similar approach: we condition the generation process by pre-training control tokens suitable for Dutch syntax and semantics and manipulate text generation at decoding time.

## 3. Data

### 3.1 Creation of a Synthetic Corpus

Although Dutch is not considered a low-resource language, there is no publicly available parallel corpus in Dutch (Bulté et al. 2018). We turn towards building a synthetic dataset using two widely used parallel datasets in English, namely WikiLarge (Zhang and Lapata 2017) and ASSET (Abstractive Sentence Simplification Evaluation and Tuning) (Alva-Manchego et al. 2020b, Martin et al. 2020) which are automatically translated to Dutch. Translation is often used for creating synthetic datasets (Cardon and Grabar 2018, Galeev et al. 2021, Sakhovskiy et al. 2021).

As a training and validation set, we use WikiLarge,[2] the most extensive parallel simplification corpus in English (Zhang and Lapata 2017). It consists of 296.402, 2000, and 359 complex-simple sentence pairs for training, validation, and testing. As a test set, we use a translated version of ASSET[3] (Alva-Manchego et al. 2020b). ASSET contains 2000 validation sentences and 359 test sentences. In addition, the ASSET dataset contains ten manually edited reference simplifications per complex source sentence and hence more variations in the rewriting of simplifications. It can be assumed that the training and test sets are not out-of-domain: The model in this study (t5-base-dutch) was pre-trained on the mc4_nl_cleaned dataset, a crawled dataset from the World Wide Web. Consequently, the pre-training dataset covers many topics, which is also the case for the training and test sets, which both originate from Wikipedia. Both training and test sets were translated into Dutch using Google Neural Machine Translation (GNMT) (Wu et al. 2016).

Table 1 shows the corpus statistics of the final translated datasets. Again, the statistics for the ten reference sets in the test set are averaged across the documents. Most importantly, the average

2. https://github.com/XingxingZhang/dress
3. https://github.com/facebookresearch

| Train dataset | | | | | |
|---|---|---|---|---|---|
| | BLEU | ChrF | ChrF++ | TER | WER |
| Case sensitive | 77.63 | 89.09 | 88.14 | 15.75 | 0.17 |
| Case insensitive | 77.99 | 89.09 | 88.37 | 15.26 | 0.17 |
| **Test dataset** | | | | | |
| | BLEU | ChrF | ChrF++ | TER | WER |
| Case sensitive | 77.79 | 89.66 | 88.90 | 16.65 | 0.18 |
| Case insensitive | 78.22 | 89.79 | 89.11 | 16.13 | 0.18 |

Table 2: Comparison of translation metrics for the machine-translated train and test dataset with respect to human translation.

sentence length of the simple sentences across the training, validation, and test set is about 20 percent lower than for their complex counterparts. The simplified sentences comprise fewer words than the complex sentences across all three train, validation, and test datasets. However, the number of sentences in the ten reference sets of the test set diverges with a higher average number of sentences, containing more sentence splits than the train and validation dataset.

### 3.2 Human Reference Translation

The quality of the machine-translated corpora is evaluated by comparison with human reference translations for a sample of the sentences in the data. All sample sentences were chosen from the set of complex sentences within the training and test datasets randomly from across the whole dataset. A detailed comparison between the resulting samples and respective statistics can be found in **Appendix B**.

For evaluation, the machine-translated output of the sample was compared against the human reference translation. The metrics BLEU (Papineni et al. 2002), chrF (Popović 2015), chrF++ (Popović 2017), and TER (Snover et al. 2006) were measured with the SacreBLEU package (Post 2018). The Word Error Rate (WER) was calculated with the jiwer package[4].The results of the evaluation are shown in Table 2. All metrics have been computed in a case-sensitive and case-insensitive version.

Regardless of the casing, the BLEU score for machine-translated corpora is high (77.63 train, 77.79 test). These scores are superior to recent results, where Aiken (2019) reports translation performance results of GNMT from English to Dutch with a BLEU score of 71. It can be concluded that the machine-translated corpora are similar to their human reference translation. As a consequence, the machine-translated corpora can be used for fine-tuning a language model.

## 4. Method

This work aims at the following: First, to apply a large pre-trained language model for sentence simplification in a low-resource setting in Dutch. Second, to control the decoded sentence structure by prepending control tokens at encoding time similar to the approaches by Sheang and Saggion (2021) and Menta and Garcia-Serrano (2022). Third, to steer further text generation by limitations at the decoding phase. Fourth, to explore sentence simplification performance in a low-resource setting in Dutch.

4. `https://pypi.org/project/jiwer/`

### 4.1 Model Setup

We fine-tune a pre-trained language model in Dutch in a "teacher-forcing" manner (Williams and Zipser 1989) on sentence simplification. This means supervised learning, where a language model is presented with a complex sentence (source sentence) and then trained toward simplifying this input. After training, the trained model produces the simplified (target) sentence given the source sentence. In the following, we use an encoder-decoder architecture (Cho et al. 2014), where the encoder is trained to encode a complex sentence, and the decoder decodes the input into a simple sentence. To leverage the knowledge of a pre-trained model in Dutch, we use a pre-trained transformer model (Vaswani et al. 2017) in Dutch from the Hugging Face Transformers Library (Wolf et al. 2020). Transformer models require pre-training; hence, any model must be exposed to training data in the target language before fine-tuning on a specific task. For Dutch, most publicly available transformer models with an encoder-decoder architecture are variants of T5. T5 is an encoder-decoder text-to-text transformer (Raffel et al. 2020) that was pre-trained with span corruption and denoising on unsupervised and supervised tasks such as question-answering translation or summarization. The following pre-trained Dutch variants of T5 are publicly available in the Hugging Face Transformers Library (Wolf et al. 2020) and have been explored: flan-t5-base[5] (Flan-T5), t5-base-dutch[6] (T5-base), and t5-v1.1-base-dutch-cased[7] (T5 V1.1). For evaluating the best model for the task at hand, performance across models was measured. To do so, all models were trained with the same learning rate (0.001) and a small set of parallel data. Tests with flan-t5-base and t5-v1.1-base-dutch-cased model showed a lower performance given the same number of training iterations and training data. The t5-base-dutch model provided good results on the tests which is in line with prior simplification research (Sheang and Saggion 2021, Taylor et al. 2022). In the remainder of this work, the t5-base-dutch model will henceforth be used.

The t5-base-dutch model has an original T5 configuration with 223M parameters, 12 attention heads, 12 layers, and a sequence length of 512 tokens. The t5-base-dutch model has not been fine-tuned on a simplification task and was pre-trained on mc4_nl_cleaned[8] data. The mc4_nl_cleaned dataset originates from the allenai/c4[9] dataset, a variant of the C4 dataset (Dodge et al. 2021) comprising sentences from the Common Crawl web scrape[10]. The Dutch portion of the mc4 dataset was extracted and stripped from inappropriate words, fill words, javascript code, short sentences, or exceedingly long words to create this dataset. The model uses a SentencePiece tokenizer (Kudo and Richardson 2018) with 32,003 tokens.

### 4.2 Simplification Control Mechanism

Several attributes are essential for sentence simplification. First, in line with the findings of Vandeghinste and Bulté (2019), the average sentence length of simplified sentences should be shorter than that of their complex counterparts and contain fewer subclauses. In their study, the average sentence length for easy text is only 8.31 words per sentence, with 14.19 words per sentence in standard Dutch sentences. Furthermore, syntax metrics such as dependency tree depth (4.44 for easy-to-read text, 6.12 for standard text) and length of clauses (8.13 for easy-to-read text, 11.46 for standard text) were found to have a reasonably high effect on readability (Vandeghinste and Bulté 2019). Consequently, the average dependency tree depth and the length of clauses should be lower in simple sentences.

Concerning simplification on a lexical level, most work is concerned with identifying difficult words (complex word identification), which are then chosen to be replaced by easier ones (substi-

5. https://huggingface.co/google/flan-t5-base
6. https://huggingface.co/yhavinga/t5-base-dutch
7. https://huggingface.co/yhavinga/t5-v1.1-base-dutch-cased
8. https://huggingface.co/datasets/yhavinga/mc4_nl_cleaned
9. https://huggingface.co/datasets/allenai/c4
10. https://commoncrawl.org/the-data/get-started/

tution generation) (Bulté et al. 2018, Paetzold and Specia 2016b). Word frequencies are the best indicator to identify the difficulty of a word (Paetzold and Specia 2016b, Wilkens et al. 2014). Hence, complex words are replaced with more familiar words to achieve sentence simplification, reducing lexical complexity. Prior work (Martin et al. 2020, Menta and Garcia-Serrano 2022, Sheang and Saggion 2021) discerns between five explicit control tokens for sentence simplification: the amount of sentence compression, word length, the amount of paraphrasing, syntactic complexity, and lexical complexity.

To control the five simplification attributes, we implement the following explicit control tokens:

- **Sentence length (***CharLengthRatio/CLR)*: The number of characters in a sentence and hence sentence length is a good metric for simplicity (Martin et al. 2018). The average number of words and syllables is commonly used to evaluate readability (Brouwer 1963, Kincaid et al. 1975). Martin et al. (2020) first implemented the *CharLengthRatio*, which sets the character length between the source sentence in relation to the character length of the target sentence. By controlling the length of the target sentence, the amount of compression on the target sentence and the deletion of content are controlled as well. A lower *CharLengthRatio* indicates a shorter target sentence.

- **Number of words (***WordLengthRatio/WLR***)**: We adapted the approach of Sheang and Saggion (2021) and include a *WordLengthRatio* to the control tokens. The *WordLengthRatio* sets the length of words in the source sentence in relation to the length of words in the target sentence. This measure is disputed: Whereas Wilkens et al. (2014) showed that for English and Portuguese, the word length is no predictor of its relative complexity, this does not necessarily apply to Dutch, which contains compound words (Macken and Tezcan 2018, Pander Maat et al. 2014). Nevertheless, shorter words are easier to read than longer words for people with difficulties in reading (Rello et al. 2013). In preliminary tests, adding a *WordLengthRatio* to the control tokens effectively replaces long words with shorter substitutes or rephrasing (Sheang and Saggion 2021). For calculation of the *WordLengthRatio*, the words have first been tokenized with the Moses tokenizer from the SacreBLEU (Post 2018) package.[11] A lower *WordLengthRatio* shows shorter words in comparison between source and target sentence.

- **Paraphrasing (***LevenshteinRatio/LR***)**: It is preferrable that the generated simplified target sentence remains close to its source sentence. Doing so, the amount of paraphrasing, meaning the number of edit operations (paraphrase, edit, delete) between the source and target sentence (Wubben et al. 2010), is measured by the normalized Levenshtein similarity (Levenshtein 1965). We follow prior implementations (Martin et al. 2022, Menta and Garcia-Serrano 2022, Sheang and Saggion 2021), and apply a normalized *LevenshteinRatio*. The *LevenshteinRatio* is based on characters with prior tokenization by the Moses tokenizer in the SacreBLEU package. A higher *LevenshteinRatio* indicates a higher similarity between the source and target sentence and a lower edit distance.

- **Lexical complexity (***WordRankRatio/WRR***)**: Paetzold and Specia (2016a) have shown that word frequencies are best for assessing the complexity of words. Especially for people with learning difficulties, frequent words are easier to read and more understandable than infrequent words (Rello et al. 2013). We follow the approach of Martin et al. (2020) as implemented by Sheang and Saggion (2021) and apply a *WordRankRatio*. The *WordRankRatio* is the third-quartile of log-ranks (inverse frequency order) of all words in the target sentence divided by the inverse frequency order of all words in the source sentence (Martin et al. 2020). This was implemented using a word embedding vocabulary. In word embedding vocabularies, more frequent words have a lower rank and, consequently, a lower log-rank. Hence sentences that employ more frequent words have a lower *WordRankRatio*.

11. `https://github.com/mjpost/sacrebleu`

| | |
|---|---|
| Source sentence | CLR_0.41 WLR_0.41 LR_0.32 WRR_0.85 DTDR_0.67 De storm kwam met maximale kracht aan land in het zuidwesten van Florida, waardoor het de sterkste orkaan was die de Verenigde Staten trof sinds de orkaan Andrew twaalf jaar eerder, in 1992, Florida trof. |
| Target sentence | Dit maakte het tot een sterke orkaan van categorie 4 op de schaal van Saffir-Simpson. |

Table 3: An example of a complex and a simple sentence.

A comparison of several word embedding vocabularies has been conducted; results are listed in **Appendix A**. The word embedding vocabulary[12] from Fares et al. (2017) scored best, probably due to its extensive vocabulary and wide coverage of topics. The word embedding vocabulary was constructed for CONLL17 based on Common Crawl Data in Dutch. The model is a Word2Vec Continuous Skip-gram model with a vector dimension size of 100 (Fares et al. 2017).

Arguably, the word rank can only be applied if the word is found in the vocabulary, which means a word without a match is not considered. If the complex sentence is longer than the simple sentence and applies many fill words, the *WordRankRatio* can be misleading: If these fill words are frequently used words, their word rank is lower, and the complex long sentence receives a lower complexity score than its simpler counterpart. To sum up, a lower *WordRankRatio* indicates the usage of more frequent words in the target sentence and is preferrable.

- **Syntactic complexity (***DependencyTreeDepthRatio/DTDR***)**: Martin et al. (2020) used *DependencyTreeDepthRatio* to approximate syntactic complexity. Their experiments showed that deeper dependency trees imply longer spans and correspond to more sophisticated sentences (Martin et al. 2020). For Dutch, dependency tree depth is a strong differentiator between easy-to-read and standard newspaper articles (Vandeghinste and Bulté 2019). *DependencyTreeDepthRatio* measures the maximum depth of the target sentence's dependency tree divided by the source sentence's maximum tree depth (Martin et al. 2020, Sheang and Saggion 2021). For the implementation of the dependency tree, the Dutch model pipeline 'nl_core_news_sm' from spacy[13] was used. A minor score for *DependencyTreeDepthRatio* indicates a better result, meaning the target sentence is less syntactically complex than its original counterpart.

  Following the implementation of Martin et al. (2020) and Sheang and Saggion (2021), the above-defined ratios are always the control token values for the target sentence divided by their respective values for the source sentence. For example, the number of words in the target sentence is divided by the number of words in the source sentence to compute the word ratio. The desired ratio must be provided for each control token before encoding. To steer these five simplification attributes, the aforementioned control tokens are implemented and prepended to the training data as individual information to each complex sentence for training. Table 3 shows an example of tokenization.

### 4.3 Decoder Search

Finally, text generation can be steered by limitations at the decoding phase. Decoding refers to the generation of a meaningful and factual sequence from tokens at inference time. Several decoding strategies are relevant to exploring.

12. `http://vectors.nlpl.eu/repository`
13. `https://spacy.io/models/nl`

**Greedy decoding** In a greedy search, the word $w_t$ with the highest probability

$$w_t = argmax_w P(w|w_{1:t-1})$$

is chosen at each timestep $t$. Greedy decoding can generate grammatical sentences. However, they are more likely to contain errors if the distributions are not learned properly, which results in the accumulation of errors in stepwise sampling (Shao et al. 2017). Moreover, greedy decoding ignores more suitable words with a lower probability in favor of the word with the highest probability. As a result, the quality of output sentences can vary highly and be factually wrong.

**Combination of top-p and top-k sampling** In top-k sampling (Fan et al. 2018), the most likely $k$ words are filtered, the probability mass is distributed over those k sampled words, and the words with the highest probability are returned. In top-p (or nucleus) sampling, the smallest set of words whose cumulative likelihood is greater than the given threshold probability $p$ (Holtzman et al. 2019) is determined. The probability is distributed over this set of words in the next step, and the best is chosen. The combination of top-p sampling followed by top-k sampling, as used in Yang et al. (2021) and Keskar et al. (2019), helps to avoid low-ranked words and gives some dynamic selection options to the model promoting diversity.

**Beam search decoding** In line with prior research (Menta and Garcia-Serrano 2022, Sheang and Saggion 2021), beam search is explored. Beam search keeps the most likely n-beams of tokens in a sequence based on conditional probability at each iteration. The option with the highest probability for the entire sequence is chosen. However, these mainly stem from one beam with a high probability value, resulting in outputs that only contain small changes to the sentence. Hence, beam search does not promote diversity and could lead to copying the original sentence (Vijayakumar et al. 2016).

### 4.4 Scarce Data

Several studies explore simplification in a low-resource setting: Surya et al. (2019) build their custom sequence-to-sequence autoencoder model specifically for low-resource settings and use a combination of 10,000 parallel training sentences from Simple Wikipedia (Hwang et al. 2015) and the Split rephrase set by Narayan et al. (2017). For their neural simplification system, Palmero Aprosio et al. (2019) use a training set of 30,000 and 53,000 rows for Italian and Spanish. Maruyama and Yamamoto (2019) report satisfactory results from pre-training a custom transformer language model, which is fine-tuned on 3000 examples of parallel data. Unfortunately, the authors did not further specify their transformer language model architecture nor publish their code. Model performance based on limited training data is vital because parallel corpora are expensive to create and require sufficient qualitative written text pairs, which can be a bottleneck in low-resource target languages. As a benchmarking test, this work explores sentence simplification results with varying dataset sizes of 2000, 6000, and 10,000 rows of data.

## 5. Experiments

### 5.1 Automatic Evaluation

We use the Easier Automatic Sentence Simplification Evaluation (EASSE) framework[14] by Alva-Manchego et al. (2019) to evaluate the quality of the generated sentence simplifications.

SARI (Xu et al. 2016) measures sentence simplicity based on add, keep, and delete operations. SARI compares the system-generated simplified output sentences to its source sentence and multiple reference sentences. The metric averages the F1 scores of add, keep and delete operations in relation to the generated output sentences. The EASSE package also implements improvements on the

14. `https://github.com/feralvam/easse`

initial SARI implementation[15] as published by Xu et al. (2016), where normalization is applied to the source, prediction, and reference sentences.

The Flesch-Kincaid Grade Level (FKGL) (Kincaid et al. 1975) is a measure that is frequently reported with simplification publications (Kumar et al. 2020, Martin et al. 2022, Martin et al. 2020, Rashid and Amirkhani 2023). In this work, the Flesch-Kincaid Grade Level is used as an auxiliary metric, given that it relies on average sentence lengths, favors shorter sentences, and does not account for grammaticality and meaning preservation in simplified sentences (Wubben et al. 2012). The Flesch-Kincaid Grade Level is the result of a linear regression between the number of words and the number of syllables per word over a simple sentence. A lower score indicates higher readability.

BLEU (Papineni et al. 2002) measures correct sentence generation by calculating the n-gram matches between the generated sentence and several reference sentences. It has been proven that the more reference translations used, the higher the BLEU score (Papineni et al. 2002, Post 2018). Although frequently reported in sentence simplification (Kumar et al. 2020, Martin et al. 2022, Surya et al. 2019), BLEU is not suitable as a primary simplification metric (Alva-Manchego et al. 2020a), given that it correlates poorly with simplicity when sentence splitting was performed. Despite this, BLEU was found to correlate highly with a human estimation of grammaticality and meaning preservation but favors longer simplifications with n-grams that are present in reference sentences (Xu et al. 2016). We evaluated sentence simplification using SARI and add, keep, and delete ratios. Average scores on BLEU and FKGL are published for comparison.

### 5.2 Training Details and Hyperparameter Search

Task-specific fine-tuning has been done on Google Colab Free Version, with a T4/K80 GPU with 15GB available RAM. All models were trained using the Hugging Face Transformers library (Wolf et al. 2020). Hyperparameter tuning was done with Optuna (Akiba et al. 2019) and logging with WandB (Biewald 2020). The average training time was 21 minutes, depending on the size of the dataset. No task prefix has been added during fine-tuning.

The seed was set to 12 to compare results across runs, and gradient accumulation was done after four training steps in every run. Gradient accumulation is the number of steps the gradient is collected before a backward pass is performed. This reduces memory size during training. We completed a hyperparameter search for the number of training epochs, the learning rate, the evaluation batch size, the training batch size, and the warmup steps. To evaluate the effect of reduced training data, we trained with varying lengths of the dataset containing 2000, 6000, and 10,000 rows. Several variations in warmup steps were tested. Warmup steps are the number of steps before the linearly increasing learning rate reaches the set learning rate. Like in prior studies (Menta and Garcia-Serrano 2022, Sheang and Saggion 2021), the best results were achieved with five warmup steps.

For training, two optimizers were tested. Following prior research (Menta and Garcia-Serrano 2022, Sheang and Saggion 2021), we used the AdamW optimizer with fixed weight decay (Loshchilov and Hutter 2017) with its default parameters. Then we tested the Adafactor optimizer (Shazeer and Stern 2018), given that it is used in the original T5 model (Raffel et al. 2020) using the default configuration. The best parameter configurations of the hyperparameter search for the optimizers AdamW and Adafactor are indicated in Table 4.

During training, all models were evaluated on the evaluation loss. The best-performing checkpoint was logged and stored for each combination of optimizer and dataset size. Next, each model checkpoint was used to test the simplification quality on the test set. The token limit was set to a maximum length of 128 tokens without further specification of generation parameters. The average time for generating simplified sentences on the test set (359 rows) was 45 minutes; the resulting SARI scores were documented.

15. `https://github.com/cocoxu/simplification`

| Parameter | Search Space | AdamW | Adafactor |
|---|---|---|---|
| Seed | – | 12 | 12 |
| Gradient accumulation | [1,4] | 4 | 4 |
| Learning rate | [1e-3 - 1e-5] | 0.0001 | 0.0001 |
| $\epsilon_{AdamW}$ | [3e-7 – 1e-8] | 1.000e-8 | na |
| epsAdafactor | – | na | 1e-30, 1e-3 |
| Betas $(\beta_1, \beta_2)_{AdamW}$ | – | 0.9, 0.999 | na |
| Batch size | [6,8,12,18] | 6 | 6 (8 with 10,000) |
| Weight decay | – | 0.1 | 0.0 |
| Epochs | [1,2,3,4] | 4 | 4 |
| Warmup steps | [0, 5, 2000, 5000] | 5 | 5 |
| Dataset size | [2000, 6000, 10,000] | 10,000 | 10,000 |

Table 4: Best parameter configurations for optimizers AdamW and Adafactor after hyperparameter search. Note that for 10000 rows, a batch size of 8 was used.

| Optimizer | Dataset size | Eval loss | SARI↑ | Add↑ | Keep↑ | Delete↑ | FKGL↓ | BLEU↑ |
|---|---|---|---|---|---|---|---|---|
| Adafactor | 2000 | 1.413 | 25.05 | 3.05 | 59.08 | 13.02 | 6.86 | 48.60 |
| | 6000 | 1.460 | 25.07 | 1.86 | 58.12 | 15.23 | 7.27 | 57.08 |
| | 10,000* | **1.412** | 26.04 | 3.35 | **59.30** | 15.47 | **6.27** | 48.32 |
| AdamW | 2000 | 1.615 | 32.26 | **4.24** | 56.91 | 35.62 | 6.72 | 41.84 |
| | 6000 | 1.472 | **34.65** | 2.26 | 55.59 | **46.09** | 8.95 | 78.95 |
| | 10,000 | 1.423 | 23.43 | 0.34 | 58.06 | 11.88 | 10.42 | **88.96** |

Table 5: Model performance on the test set using only 2000, 6000, and 10,000 training examples. The best result in each column is bolded, and the best model is marked with *. Thse scores are not yet indicative, given that the models have been pre-trained on a fixed set of control tokens. (↑ Higher is better, ↓ lower is better.)

Overall, based on the evaluation loss, the training performance with the Adafactor optimizer was better than with the AdamW optimizer. For the Adafactor optimizer, the training loss with only 2000 rows of training data was equal to that with 10000 rows of data, applying the same parameter configuration. Furthermore, the SARI value of the Adafactor model with 2000 rows of training data (25.05) is not much lower than with 10000 rows of training data (26.04). To conclude, for the Adafactor optimizer, the best results are generated with 10000 rows of data. Regarding AdamW, the model performs even better on 6000 rows of training data (34.65) and on 2000 rows of data (32.25) than with 10,000 rows of training data (23.43) when measured by SARI.

However, the metrics for sentence simplification (SARI, FKGL, BLEU) are not indicative, given that the models have only been pre-trained on an exemplary set of control tokens and not on the final set of tokens. Moreover, no adjustment to the decoder has been made yet. Consequently, the best-performing model has been chosen by manual assessment of the generated output rather than by the SARI score. A cross-check on the generated output texts of all models reveals hallucinations and repetitive sentences (AdamW, 6000 rows; AdamW, 2000 rows). The model trained with the Adafactor optimizer on 10,000 rows of data contained the least number of hallucinations and repetitions and performed best on the test set (evaluation loss 1.412). Consequently, this model was chosen as the baseline for further simplification experiments.

Table 5 displays model performance without prior configuration of the control token on the test set (best-performing model* Adafactor, 10000 rows of data). Table 6 displays an example of the generated text of the best-performing model. The model successfully reproduces the original

| | |
|---|---|
| Original | Zijn volgende werk, zaterdag, volgt op een bijzonder veelbewogen dag in het leven van een succesvolle neurochirurg. |
| Simplified | Zijn volgende werk, zaterdag **(zaterdag)** volgt op een bijzonder veelbewogen dag in het leven van een neurochirurg. **Het is de laatste werkdag die hij heeft doorgebracht als neurochirurg bij een ziekenhuis. Hij was erg onder de indruk toen hij** |
| Original | Daar stierf hij zes weken later, op 13 januari 888. |
| Simplified | **Hij stierf** zes weken later, op 13 januari 888. **Daar stierf hij zes weken eerder dan zijn vader in de jaren zestig had gedaan. Zes weken later stierf hij daar. Dat was een jaar na zijn dood** |
| Original | Ze zijn cultureel verwant aan de kustbewoners van Papoea-Nieuw-Guinea. |
| Simplified | Ze zijn cultureel verwant aan de kustbewoners van Papoea-Nieuw-Guinea. **De zee is kleiner dan het vasteland in Nieuw-Guinea, maar wel groter als een eilandje bij elkaar. Het wordt ook gebruikt om te vissen op zand** |

Table 6: Examples of generated simplifications with the best model without prior configuration of the control tokens. Incorrect transformations (lost or changed meaning) are marked in bold.

sentence with minor simplifications yet generates additional sentences containing hallucinations or reformulations of the original sentence.

### 5.3 Controllable Generation with Control Tokens

As described in section 4.2, the implemented control tokens are *CharLengthRatio, WordLengthRatio, LevenshteinRatio, WordRankRatio*, and *DependencyTreeDepthRatio*. The control token values from prior studies cannot be used as these token values have been optimized for English text but not for Dutch text. To understand the effect of control tokens on output generation for Dutch text, a detailed analysis of the control tokens is performed. We do so by keeping all token values at 0.9 and modifying only the values of the control token under examination. Table 7 lists original complex sentences from the test set, which serve as a reference for the following modifications. In the tables that follow, the corresponding reference number of the original sentence is indicated in the rightmost column.

Experimenting with different values for *LevenshteinRatio* and *WordRankRatio*, we observe the same effects as Martin et al. (2020): If the *LevenshteinRatio* and *WordRankRatio* are set to low values (0.4 for *LR* and 0.2 for *WRR*), the output is ungrammatical, and the sentence produced is nonsensical. Hence, the value for Levenshtein similarity needs to be carefully chosen, avoiding extreme values. Table 8 shows that low *WordRankRatio* values (0.2) in combination with medium values (0.6, 0.7) for *LevenshteinRatio* values cause erroneous sentences that are ungrammatical or factually wrong. On the other hand, combining the *LevenshteinRatio* of 0.7 and *WordRankRatio* value of 0.4 produces grammatical sentences with correct meaning and represents the minimum viable combination.

Table 9 shows the effects of a reduction of token values for *CharLengthRatio* and *WordLength-Ratio*. For *CharLengthRatio*, with $CLR = 0.7$, subclauses are shortened, and with $CLR = 0.4$, one subclause has been entirely omitted. It is difficult to spot any modification in word length as an effect of reducing the *WordLengthRatio* token value, such as a replacement of a long word by a shorter word. It would have been expected that longer words, such as "belangrijkste" or "toegangspoort" would be replaced by shorter words.

For varying *WordRankRatio* token values, we observe a replacement of complex words. Thus some level of lexical simplification is present: In Table 10, complex words such as "Desalniettemin" are produced with higher values of *WordRankRatio* while not present with lower values. However, a word like "emuleerde" is not replaced, whereas "vertegenwoordigt" is replaced by "is". The desired effect only occurs with very low token values (0.2) for *WordRankRatio*.

| Original sentences (complex version) | Example |
|---|---|
| Zijn volgende werk, zaterdag, volgt op een bijzonder veelbewogen dag in het leven van een succesvolle neurochirurg. | 1 |
| Sinds 2000 ontvangt de ontvanger van de Kate Greenaway-medaille ook de Colin Mears Award ter waarde van £ 5000. | 2 |
| Fives is een Britse sport waarvan wordt aangenomen dat deze dezelfde oorsprong heeft als veel racketsporten. | 3 |
| Beide namen werden opgeheven in 2007 toen ze werden samengevoegd tot The National Museum of Scotland. | 4 |
| Jeddah is de belangrijkste toegangspoort tot Mekka, de heiligste stad van de islam, die valide moslims minstens één keer in hun leven moeten bezoeken. | 5 |
| Veel soorten waren tegen het einde van de negentiende eeuw verdwenen, met Europese vestiging. | 6 |
| Perry Saturnus (met Terri) versloeg Eddie Guerrero (met Chyna) om het WWF European Championship te winnen (8:10) Saturnus drukte Guerrero vast na een duikelleboog. | 7 |
| Desalniettemin emuleerde Tagore talloze stijlen, waaronder handwerk uit het noorden van New Ireland, Haida- gravures van de westkust van Canada (British Columbia) en houtsneden van Max Pechstein. | 8 |
| Men denkt dat de Grote Donkere Vlek een gat vertegenwoordigt in het methaanwolkendek van Neptunus. | 9 |
| In zuivere vorm komt dextromethorfan voor als een wit poeder. | 10 |
| De drummers worden gevolgd door dansers, die vaak de sogo spelen (een kleine trommel die bijna geen geluid maakt) en de neiging hebben om een uitgebreidere - zelfs acrobatische - choreografie te hebben. | 11 |
| De BZÖ verschilt echter een beetje in vergelijking met de Partij voor de Vrijheid, zoals voor een referendum over het Verdrag van Lissabon maar tegen een terugtrekking uit de EU. | 12 |
| Ze zijn cultureel verwant aan de kustbewoners van Papoea-Nieuw-Guinea. | 13 |
| Reizen door zeer afgelegen gebieden, op afgelegen wegen, vereist echter planning vooraf en een geschikt, betrouwbaar voertuig (meestal een vierwielaandrijving). | 14 |
| Alessandro ('Sandro') Mazzola (geboren op 8 november 1942) is een voormalig Italiaans voetballer. | 15 |

Table 7: Extract of complex sentences from the test set.

| | | |
|---|---|---|
| CLR 0.9 WLR 0.9 | **Hij is** neurochirurg **en studeerde aan de Universiteit van Californië** | 1 |
| **LR 0.4 WRR 0.2** | Colin Mears Award **(1999) is een onderscheiding van de Britse actrice en** | 2 |
| DTDR 0.9 | **actrice Kate Greenaway.** | |
| CLR 0.9 WLR 0.9 | Fives is een Britse sport. | 3 |
| **LR 0.6 WRR 0.4** | **Hij is** neurochirurg **en vindt zijn weg naar de top**. | 1 |
| DTDR 0.9 | Het National Museum of Scotland **werd opgeheven in 2007**. | 4 |
| CLR 0.9 WLR 0.9 | Fives is een sport waarvan wordt aangenomen dat deze dezelfde oorsprong heeft als | 3 |
| **LR 0.7 WRR 0.2** | veel racketsporten. | |
| DTDR 0.9 | **Hij is** een neurochirurg. | 1 |
| | Het National Museum of Scotland **werd opgeheven in 2007** | 4 |
| CLR 0.9 WLR 0.9 | Fives is een sport waarvan wordt aangenomen dat deze dezelfde oorsprong heeft als | 3 |
| **LR 0.7 WRR 0.4** | veel racketsporten. | |
| DTDR 0.9 | **Hij is** een neurochirurg. | 1 |
| | Beide namen werden opgeheven in 2007 toen ze samengevoegd werden tot het National Museum of Scotland. | 4 |

Table 8: Lower *LevenshteinRatio* tokens cause shorter sentences and contain hallucinations. Incorrect transformations (lost or changed meaning) are marked in bold.

| | | |
|---|---|---|
| **CLR 0.7 WLR 0.7** | Jeddah is de belangrijkste toegangspoort tot Mekka, de heiligste | 5 |
| LR 0.9 WRR 0.9 | stad van de islam. | |
| DTDR 0.9 | Veel soorten waren tegen het einde van de negentiende eeuw verdwenen, met Europese vestiging. | 6 |
| **CLR 0.4 WLR 0.4** | Jeddah is de belangrijkste toegangspoort tot Mekka, de heiligste | 5 |
| LR 0.9 WRR 0.9 | stad van de islam. | |
| DTDR 0.9 | Veel soorten waren tegen het einde van de negentiende eeuw verdwenen. | 6 |

Table 9: Effect of different token values for *CharLengthRatio* and *WordLengthRatio*. Incorrect transformations (lost or changed meaning) are marked in bold.

| | | |
|---|---|---|
| CLR 0.9 WLR 0.9<br>LR 0.9 **WRR 0.2**<br>DTDR 0.9 | Perry Saturn (met Terri) versloeg Eddie Guerrero (met Chyna) om het WWF European Championship te winnen. | 7 |
| CLR 0.9 WLR 0.9<br>LR 0.9 **WRR 0.7**<br>DTDR 0.9 | Perry Saturn (met Terri) versloeg Eddie Guerrero (met Chyna) om het WWF European Championship te winnen (8:10) Saturnus drukte Guerrero vast na een duikelleboog. | 7 |
| CLR 0.9 WLR 0.9<br>LR 0.7 **WRR 0.2**<br>DTDR 0.9 | Tagore emuleerde **veel** stijlen, waaronder handwerk uit het noorden van New Ireland, Haida-gravures van de westkust van Canada (British Columbia) en houtsneden van Max Pechstein. | 8 |
| CLR 0.9 WLR 0.9<br>LR 0.7 **WRR 0.4**<br>DTDR 0.9 | Desalniettemin emuleerde Tagore **veel** stijlen, waaronder handwerk uit het noorden van New Ireland, Haida-gravures van de westkust van Canada (British Columbia) en houtsneden van Max Pechstein. | 8 |
| CLR 0.9 WLR 0.9<br>LR 0.9 **WRR 0.2**<br>DTDR 0.9 | Men denkt dat de Grote Donkere Vlek een gat **is** in het methaanwolkendek van Neptunus. | 9 |
| CLR 0.9 WLR 0.9<br>LR 0.9 **WRR 0.9**<br>DTDR 0.9 | Men denkt dat de Grote Donkere Vlek een gat vertegenwoordigt in het methaanwolkendek van Neptunus. | 9 |

Table 10: Effect of different token values for *WordRankRatio*. Incorrect transformations (lost or changed meaning) are marked in bold.

| | | |
|---|---|---|
| CLR 0.9 WLR 0.9 | Dextromethorfan **is** een wit poeder. | 10 |
| LR 0.7 WRR 0.5<br>**DTDR 0.9** | De drummers worden gevolgd door dansers, die vaak de sogo spelen (een kleine trommel die bijna geen geluid maakt) en de neiging hebben om **meer uitgebreide choreografieën** te hebben. | 11 |
| | De BZÖ **is een beetje anders dan de PVV**, **net als vóór een referendum over het Verdrag van Lissabon.** | 12 |
| CLR 0.9 WLR 0.9 | Dextromethorfan **is** een poeder. | 10 |
| LR 0.7 WRR 0.5<br>**DTDR 0.7** | De drummers worden gevolgd door dansers, die vaak de sogo spelen (een kleine trommel die bijna geen geluid maakt) en de neiging hebben om **meer uitgebreide** - zelfs acrobatische - **choreograf** | 11 |
| | De BZÖ **is een beetje anders dan de PVV**, **net als vóór een referendum over het Verdrag van Lissabon.** | 12 |
| CLR 0.9 WLR 0.9 | Dextromethorfan **is** een poeder. | 10 |
| LR 0.7 WLR 0.5<br>**DTDR 0.6** | De drummers worden gevolgd door dansers, die vaak de sogo spelen (een kleine trommel die bijna geen geluid maakt). | 11 |
| | De BZÖ **is een partij die voorstander is van een terugtrekking uit de EU.** | 12 |

Table 11: Effect of different token values for *DependencyTreeDepthRatio*. Incorrect transformations (lost or changed meaning) are marked in bold.

Tests on the *DependencyTreeDepthRatio* in Table 11 show that a low token value causes the sentence to be split into multiple shorter sentences or advances the creation of shorter sentences. Also, subclauses are omitted, which is a desired effect of simplifying sentence structure.

### 5.4 Hyperparameter Search of Control Tokens

Following this prior experimentation, the best set of control tokens was identified with a hyperparameter search using Optuna (Akiba et al. 2019) and logging using WandB (Biewald 2020). The hyperparameter search for the best combination of control token ratios works as follows: A trial study with target ratios for each control token between 0.2 and 1.0 was set up. Then, the hyperparameter search was guided with independent sampling with the tree-structured parzen estimator algorithm (Bergstra et al. 2011) using an incremental search value of 0.05. For each trial, given a set of target ratios (*CharLengthRatio, WordLengthRatio, LevenshteinRatio, WordRankRatio*, and *DependencyTreeDepthRatio*), the control tokens per sentence are generated and prepended to the

| Control Tokens | | | | | Scores | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *CLR* | *WLR* | *LR* | *WRR* | *DTDR* | SARI↑ | Add↑ | Keep↑ | Delete↑ | BLEU↑ | FKGL↓ |
| **0.7** | **0.6** | **0.6** | **0.55** | **0.75** | **37.40** | **2.35** | **53.90** | **55.93** | **80.88** | **7.92** |
| 0.55 | 0.75 | 0.6 | 0.7 | 0.7 | 37.31 | 2.33 | 54.53 | 55.04 | 82.84 | 8.13 |
| 0.8 | 0.75 | 0.6 | 0.5 | 0.75 | 36.99 | 2.29 | 54.45 | 54.22 | 82.89 | 8.21 |
| 0.7 | 0.6 | 0.6 | 0.55 | - | 37.36 | 2.51 | 53.00 | 57.00 | 81.15 | 7.84 |
| 0.75 | 0.55 | 0.6 | 0.7 | - | 36.85 | 2.27 | 54.59 | 53.68 | 83.71 | 8.16 |
| 0.7 | 0.5 | 0.6 | 0.65 | - | 36.78 | 2.13 | 54.28 | 53.92 | 83.04 | 8.16 |
| 0.85 | 0.7 | 0.55 | - | - | 36.32 | 2.26 | 54.02 | 52.67 | 80.35 | 8.34 |
| 0.65 | 0.7 | 0.65 | - | - | 35.36 | 2.08 | 55.12 | 48.90 | 83.55 | 8.49 |
| 0.65 | 0.85 | 0.8 | - | - | 35.17 | 2.04 | 55.66 | 47.81 | 84.30 | 8.77 |
| 0.7 | 0.6 | - | - | - | 37.32 | 2.18 | 52.62 | 57.16 | 78.98 | 7.62 |
| 0.55 | 0.85 | - | - | - | 36.30 | 2.12 | 53.64 | 53.15 | 80.64 | 8.16 |
| 0.6 | 0.75 | - | - | - | 35.50 | 1.83 | 54.91 | 49.75 | 83.73 | 8.49 |

Table 12: Results of hyperparameter search on the best-performing model with varying control token values.

complex sentence, e. g. the complex sentences get a prefix that consists of the control tokens. This creates an individual train set for each combination of control tokens. The best-performing model is trained on this training set in the subsequent step. The training parameters are identical to the best-performing model with the Adafactor optimizer trained on 10,000 rows of data, as described in Table 5.

For testing, the complex sentences of the test set are prepended with the respective control tokens for a given target ratio, as described in section 4.2 and shown in Table 3. These sentences are then fed into generation using the pre-trained model. Next, each generated set of output sentences was evaluated on the SARI score. In addition, we also report the respective SARI add, keep, and delete scores (Fadd, Fkeep, and Fdelete) and FKGL and BLEU scores. Finally, the token values that achieved the best scores on the test set were selected. Table 12 shows the control token values after the hyperparameter search. The resulting best set, based on SARI, was the following combination of tokens: $CLR = 0.7$, $WLR = 0.6$, $LR = 0.6$, $WRR = 0.55$, $DTDR = 0.75$, with a resulting SARI score of 37.40. Hence, adding control tokens significantly improves the performance (SARI +11.27).

Table 13 shows the individual influence of each token on SARI performance. Each token has a significant influence on performance improvement (avg. +6.65 SARI). With only one token added, *LevenshteinRatio* performs best on the test set (+8.16), which is in concordance with the results of (Sheang and Saggion 2021). Overall, the results designate that adding control tokens significantly improves simplification performance, and all control tokens are essential to the overall simplification result. Like prior studies, *LevenshteinRatio* and *WordRankRatio* are the best single tokens (Martin et al. 2020, Sheang and Saggion 2021).

A significant improvement is made with two control tokens added (SARI 37.52). With four control tokens, *CLR*, *WLR*, *LR*, and *WRR*, the SARI score is almost as high as with all tokens (37.36). The best result is produced with all control tokens added (SARI 37.40). Interestingly, adding *LR* as a third control token does lower the SARI score. This could be because the alignment of sentences in the original English test dataset (ASSET) is good, and the test dataset itself does not contain much reordering compared to other datasets (Zhao et al. 2022). Moreover, the compression ratio in the original test set is high (Alva-Manchego et al. 2020b, Zhao et al. 2022), so control tokens related to sentence length should bring a change in performance. Thus, *CharLengthRatio* and *DependencyTreeDepthRatio* should be of equal importance to the other tokens, which is confirmed by the average improvement in SARI across single control tokens compared to that of *WLR*, *LR*,

| Control Tokens | | | | | Scores | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CLR | WLR | LR | WRR | DTDR | SARI↑ | Add↑ | Keep↑ | Delete↑ | BLEU↑ | FKGL↓ |
| **0.7** | **0.6** | **0.6** | **0.55** | **0.75** | **37.40** | **2.35** | **53.90** | **55.93** | 80.88 | 7.92 |
| 0.7 | 0.6 | 0.6 | 0.55 | - | 37.36 | 2.51 | 53.00 | 57.00 | 81.15 | 7.84 |
| 0.7 | 0.6 | 0.6 | - | - | 35.08 | 1.89 | 54.98 | 48.36 | 81.15 | 7.84 |
| 0.7 | 0.6 | - | - | - | 37.32 | 2.18 | 52.62 | 57.16 | 78.98 | 7.62 |
| 0.7 | - | - | - | - | 32.88 | 1.67 | 55.83 | 41.13 | 85.04 | 9.21 |
| - | 0.6 | - | - | - | 34.02 | 1.85 | 55.28 | 44.94 | 85.28 | 8.96 |
| - | - | 0.6 | - | - | 34.20 | 1.86 | 54.85 | 45.90 | 84.19 | 8.86 |
| - | - | - | 0.55 | - | 34.46 | 1.95 | 54.47 | 46.98 | 81.82 | 8.77 |
| - | - | - | - | 0.75 | 32.34 | 1.51 | 56.24 | 39.29 | 85.78 | 9.46 |

Table 13: Relative feature importance of the best result and each single control token.

| | |
|---|---|
| Zijn volgende werk, zaterdag, volgt op een bijzonder veelbewogen dag in het leven van een succesvolle neurochirurg. **Het is een bijzonder veelbewogen dag. Het is een bijzonder veelbewogen dag.** | 1 |

Table 14: An example of sentences generated with wrong meaning after control token training.

and *WRR* (see Table 13). The test dataset also contains a high degree of substitution, replacing complicated words with simpler words (Zhao et al. 2022).

Consequently, the token *WordRankRatio* should significantly affect the final simplification result. In their paper, Sheang and Saggion (2021) noted that the addition of the *WordLengthRatio* token fulfills its purpose of replacing words with shorter words. Yet adding the *WordLengthRatio* token also lowered the SARI score in their study, which is not corroborated by the results of this work.

### 5.5 Decoder-constrained Sentence Generation

Finally, the generation of simplified output is steered by constrained decoding at inference time. The test showed that the generated output of sequence-to-sequence transformer models such as T5-base is repetitive. This seems to be a flaw in the training method used, where some tokens are assumed to be more challenging to learn than others (Jiang et al. 2020) or a mistake in the corresponding probability distribution (Welleck et al. 2019). This is also the case with the current model. An exemplary output sentence is shown in Table 14.

The model's repetitiveness could not be remedied by additional padding at encoding and decoding time nor by limiting the length of the output sequence. Given that the complex sentences are of varying size, this is also the case at inference time, and the simplified output sentences vary in length. As of now, the generation of several output sentences could be helped by setting the end of sentence token equal to 4, which corresponds to a dot (".") and defines the end of a sentence. Yet this solution is error-prone because any sentence containing a dot will be cut off directly afterward. To find the proper control mechanism, several decoding strategies have been assessed, among which are greedy decoding, a combination of top-p and top-k search, as well as beam search decoding. A hyperparameter search with various combinations for each decoding strategy has been executed to evaluate the best decoding strategy. In the first evaluation step, each decoder output for a given parameter combination has been assessed manually on its factuality and grammaticality. The set of determined parameters during the search for all three decoding strategies is listed in Table 15.

Several parameter configurations have been evaluated for greedy decoding, with the final set of parameters listed in Table 15. The generated output of greedy decoding varies highly in quality and can be factually wrong, as demonstrated in Table 16. With top-p and top-k sampling, the outcome sentences were often incoherent and nonsense, where hallucinated words were added to the sentence that had not been part of the source sentence. Therefore, the combination of top-p and

| Decoding parameters | do_ sample | top-k | top-p | num_ beams | repetition penalty | early stopping | max_length | min_length |
|---|---|---|---|---|---|---|---|---|
| Greedy decoding | False | – | – | – | – | – | 50 | 3 |
| Top-p & top-k sampling | True | 5 | 0.98 | – | – | – | 50 | 3 |
| Beam search | False | – | – | 8 | 1.2 | True | 50 | 3 |

Table 15: Decoding parameters for the three decoding strategies.

| | | |
|---|---|---|
| Greedy decoding | Ze zijn cultureel verwant aan **Papoea-Nieuw-Guinea.** | 13 |
| | Reizen door zeer afgelegen gebieden, op afgelegen wegen, vereist echter planning vooraf. | 14 |
| | **Hij speelt voor het nationale team van Italië.** | 15 |
| Top-p & top-k sampling | Ze zijn cultureel verwant aan **Papoea-Nieuw-Guinea.** | 13 |
| | Het vereist echter voorbereiding vooraf en een **goed** voertuig. | 14 |
| | **In 1982 werd hij ontslagen uit het nationale team van Italië.** | 15 |
| Beam search | Ze zijn cultureel verwant aan **Papoea-Nieuw-Guinea**. | 13 |
| | Reizen door zeer afgelegen gebieden, op afgelegen wegen, vereist echter planning vooraf. | 14 |
| | **Hij speelt voor het nationale team van Italië.** | 15 |

Table 16: An example of sentences generated with the three decoding methods. Incorrect transformations (lost or changed meaning) are marked in bold.

top-k sampling should not be considered. However, sampling methods often produce repetitive and gibberish output (Holtzman et al. 2019). The decoding strategy with beam search generated a stable result with fewer hallucinations. Multiple configurations have been assessed, with the final set of parameters listed in Table 15.

In the next step, the set of 84 randomly sampled sentences (see section 4.4) from the test dataset was taken to generate text based on each of the three decoding strategies. Then, the sentences were manually compared to their original sentence and scored on factuality (Devaraj et al. 2022) and grammaticality. The result of the comparison is shown in Table 17.

Manual inspection showed that greedy decoding suffers from an extreme shortening of sentences where subclauses are mostly cut off. Greedy decoding, however, effectively replaces complex words such as “onmisbaar” with “belangrijk”. The result from the decoder with a combination of top-p and top-k sampling revealed more hallucinations than other decoders, yet provided factually correct summarization of facts, especially in subclauses. The decoding strategy with beam search mainly consisted of nearly identical copying of sentences where some minor words are replaced. An extract of output sentences is shown in **Appendix C**. To conclude, no decoding strategy provides error-free results. Beam search should be chosen for a solution that is factually primarily correct. For a solution that offers significant shortening, greedy decoding is most suitable.

| | Factuality | | | Grammaticality | |
|---|---|---|---|---|---|
| Decoding method | Correct | Wrong | Hallucination | Correct | Wrong |
| Greedy decoding | 69 | 14 | 1 | 81 | 3 |
| Top-p & top-k sampling | 57 | 24 | 3 | 82 | 2 |
| Beam search | 78 | 6 | 0 | 83 | 1 |

Table 17: Comparison of sampled simplifications between three decoding methods on factuality and grammaticality.

| Data | SARI↑ | Add↑ | Keep↑ | Delete↑ | FKGL↓ | BLEU↑ |
|---|---|---|---|---|---|---|
| Identity baseline | 19.79 | 24.72 | 59.38 | 0.0 | 10.75 | 90.11 |
| Trained baseline | 26.04 | 3.35 | 59.30 | 15.47 | 6.27 | 48.32 |
| Trained baseline with control tokens | 37.40 | 2.35 | 53.90 | 55.93 | 80.88 | 7.92 |
| Greedy decoding | 36.26 | 2.05 | 54.93 | 51.77 | 8.30 | 83.37 |
| Top-p & top-k sampling | 38.04 | 3.26 | 49.56 | 61.30 | 7.84 | 66.16 |
| Beam search | 36.85 | 2.28 | 54.14 | 54.15 | 8.05 | 83.38 |
| Reference baseline | 53.20 | 24.72 | 62.94 | 71.95 | 7.28 | 100.00 |

Table 18: Output generated with different decoding strategies compared with the identity and reference baseline.

# 6. Results

## 6.1 Evaluation of Dutch Sentence Simplification

After fine-tuning on Dutch data, the best model is the T5 base model for Dutch, which has been trained with an Adafactor optimizer for four epochs with a training set of 10000 rows (trained baseline in Table 18). After hyperparameter search, the trained model plus all tokens $CLR = 0.7$, $WLR = 0.6$, $LR = 0.6$, $WRR = 0.55$, $DTDR = 0.75$ performs best on the test set with a resulting SARI score of 37.40 (trained baseline with control tokens in Table 18). Therefore, adding control tokens significantly improves the performance on SARI (+11.27), and adding all tokens is best. It needs to be mentioned that a complete evaluation of the simplification output on the SARI score alone is insufficient. For example, although the decoder with top-p and top-k sampling scores high in SARI (38.04), its simplifications are factually incorrect. In this case, a result with a lower SARI ranking is favored instead of a higher one. In conclusion, simplification results should not only be chosen by their SARI score but also require evaluation of factuality and grammaticality. The decoder that generates (factually) correct results is beam search followed by greedy decoding, where most (but not all) results are accurate. The final output by beam search generates a 36.85 SARI score, and the output with greedy decoding scores a 36.26 score on the test set.

For further benchmarking, we compare the final simplification results to the identity baseline and the reference baseline. The identity baseline takes the original sequence as a simplification, and the reference baseline is a randomly chosen reference simplification from the reference simplifications of the test set. All simplification results score better than the identity baseline but worse when compared to the reference baseline (reference set no. 4) when compared with SARI. The results are shown in Table 18.

A similar approach in English language (Sheang and Saggion 2021) shows that the T5 base model performs best on the final token set of *CharLengthRatio, WordLengthRatio, LevenshteinRatio, WordRankRatio*, and *DependencyTreeDepthRatio* with a SARI score of 45.04 on the original ASSET dataset. The best token values in their study are $CLR = 0.95$, $WLR = 0.5$, $LR = 0.75$, $WRR = 0.75$, $DTDR = 0.75$. For comparison, this combination generates an output with a SARI score of 32.22 if the ASSET dataset is translated into Dutch.

Menta and Garcia-Serrano (2022) report a SARI value of 37.40 on a T5-small model with a combination of the following tokens $CLR = 0.6$, $WLR = 0.75$, $LR = 0.6$, $DTDR = 0.95$ but replace the *WordRankRatio* token with a *Language-Fill Mask* token ($LMFMR = 0.75$) for simple word prediction. A similar combination with the base Dutch data of this work and a *WRR* token value of 0.75 instead of the *LMFMR* token generates an output that scores 33.38 in SARI on the original ASSET dataset. To conclude, the control token values for Dutch language differ from token values for English language, suggesting that these are language dependent. Unfortunately, other simplification approaches in Dutch do not report a SARI score; a comparison to former simplification approaches in Dutch is not possible.

### 6.2 Discussion

Finally, the study was limited in several aspects. First, text generation with transformer models such as T5 faces the issue that text generation is highly repetitive. The reasons for repetitive generation remain unclear, and research in this field is still in its infancy. One approach to mitigating repetitive generation is modifying the training loss (Welleck et al. 2019). To prevent repetitive outcomes, an end-of-sentence token was introduced. This leads to the resulting model not handling splits well and only returning one sentence simplification each.

Furthermore, the system's quality depends on the training method and data on which the model has been trained. The base data for the training dataset (WikiLarge) and the test dataset (ASSET) differ to some extent. As such, the split ratio in WikiLarge is much lower (0.1) than in ASSET (0.3) (Zhao et al. 2022). It can be argued that the used training set has limited suitability for sentence simplification, given that it is found to be noisy (Xu et al. 2016) and contains many poorly aligned sentence pairs (Zhao et al. 2022). The original (English) training dataset is also known to drop words and apply drastic shortening in simplified sentences.

Contrary to this, the test dataset was found to be of high quality (Vásquez-Rodríguez et al. 2021, Zhao et al. 2022), containing sentence splitting, paraphrasing, and a high degree of compression (Alva-Manchego et al. 2020a). Overall, structurally similar datasets would have been more suitable. Hence the model and control tokens would not have been trained and evaluated on differing datasets. Moreover, since the pre-training data consisted of Dutch news, no prior pre-training for domain-specific words was needed. In the hypothetical case that an existing domain shift between pre-training data and fine-tuning data has been overlooked, pre-training in the style of (Maruyama and Yamamoto 2019) could help towards better fine-tuning performance.

Additionally, this work has shown that a minimum of 2000 rows of parallel data are required to provide mediocre results, and 10,000 rows of data are required for better results. Given that this work was based on translation data, a high-quality parallel dataset containing at least 2000 parallel sentence pairs is needed to further Dutch sentence simplification. With the absence of the latter, further research could assess a fully unsupervised approach by applying techniques that do not require parallel data or focus on synthetic parallel corpus creation. Moreover, NMT engines such as Google Translate stick to the syntactic structure of their source sentence (Webster et al. 2020). This creates a limitation: typical Dutch syntactic constructions might not be present in the translated dataset, and hence the system might not perform well for sentences with typical Dutch syntactic constructions. As a consequence, further research should include datasets with original Dutch syntax.

The implemented explicit control tokens cover the features that have been found to make the differentiation between standard and easy texts in Dutch (Vandeghinste and Bulté 2019), have been stated as central features in sentence simplification (Shardlow 2014), and implemented in prior controllable simplified sentence generation (Martin et al. 2020, Menta and Garcia-Serrano 2022, Sheang and Saggion 2021). The control token values for Dutch language in this study differ from token values for English language, suggesting that these values are language dependent. Further research should identify whether there are even more suitable control tokens for Dutch. This work deployed hyperparameter search to identify optimal control token values. An area of further research is the identification of precomputed control token values based on the input text and their automated use/inclusion into training. An isolated test of each control token (see Table 13) has shown that each control token contributes to sentence simplification. However, the individual effect of each control token is hard to quantify, given that there is not only one "correct" reference simplification which, in turn, makes the deduction of a perfect set of control tokens in hyperparameter search difficult. In that sense, evaluating generated outputs solely on SARI can be misleading, as it does not account for grammaticality and overall meaning preservation.

The authors did not have access to resources for a human-based evaluation of simplification results. Hence, this work relies solely on automatic evaluation metrics. However, automatic evaluation

metrics have limited suitability for assessing sentence simplification as discussed in section 5.1. An inclusion of human-based simplification evaluation in further research is desirable.

Moreover, tests have shown that the effect of the *WordLengthRatio* is only effective with very low token values (0.2). Concerning word frequency estimation in the *WordRankRatio* token, compound words in Dutch (Macken and Tezcan 2018, Pander Maat et al. 2014) remain an obstacle. One option could be to split complex compound words into their components (Macken and Tezcan 2018, Vandeghinste 2002), so the frequency of these split words could be determined more quickly and return a complexity score based on more matches in the frequency database. On the other hand, the resulting complexity score falsifies the overall result.

## 7. Conclusions

We conditioned a Dutch sequence-to-sequence model (T5) on sentence simplification in this work. The primary focus was on incorporating control tokens into the training data to enable the generation of simplified text. These control tokens targeted various attributes, including sentence compression, word length, paraphrasing, and lexical and syntactic complexity. Furthermore, the values of these control tokens are adjustable, allowing customization according to the requirements of diverse target audiences. This adaptability allows to cater to a range of users with varying levels of language proficiency or specific simplification needs.

To evaluate the system performance, synthetic datasets for Dutch were created. These datasets were domain-general and created from datasets commonly used in English sentence simplification tasks. The goal was to assess the model's capabilities when trained and tested on these Dutch-specific datasets. The results clarified the data requirements for achieving satisfactory outcomes: Mediocre results necessitated approximately 2000 rows of parallel data, while better performance was attainable with a larger dataset of 10000 rows. This information is especially valuable for estimating the data collection efforts for future model applications and shows that synthetically generated data can be used for training and testing.

In general, the evaluation of the model's performance on the test dataset revealed promising results, with a SARI score of 37.40 after adjusting the control tokens. In the final configuration, the system achieved a slightly lower SARI score of 36.85, depending on the parameter configuration of the decoder. Although the proposed approach did not outperform its own reference baseline, it demonstrated the potential for sentence simplification tasks. Moreover, the simplification results indicate that evaluating generated outputs solely on SARI can be misleading, as it does not account for grammaticality and overall meaning preservation.

To conclude, all control tokens play a vital role in enhancing the simplicity of the generated output, which led to the identification of a specific set of control tokens tailored for the Dutch language. However, the individual effect of each control token is hard to quantify, assuming the effectiveness of the implemented control tokens at varying levels and token values. Further research should identify whether there are even more suitable control tokens for Dutch. This work provides insights into applying supervised training on parallel data based on large language models for sentence simplification in Dutch language. Furthermore, the results contribute to understanding data requirements and the impact of control tokens for simplification systems, thereby paving the way for future advancements in Dutch language simplification.

## References

Aiken, Milam (2019), An updated evaluation of google translate accuracy, *Studies in Linguistics and Literature* **3**, pp. p253.

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019), Optuna: A next-generation hyperparameter optimization framework, *Proceedings of the 25th ACM*

*SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631.

Aluisio, Sandra, Lucia Specia, Thiago Pardo, Erick Maziero, and Renata Fortes (2008), Towards brazilian portuguese automatic text simplification systems, pp. 240–248.

Alva-Manchego, Fernando, Carolina Scarton, and Lucia Specia (2020a), Data-driven sentence simplification: Survey and benchmark, *Computational Linguistics* **46** (1), pp. 135–187, MIT Press, Cambridge, MA. https://aclanthology.org/2020.cl-1.4.

Alva-Manchego, Fernando, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia (2020b), ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 4668–4679. https://aclanthology.org/2020.acl-main.424.

Alva-Manchego, Fernando, Louis Martin, Carolina Scarton, and Lucia Specia (2019), EASSE: Easier automatic sentence simplification evaluation, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, Association for Computational Linguistics, Hong Kong, China, pp. 49–54. https://aclanthology.org/D19-3009.

Bergstra, James, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl (2011), Algorithms for hyper-parameter optimization, *in* Shawe-Taylor, J., R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 24, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Biewald, L. (2020), Experiment tracking with weights and biases. software available from wandb.com. https://www.wandb.com/.

Brouwer, R.H.M. (1963), Onderzoek naar de leesmoeilijkheden van Nederlands proza, *Pedagogische Studiën* **40**, pp. 454–464.

Bulté, Bram, Leen Sevens, and Vincent Vandeghinste (2018), Automating lexical simplification in Dutch, *Computational Linguistics in the Netherlands Journal* **8**, pp. 24–48. https://clinjournal.org/clinj/article/view/78.

Cardon, Rémi and Natalia Grabar (2018), Identification of parallel sentences in comparable monolingual corpora from different registers, *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, Association for Computational Linguistics, pp. 83–93. http://aclweb.org/anthology/W18-5610.

Carroll, John, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait (1998), Practical simplification of english newspaper text to assist aphasic readers, *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*.

Chandrasekar, R., Christine Doran, and B. Srinivas (1996), Motivations and methods for text simplification, *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*. https://aclanthology.org/C96-2183.

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014), Learning phrase representations using RNN encoder–decoder for statistical machine translation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734. https://aclanthology.org/D14-1179.

Coster, Will and David Kauchak (2011), Learning to simplify sentences using Wikipedia, *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, Association for Computational Linguistics, Portland, Oregon, pp. 1–9. https://aclanthology.org/W11-1601.

Daelemans, Walter, Anja Höthker, and Erik Tjong Kim Sang (2004), Automatic sentence simplification for subtitling in Dutch and English, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, European Language Resources Association (ELRA), Lisbon, Portugal. http://www.lrec-conf.org/proceedings/lrec2004/pdf/697.pdf.

Dehghan, Mohammad, Dhruv Kumar, and Lukasz Golab (2022), GRS: Combining generation and revision in unsupervised sentence simplification, *Findings of the Association for Computational Linguistics: ACL 2022*, Association for Computational Linguistics, Dublin, Ireland, pp. 949–960. https://aclanthology.org/2022.findings-acl.77.

Devaraj, Ashwin, William Sheffield, Byron Wallace, and Junyi Jessy Li (2022), Evaluating factuality in text simplification, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, pp. 7331–7345. https://aclanthology.org/2022.acl-long.506.

Dodge, Jesse, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner (2021), Documenting large webtext corpora: A case study on the colossal clean crawled corpus, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 1286–1305. https://aclanthology.org/2021.emnlp-main.98.

Drndarević, Biljana and Horacio Saggion (2012), Towards automatic lexical simplification in Spanish: An empirical study, *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, Association for Computational Linguistics, Montréal, Canada, pp. 8–16. https://aclanthology.org/W12-2202.

Fan, Angela, Mike Lewis, and Yann Dauphin (2018), Hierarchical neural story generation, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Melbourne, Australia, pp. 889–898. https://aclanthology.org/P18-1082.

Fares, Murhaf, Andrey Kutuzov, Stephan Oepen, and Erik Velldal (2017), Word vectors, reuse, and replicability: Towards a community repository of large-text resources, *Proceedings of the 21st Nordic Conference on Computational Linguistics*, Association for Computational Linguistics, Gothenburg, Sweden, pp. 271–276. https://aclanthology.org/W17-0237.

Févry, Thibault and Jason Phang (2018), Unsupervised sentence compression using denoising auto-encoders, *Proceedings of the 22nd Conference on Computational Natural Language Learning*, Association for Computational Linguistics, Brussels, Belgium, pp. 413–422. https://aclanthology.org/K18-1040.

Filippova, Katja and Michael Strube (2008), Dependency tree based sentence compression, *Proceedings of the Fifth International Natural Language Generation Conference*, pp. 25–32.

Galeev, Farit, Marina Leushina, and Vladimir Ivanov (2021), rubts: Russian sentence simplification using back-translation, *Proc. Computational Linguistics and Intellectual Tech* pp. 1–8.

Ghalandari, Demian Gholipour, Chris Hokamp, and Georgiana Ifrim (2022), Efficient unsupervised sentence compression by fine-tuning transformers with reinforcement learning, *arXiv preprint arXiv:2205.08221*.

Glavaš, Goran and Sanja Štajner (2013), Event-centered simplification of news stories, *Proceedings of the Student Research Workshop associated with RANLP 2013*, INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, pp. 71–78. https://aclanthology.org/R13-2011.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov (2018), Learning word vectors for 157 languages, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan. https://aclanthology.org/L18-1550.

Heilman, Michael and Noah A. Smith (2010), Good question! statistical ranking for question generation, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Los Angeles, California, pp. 609–617. https://aclanthology.org/N10-1086.

Holtzman, Ari, Jan Buys, Maxwell Forbes, and Yejin Choi (2019), The curious case of neural text degeneration, *CoRR*. http://arxiv.org/abs/1904.09751.

Hwang, William, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu (2015), Aligning sentences from standard Wikipedia to Simple Wikipedia, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Denver, Colorado, pp. 211–217. https://aclanthology.org/N15-1022.

Jiang, Shaojie, Thomas Wolf, Christof Monz, and Maarten Rijke (2020), Tldr: Token loss dynamic reweighting for reducing repetitive utterance generation, *CoRR*. https://doi.org/https://doi.org/10.48550/arXiv.2003.11963.

Kajiwara, Tomoyuki and Mamoru Komachi (2018), Text simplification without simplified corpora, *Journal of Natural Language Processing* **25**, pp. 223–249.

Keskar, Nitish Shirish, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher (2019), Ctrl: A conditional transformer language model for controllable generation.

Kincaid, J Peter, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom (1975), Derivation of new readability formulas for navy enlisted personnel, *Technical report*, Institute for Simulation and Training, University of Central Florida.

Kudo, Taku and John Richardson (2018), SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Brussels, Belgium, pp. 66–71. https://aclanthology.org/D18-2012.

Kumar, Dhruv, Lili Mou, Lukasz Golab, and Olga Vechtomova (2020), Iterative edit-based unsupervised sentence simplification, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 7918–7928. https://aclanthology.org/2020.acl-main.707.

Laban, Philippe, Tobias Schnabel, Paul Bennett, and Marti A. Hearst (2021), Keep it simple: Unsupervised simplification of multi-paragraph text, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, pp. 6365–6378. https://aclanthology.org/2021.acl-long.498.

Lee, John and Chak Yan Yeung (2018), Personalizing lexical simplification, *Proceedings of the 27th International Conference on Computational Linguistics*, Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 224–232. https://aclanthology.org/C18-1019.

Levenshtein, Vladimir I. (1965), Binary codes capable of correcting deletions, insertions, and reversals, *Soviet physics. Doklady* **10**, pp. 707–710. https://api.semanticscholar.org/CorpusID:60827152.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020), BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 7871–7880. https://aclanthology.org/2020.acl-main.703.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer (2020), Multilingual denoising pre-training for neural machine translation, *Transactions of the Association for Computational Linguistics* **8**, pp. 726–742, MIT Press, Cambridge, MA. https://aclanthology.org/2020.tacl-1.47.

Loshchilov, Ilya and Frank Hutter (2017), Fixing weight decay regularization in adam, *CoRR*. http://arxiv.org/abs/1711.05101.

Macken, Lieve and Arda Tezcan (2018), Dutch compound splitting for bilingual terminology extraction, *in* Mitkov, Ruslan and Monti, Johanna and Corpas Pastor, Gloria and Seretan, Violeta, editor, *Multiword units in machine translation and translation technology*, Vol. 341 of *Current Issues in Linguistic Theory*, John Benjamins, pp. 148–162. http://doi.org/10.1075/cilt.341.07mac.

Maddela, Mounica, Fernando Alva-Manchego, and Wei Xu (2021), Controllable text simplification with explicit paraphrasing, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, pp. 3536–3553. https://aclanthology.org/2021.naacl-main.277.

Martin, Louis, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot (2022), MUSS: Multilingual unsupervised sentence simplification by mining paraphrases, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 1651–1664. https://aclanthology.org/2022.lrec-1.176.

Martin, Louis, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes (2020), Controllable sentence simplification, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 4689–4698. https://aclanthology.org/2020.lrec-1.577.

Martin, Louis, Samuel Humeau, Pierre-Emmanuel Mazaré, Éric de La Clergerie, Antoine Bordes, and Benoît Sagot (2018), Reference-less quality estimation of text simplification systems, *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, Association for Computational Linguistics, Tilburg, the Netherlands, pp. 29–38. https://aclanthology.org/W18-7005.

Maruyama, Takumi and Kazuhide Yamamoto (2019), Extremely low resource text simplification with pre-trained transformer language model, *2019 International Conference on Asian Language Processing (IALP)*, pp. 53–58.

Menta, Antonio and Ana Garcia-Serrano (2022), Controllable Sentence Simplification Using Transfer Learning, *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, CEUR Workshop Proceedings, CEUR-WS.org, Bologna, Italy.

Narayan, Shashi and Claire Gardent (2014), Hybrid simplification using deep semantics and machine translation, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Baltimore, Maryland, pp. 435–445. https://aclanthology.org/P14-1041.

Narayan, Shashi, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina (2017), Split and rephrase, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 606–616. https://aclanthology.org/D17-1064.

Nisioi, Sergiu, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu (2017), Exploring neural text simplification models, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Vancouver, Canada, pp. 85–91. https://aclanthology.org/P17-2014.

Paetzold, Gustavo and Lucia Specia (2016a), SemEval 2016 task 11: Complex word identification, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Association for Computational Linguistics, San Diego, California, pp. 560–569. https://aclanthology.org/S16-1085.

Paetzold, Gustavo and Lucia Specia (2016b), Unsupervised lexical simplification for non-native speakers, *Proceedings of the AAAI Conference on Artificial Intelligence*. https://ojs.aaai.org/index.php/AAAI/article/view/9885.

Palmero Aprosio, Alessio, Sara Tonelli, Marco Turchi, Matteo Negri, and Mattia A. Di Gangi (2019), Neural text simplification in low-resource conditions using weak supervision, *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 37–44. https://aclanthology.org/W19-2305.

Pander Maat, Henk, Rogier Kraf, Antal van den Bosch, Nick Dekker, Maarten van Gompel, Suzanne Kleijn, Ted Sanders, and Ko van der Sloot (2014), T-scan: a new tool for analyzing Dutch text, *Computational Linguistics in the Netherlands Journal* **4**, pp. 53–74. https://clinjournal.org/clinj/article/view/40.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002), Bleu: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp. 311–318. https://aclanthology.org/P02-1040.

Popović, Maja (2015), chrF: character n-gram F-score for automatic MT evaluation, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Association for Computational Linguistics, Lisbon, Portugal, pp. 392–395. https://aclanthology.org/W15-3049.

Popović, Maja (2017), chrF++: words helping character n-grams, *Proceedings of the Second Conference on Machine Translation*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 612–618. https://aclanthology.org/W17-4770.

Post, Matt (2018), A call for clarity in reporting BLEU scores, *Proceedings of the Third Conference on Machine Translation: Research Papers*, Association for Computational Linguistics, Brussels, Belgium, pp. 186–191. https://aclanthology.org/W18-6319.

Qiang, Jipeng, Feng Zhang, Yun Li, Yunhao Yuan, Yi Zhu, and Xindong Wu (2022), Unsupervised statistical text simplification using pre-trained language modeling for initialization, *Frontiers of Computer Science*. https://doi.org/10.1007/s11704-022-1244-0.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020), Exploring the limits of transfer learning with a unified text-to-text transformer, *J. Mach. Learn. Res.*, JMLR.org.

Rashid, Mohammad Amin and Hossein Amirkhani (2023), Improving edit-based unsupervised sentence simplification using fine-tuned bert, *Pattern Recognition Letters* **166**, pp. 112–118. https://www.sciencedirect.com/science/article/pii/S0167865523000168.

Rello, Luz, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion (2013), Frequent words improve readability and short words improve understandability for people with dyslexia, *in* Kotzé, Paula, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2013*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 203–219.

Reusens, Manon, Michael Reusens, Marc Callens, Seppe vanden Broucke, and Bart Baesens (2022), Comparison of different modeling techniques for flemish twitter sentiment analysis, *Analytics* **1** (2), pp. 117–134. https://www.mdpi.com/2813-2203/1/2/9.

Ruitenbeek, Ward, Victor Zwart, Robin Van Der Noord, Zhenja Gnezdilov, and Tommaso Caselli (2022), "zo grof !": A comprehensive corpus for offensive and abusive language in Dutch, *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, Association for Computational Linguistics, Seattle, Washington (Hybrid), pp. 40–56. https://aclanthology.org/2022.woah-1.5.

Saggion, H. (2017), *Automatic text simplification*, Synthesis lectures on human language technologies (Vol. 10 (1)), Morgan & Claypool Publishers.

Sakhovskiy, Andrey, Alexandra Izhevskaya, Alena Pestova, E. Tutubalina, Valentin Malykh, Ivan Smurov, and E. Artemova (2021), Rusimplesenteval-2021 shared task: Evaluating sentence simplification for russian. https://api.semanticscholar.org/CorpusID:240149260.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde (2018), Less is more: A rule-based syntactic simplification module for improved text-to-pictograph translation, *Data and Knowledge Engineering*.

Shao, Yuanlong, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil (2017), Generating high-quality and informative conversation responses with sequence-to-sequence models, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 2210–2219. https://aclanthology.org/D17-1235.

Shardlow, Matthew (2014), A survey of automated text simplification, *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing 2014*, The Science and Information Organization. http://dx.doi.org/10.14569/SpecialIssue.2014.040109.

Shazeer, Noam and Mitchell Stern (2018), Adafactor: Adaptive learning rates with sublinear memory cost.

Sheang, Kim Cheng and Horacio Saggion (2021), Controllable sentence simplification with a unified text-to-text transfer transformer, *Proceedings of the 14th International Conference on Natural Language Generation*, Association for Computational Linguistics, Aberdeen, Scotland, UK, pp. 341–352. https://aclanthology.org/2021.inlg-1.38.

Siddharthan, Advaith (2006), Syntactic simplification and text cohesion, *Research on Language and Computation* **4**, pp. 77–109, Springer.

Siddharthan, Advaith, Ani Nenkova, and Kathleen McKeown (2004), Syntactic simplification for improving content selection in multi-document summarization, *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, COLING, Geneva, Switzerland, pp. 896–902. https://aclanthology.org/C04-1129.

Snover, Matthew, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul (2006), A study of translation edit rate with targeted human annotation, *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, Association for Machine Translation in the Americas, Cambridge, Massachusetts, USA, pp. 223–231. https://aclanthology.org/2006.amta-papers.25.

Specia, Lucia (2010), Translating from complex to simplified sentences, *in* Pardo, Thiago Alexandre Salgueiro, António Branco, Aldebaro Klautau, Renata Vieira, and Vera Lúcia Strube de Lima, editors, *Computational Processing of the Portuguese Language*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 30–39.

Stajner, Sanja (2021), Automatic text simplification for social good: Progress and challenges, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, Online, pp. 2637–2652. https://aclanthology.org/2021.findings-acl.233.

Surya, Sai, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan (2019), Unsupervised neural text simplification, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp. 2058–2068. https://aclanthology.org/P19-1198.

Taylor, Zachary W., Maximus H. Chu, and Junyi Jessy Li (2022), Text simplification of college admissions instructions: A professionally simplified and verified corpus, *Proceedings of the 29th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, pp. 6505–6515. https://aclanthology.org/2022.coling-1.566.

Ten Oever, Sanne and Andrea E Martin (2021), An oscillating computational model can track pseudo-rhythmic speech by using linguistic predictions, *Elife* **10**, pp. e68066, eLife Sciences Publications Limited.

Tulkens, Stéphan, Chris Emmery, and Walter Daelemans (2016), Evaluating unsupervised Dutch word embeddings as a linguistic resource, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, European Language Resources Association (ELRA), Portorož, Slovenia, pp. 4130–4136. https://aclanthology.org/L16-1652.

Vandeghinste, Vincent (2002), Lexicon optimization: Maximizing lexical coverage in speech recognition through automated compounding, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, European Language Resources Association (ELRA), Las Palmas, Canary Islands - Spain. http://www.lrec-conf.org/proceedings/lrec2002/pdf/4.pdf.

Vandeghinste, Vincent and Bram Bulté (2019), Linguistic Proxies of Readability: Comparing Easy-to-Read and regular newspaper Dutch, *Computational Linguistics in the Netherlands Journal* **9**, pp. 81–100. https://clinjournal.org/clinj/article/view/97.

Vandeghinste, Vincent and Yi Pan (2004), Sentence compression for automated subtitling: A hybrid approach, *Text Summarization Branches Out*, Association for Computational Linguistics, Barcelona, Spain, pp. 89–95. https://aclanthology.org/W04-1015.

Vandeghinste, Vincent, Bram Bulté, and Liesbeth Augustinus (2019), Wablieft: An easy-to-read newspaper corpus for Dutch, *Proceedings of the CLARIN Annual Conference*, Leipzig, Germany, pp. 188–191.

Vásquez-Rodríguez, Laura, Matthew Shardlow, Piotr Przybyła, and Sophia Ananiadou (2021), Investigating text simplification evaluation, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, Online, pp. 876–882. https://aclanthology.org/2021.findings-acl.77.

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

Vickrey, David and Daphne Koller (2008), Sentence simplification for semantic role labeling, *Proceedings of ACL-08: HLT*, Association for Computational Linguistics, Columbus, Ohio, pp. 344–352. https://aclanthology.org/P08-1040.

Vijayakumar, Ashwin K., Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra (2016), Diverse beam search: Decoding diverse solutions from neural sequence models, *CoRR*. http://arxiv.org/abs/1610.02424.

Watanabe, Willian, Arnaldo Junior, Vinícius Uzêda, Renata Fortes, Thiago Pardo, and Sandra Aluisio (2009), Facilita: Reading assistance for low-literacy readers, pp. 29–36.

Webster, Rebecca, Margot Fonteyne, Arda Tezcan, Lieve Macken, and Joke Daems (2020), Gutenberg goes neural: Comparing features of dutch human translations with raw neural machine translation outputs in a corpus of english literary classics, *Informatics*. https://www.mdpi.com/2227-9709/7/3/32.

Welleck, Sean, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston (2019), Neural text generation with unlikelihood training, *CoRR*. http://arxiv.org/abs/1908.04319.

Wenzek, Guillaume, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave (2020), CCNet: Extracting high quality monolingual datasets from web crawl data, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 4003–4012. https://aclanthology.org/2020.lrec-1.494.

Wilkens, Rodrigo, Alessandro Dalla Vecchia, Marcely Zanon Boito, Muntsa Padró, and Aline Villavicencio (2014), Size does not matter. frequency does. a study of features for measuring lexical complexity, *in* Bazzan, Ana L.C. and Karim Pichara, editors, *Advances in Artificial Intelligence – IBERAMIA 2014*, Springer International Publishing, Cham, pp. 129–140.

Williams, Ronald J. and David Zipser (1989), A learning algorithm for continually running fully recurrent neural networks, *Neural Computation* **1** (2), pp. 270–280.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush (2020), Transformers: State-of-the-art

natural language processing, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online, pp. 38–45. https://aclanthology.org/2020.emnlp-demos.6.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean (2016), Google's neural machine translation system: Bridging the gap between human and machine translation, *CoRR*. http://arxiv.org/abs/1609.08144.

Wubben, Sander, Antal van den Bosch, and Emiel Krahmer (2010), Paraphrase generation as monolingual translation: Data and evaluation, *Proceedings of the 6th International Natural Language Generation Conference*, Association for Computational Linguistics. https://aclanthology.org/W10-4223.

Wubben, Sander, Antal van den Bosch, and Emiel Krahmer (2012), Sentence simplification by monolingual machine translation, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Jeju Island, Korea, pp. 1015–1024. https://aclanthology.org/P12-1107.

Xu, Wei, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch (2016), Optimizing statistical machine translation for text simplification, *Transactions of the Association for Computational Linguistics* **4**, pp. 401–415, MIT Press, Cambridge, MA. https://aclanthology.org/Q16-1029.

Yang, Yang, Juan Cao, Yujun Wen, and Pengzhou Zhang (2021), Table-to-text generation with accurate content copying. https://doi.org/10.1038/s41598-021-00813-6.

Zhang, Xingxing and Mirella Lapata (2017), Sentence simplification with deep reinforcement learning, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 584–594. https://aclanthology.org/D17-1062.

Zhao, Sanqiang, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto (2018), Integrating transformer and paraphrase rules for sentence simplification, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, pp. 3164–3173. https://aclanthology.org/D18-1355.

Zhao, Sanqiang, Rui Meng, Hui Su, and Daqing He (2022), Divide-and-conquer text simplification by scalable data enhancement, *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Virtual), pp. 166–172. https://aclanthology.org/2022.tsar-1.15.

Zhu, Zhemin, Delphine Bernhard, and Iryna Gurevych (2010), A monolingual tree-based translation model for sentence simplification, *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Coling 2010 Organizing Committee, Beijing, China, pp. 1353–1361. https://aclanthology.org/C10-1152.

## Appendix A. Dutch Word Embeddings

Several Dutch word embedding vocabularies have been tested. The final choice was the embedding vocabulary from Fares et al. (2017), as explained in section 4.2. In the following, the remaining evaluated word embeddings are listed.

- The Dutch fastText model (called cc.nl.300 model in the code), which contains 2,000,000 words, is trained on Common Crawl and Wikipedia data using fastText (Grave et al. 2018). The embedding is trained with CBOW using position weights and a dimension of 300. The embeddings are hosted on fastText and available under `https://fasttext.cc/docs/en/crawl-vectors.html` and referred to in scholarly work (Reusens et al. 2022).
- The Dutch word embeddings called "coosto" (called coosto model in the code) contains 250,479 words. The embeddings are derived from Dutch social media messages from news, blogs, and forum posts extracted in 2017 out of 624 million messages from 660 million texts. The data is available at: `https://github.com/coosto/dutch-word-embeddings`. The Coosto model is frequently used in scholarly work where it achieves competitive results (Reusens et al. 2022, Ruitenbeek et al. 2022, Ten Oever and Martin 2021).
- The Dutch word embeddings from Tulkens et al. (2016) (called comb_320 model in the code) contain 989,820 words from a combination of the Roularta corpus, a Wikipedia dump, the SoNaR corpus, the COW corpus, and a social media dataset (see paper for sources). Available under: `https://github.com/clips/dutchembeddings`. The embeddings are also used in scholarly work (Ten Oever and Martin 2021).

## Appendix B. Sample Set Statistics

Table 19 shows sample set statistics between the original training dataset (WikiLarge) and the test dataset (ASSET) in English and the respective machine translation and the human reference translation in Dutch. The numbers have been calculated on detokenized input over all documents. The number of characters includes spacing. Given that the original test dataset only contains 359 sentences, a sample size of 84 was chosen.

| **Sample of the training dataset** | | | |
|---|---|---|---|
| | **WikiLarge (English)** | **Machine Translation (GNMT, Dutch)** | **Human Translation (Dutch)** |
| Sentences | 106 | 107 | 108 |
| Words | 2049 | 2032 | 2035 |
| Characters (w. spaces) | 12177 | 13139 | 13049 |
| Avg. sent. length (w/sent) | 19.33 | 18.99 | 18.84 |
| **Sample of the test dataset** | | | |
| | **ASSET (English)** | **Machine Translation (GNMT, Dutch)** | **Human Translation (Dutch)** |
| Sentences | 84 | 84 | 84 |
| Words | 1389 | 1377 | 1348 |
| Characters (w. spaces) | 8339 | 9045 | 8936 |
| Avg. sent. length (w/sent) | 16.54 | 16.39 | 16.04 |

Table 19: Statistics of the sampled corpora for the training and test data

## Appendix C. Examples of Decoder Output

Table 20 shows example sentences generated by the three decoding methods tested in the experiments for decoder-constrained sentence generation in section 5.5. The corresponding reference numbers of the original sentence can be found on the right side.

| Decoding method | Output sentences | |
|---|---|---|
| Greedy decoding | Zijn volgende werk, zaterdag, volgt op een bijzonder veelbewogen dag in zijn leven. | 1 |
| | De Kate Greenaway-medaille (1999) is een onderscheiding van de Britse actrice Kate Greenaway. | 2 |
| | Fives is een sport waarvan wordt aangenomen dat deze dezelfde oorsprong heeft als veel racketsporten. | 3 |
| | Beide namen werden opgeheven in 2007 toen de naam werd veranderd. | 4 |
| | Jeddah is de belangrijkste toegangspoort tot Mekka, de heiligste stad van de islam. | 5 |
| | Veel soorten verdwenen tegen het einde van de negentiende eeuw. | 6 |
| | De 20e Perry Saturn (met Terri) versloeg Eddie Guerrero na een duikelleboog. | 7 |
| | Desalniettemin emuleerde Tagore veel stijlen, waaronder handwerk uit het noorden van New Ireland, Haida-gravures van de westkust van Canada (British Columbia) en outsneden | 8 |
| Top-p & top-k sampling | Hij werkt op een bijzonder veelbewogen dag in zijn leven. | 1 |
| | In 2000 kreeg de ontvanger van de Kate Greenaway-medaille ook de Colin Mears Award. | 2 |
| | Fives is een Britse sport die wordt beschouwd als een racketsport. | 3 |
| | Beide namen werden opgeheven in 2007 toen de naam werd veranderd. | 4 |
| | Jeddah is de belangrijkste toegangspoort tot Mekka, de heiligste stad van de islam. | 5 |
| | Veel soorten verdwenen tegen het einde van de negentiende eeuw. | 6 |
| | De Perry Saturn (met Terri) versloeg Eddie Guerrero na een duikelleboog. | 7 |
| | Tagore emuleerde veel stijlen, waaronder handwerk uit het noorden van New Ireland, Haida-gravures van de westkust van Canada (British Columbia) en sneden van Max Pechstein. | 8 |
| Beam search | Zijn volgende werk, zaterdag, volgt op een bijzonder veelbewogen dag. | 1 |
| | De winnaar van de Kate Greenaway-medaille ontvangt ook de Colin Mears Award ter waarde van £ 5000. | 2 |
| | Fives is een sport waarvan wordt aangenomen dat deze dezelfde oorsprong heeft als veel racketsporten. | 3 |
| | Beide namen werden opgeheven in 2007. | 4 |
| | Jeddah is de belangrijkste toegangspoort tot Mekka. | 5 |
| | Veel soorten verdwenen tegen het einde van de negentiende eeuw. | 6 |
| | Het WWF European Championship (8:10) Saturnus drukte Guerrero vast na een duikelleboog. | 7 |
| | Desalniettemin emuleerde Tagore tal van stijlen, waaronder handwerk uit het noorden van New Ireland, Haida-gravures van de westkust van Canada (British Columbia) en hout | 8 |

Table 20: Example sentences generated by the three decoding methods