Computational Linguistics in the Netherlands Journal 14 (2025) 21-41 Submitted 12/2024; Published 07/2025

Optimising Controllable Sentence Simplification for Dutch

Florelien Soete* FLORELIEN.SOETE@GMAIL.COM
Vincent Vandeghinste*,** VINCENT.VANDEGHINSTE@IVDNT.ORG

*KU Leuven, Belgium

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

Abstract

The concept of *Easy Language* (Vandeghinste et al. 2021) involves the use of simple text, avoiding complex grammatical constructions and difficult vocabulary. Recent approaches (Seidl and Vandeghinste 2024) have shown promising results for text simplification using the pre-trained encoder-decoder T5 model (Raffel et al. 2020). This paper investigates new control tokens with a Dutch T5 large language model, and predicts sentence-dependent control token values with BERTje (de Vries et al. 2019), based on each input instance and the desired output complexity.

Control tokens monitor the splitting and reformulation of the simplified sentence to control the degree of simplification (Sheang et al. 2022). Instead of fixed values for control tokens, the characteristics and complexity of the difficult sentences will be taken into account. Agrawal and Carpuat (2023) show that this approach improves the quality and controllability of the simplified outputs compared to using standardised control values.

Our dataset consists of selected parallel (complex-simple) sentence pairs of the LEESPLANK dataset.[1] The introduction of new control tokens has not proven to enhance the model's ability to simplify sentences. But introducing BERTje to predict the actual control token values given a complex sentence has resulted in better performances and more accurate sentence simplification.

1. Introduction

Text simplification is an NLP task aimed at making complex text easier to read and understand. It has evolved from manual methods to automated approaches using deep learning, leading to challenges like data scarcity. The main target audience includes non-native speakers and individuals with neurocognitive disorders, such as dyslexia, aphasia, autism, and intellectual disabilities. Text simplification involves modifications at the word, sentence, and text level, with *sentence simplification* specifically focusing on restructuring and reformulating sentences to improve clarity and comprehension.

Sentence simplification involves two types of simplification: lexical simplification and syntactic simplification. *Lexical simplification* is related to simplifying difficult vocabulary. *Syntactic simplification* involves changing sentence structure such as cutting long sentences into shorter segments. Both types have to preserve key ideas and approximate meanings (Saggion 2022).

Control tokens can be used to control how much splitting, rephrasing, and complexity reduction are aimed for when simplifying the sentences. By adding these tokens to the input during training, the model learns to associate them with the changes in the output. At inference time, they can be used to control the output towards the desired simplification.

This paper is a follow-up to Seidl and Vandeghinste (2024), which compared control token values for Dutch versus English sentences. It concluded that these values are language-dependent. The current study investigates how and whether the use of control tokens can be further improved for Dutch.

1. `https://huggingface.co/datasets/UWV/Leesplank_NL_wikipedia_simplifications`

While research on sentence simplification is abundant for English, research for Dutch remains scarce. We aim to address this gap by testing new control tokens for sentence simplification, specifically tailored for Dutch, and by proposing a straightforward method to predict the actual control token values given a complex sentence.

In particular, the following hypotheses will be examined:

- What is the impact of new control tokens in a supervised sentence simplification approach for a Dutch parallel dataset?
- How can the effectiveness of the existing control tokens be further optimised?
- Does the new method for predicting control tokens improve sentence simplification on a controllable sentence simplification task compared to the method using standardised values for control tokens?

The code for this paper is available from Github.[2]

2. Related Work

2.1 Easy Language for Dutch

While the terms *Easy language* and *Plain language* are often used interchangeably, they have different target audiences and goals (Vandeghinste et al. 2021). *Plain language* functions as a standard for communication between government bodies or public institutions and the general public. The International Plain Language Federation defines plain language as a type of communication that is clear, concise, and easy to understand. It allows intended readers to understand information easily and use that information effectively (Vandeghinste et al. 2021). Plain language is aimed at making information accessible to everyone, regardless of their literacy level or abilities.

Easy language, on the other hand, is specifically targeted at two distinct groups: people with low literacy and people with impairments. *Wablieft*,[3] a Belgian newspaper written in Easy language, describes Easy language as using no difficult words, jargon, or figurative speech, using terminology that everyone can understand (Vandeghinste and Bulté 2019). Easy language is designed to be even more accessible than plain language, taking into account the needs of individuals who may struggle with complex language structures or vocabulary.

2.2 Text simplification for Dutch

Research on Dutch text simplification is currently constrained by limited available resources, as noted by Bulté et al. (2018), Vandeghinste and Bulté (2019), and Seidl and Vandeghinste (2024). However, there are notable efforts to overcome these challenges and develop effective simplification techniques.

Bulté et al. (2018) implemented a lexical simplification pipeline by identifying difficult words and replacing them with simple words, creating a data-driven, automatic lexical simplification tool by evaluating the complexity of words in a text, considering two factors: the average self-estimated age of acquisition of base words (Brysbaert et al. 2014) and the word frequency.

Sevens et al. (2018) discussed a rule-based syntactic simplification module for improving text-to-pictograph translation, particularly for individuals with intellectual disabilities and proposed a hand-crafted simplification system that utilized syntactic parsing for sentence analysis.

Wubben et al. (2012) used statistical machine translation methods, requiring parallel simplification data.

2. https://github.com/florelien/Thesis-2024
3. https://wablieft.be/nl

Vandeghinste and Pan (2004) described a sentence compression system for automatically producing subtitles for television programs based on written transcripts. Syntactic compression rules are derived from a syntactically annotated parallel corpus of autocues/subtitles.

Martin et al. (2022) implemented a text simplification model that can be trained on unlabeled data. This involved extracting pairs of sentences that conveyed similar meanings but were phrased differently. By harnessing paraphrased sentences found on the web, a synthetic corpus was created, rich in linguistic variations. This did not only address the issue of limited parallel data but also introduced a method capable of capturing diverse expressions for the same content.

2.3 Dutch Corpora for text simplification

Parallel corpora containing aligned sentences or texts in both the original Dutch text and simplified versions could be used for training and evaluating text simplification models.

Vlantis et al. (2024) created a dataset[4] of 1311 parallel sentence pairs with simplifications, automatically sentence-aligned from texts that were manually simplified for simplification evaluation. The approximately 50 documents from the Communications Department of the City of Amsterdam comprise various types such as reports, citizen letters, and newsletters and cover a diverse range of topics. Vanackere and Vandeghinste (2022)[5] created a comparable corpus of articles from the regular newspaper *De Standaard* and the easy-to-read newspaper *Wablieft*. It consists of 12,687 *Wablieft* articles from 2012-2017 (see Vandeghinste et al. (2019)) and compared them with 206,466 *De Standaard* articles from 2013-2017. The most comparable articles have been retrieved and a comparability score is provided. This dataset does not contain sentence alignment.

The fully synthetic Chatgpt-Dutch-simplification parallel dataset (Van de Velde et al. 2023)[6] was made for sentence simplification using text-to-text transfer transformers. It consists of Dutch source sentences along with their corresponding simplified sentences. Both source and target have been generated with ChatGPT. The dataset consists of 1013 training sentences, 126 validation sentences, and 128 test sentences. The Leesplank_NL dataset[7] consists of 2.87 million paragraphs from Dutch Wikipedia and their corresponding synthetic simplifications, generated by GPT-4 (OpenAI 2023). This naturally leads to the question: 'Why is a new model required if GPT-4 can already simplify text?'. While GPT-4 can do this task, its simplifications are not sufficiently controllable.

There are also a number of datasets based on automatic translation of English simplification datasets.

The Dutch SimpleWiki dataset[8] consists of 167,000 aligned sentence pairs and is an automatic translation of the SimpleWiki dataset.[9] The SimpleWiki dataset is derived from aligning sentences between Simple English Wikipedia and English Wikipedia (Coster and Kauchak 2011). The Dutch WikiLarge dataset (Seidl and Vandeghinste 2024)[10] contains the first 10,000 rows of the WikiLarge dataset. [11] WikiLarge is a large-scale parallel dataset, consisting of complex-simple sentence pairs extracted from English Wikipedia and Simple English Wikipedia (Zhang and Lapata 2017). The Dutch ASSET dataset (Seidl and Vandeghinste 2024)[12] is a machine translation of the English ASSET dataset (Alva-Manchego et al. 2020).[13] This dataset provides a comprehensive benchmark for evaluating sentence simplification models by capturing the varied range of text alterations human editors perform. However, whether the translated version retains the same level of richness in text

4. https://github.com/Amsterdam-AI-Team/dutch-municipal-text-simplification/tree/master/complex-simple-sentences
5. https://github.com/nivack/comparable_corpus_Wablieft_deStandaard
6. https://huggingface.co/datasets/BramVanroy/chatgpt-dutch-simplification
7. https://huggingface.co/datasets/UWV/Leesplank_NL_wikipedia_simplifications
8. https://huggingface.co/datasets/NetherlandsForensicInstitute/simplewiki-translated-nl
9. https://cs.pomona.edu/~dkauchak/simplification/
10. https://github.com/tsei902/simplify_dutch/tree/main/resources/datasets
11. https://github.com/XingxingZhang/dress
12. idem
13. https://github.com/facebookresearch

alterations depends on the quality of the machine translation process. (Seidl and Vandeghinste 2024) have a section on quality testing, which assumes that machine translations of Easy are also Easy. For a selection of sentences, human translations were made, and the MT output was compared with these references, which resulted in a BLEU score of 77. (Seidl and Vandeghinste 2024) concluded that the machine-translated corpora are similar to their human reference translation, and thus this corpora can be used for fine-tuning a language model.

Note that we keep an updated list of sentence simplification datasets at K-Dutch, the CLARIN Knowledge Centre for Dutch.[14]

2.4 Controllable Text Simplification

While text simplification is typically seen as rewriting complex text in simpler language, practical applications require adjusting the simplicity level for specific types of readers. In controllable text simplification, a model rewrites text according to specified attributes, tailoring the output to the intended audience. Techniques for controlling these attributes during generation can include adjustments to the training process or applying constraints during inference.

One method involves inserting special tokens at the start of the input sequence to represent the desired attributes, which guides the generation of the output text. This approach has been used to control aspects like pronoun forms, formality, and style in various language tasks.

Recently, this has been applied to control the amount and type of simplification. Scarton and Specia (2018) trains a sequence-to-sequence model to generate text suitable for specific age or educational levels by adding annotations to the source sentences.

Scarton and Specia (2018) and Kew and Ebling (2022) have control tokens that use the grade level as a proxy for preferred level of reading difficulty. More closely to the actual text, it is possible to manage how complicated the content is by inserting control tokens at the start of the sentence (Martin et al. 2020). These tokens represent different types of simplification actions and help adjust how much the sentence is compressed or rephrased.

Control tokens also determine the complexity of vocabulary and sentence structure. Their use during the training process guide the model to produce outputs tailored to specific requirements. These tokens guide the generation of outputs at inference, ensuring the desired level of compression, paraphrasing, and complexity.

Martin et al. (2020) described several control tokens, each fulfilling a unique role in the simplification process. These tokens regulate the degree of sentence compression, the similarity between the original and simplified sentences, the complexity of the words, and the complexity of the syntactic structure. Sheang et al. (2022) introduced a control token that indicates the difference in number of words between the original and its simplified sentence as a percentage. Agrawal and Carpuat (2023) added a control token that focuses on the character-level Levenshtein similarity, specifically accounting for replace operations between the original and target sentences, and a token that measures the proportion of direct copying from the original to the target sentence.

Seidl and Vandeghinste (2024) investigated the five control tokens of Martin et al. (2020) and Sheang et al. (2022) using a Dutch T5 LLM. These control tokens have proven to enhance the ability to simplify sentences.
One big limitation of most of these methods is that they use fixed control token values for any sentence. However, complex sentences need to be simplified more, while simple sentences do not need to change a lot. A solution to this problem is presented next, where the control token values are predicted based on the sentence's complexity.

14. `https://kdutch.ivdnt.org/wiki/Simplification_Data`

3. Method

We explore new control tokens and their effects. Following Seidl and Vandeghinste (2024), we use a Dutch T5 language model, which we fine-tune on simplified sentences. Where Seidl and Vandeghinste (2024) used fixed values for control tokens without considering the complexity of individual source sentences, we take complexity of source sentences into account to generate appropriate control tokens.(Agrawal and Carpuat 2023).

3.1 Data

We create the *LeesplankSS* dataset, which consists of sentence pairs selected from the LEESPLANK dataset. Sentences from a complex paragraph were aligned with sentences from a simple paragraph using the *all-MiniLM-L6-v2*[15] sentence transformer (Reimers and Gurevych 2019) by computing the cosine similarity of the embeddings generated by the sentence transformer. Sentence pairs where cosine similarity > 0.8 were selected. For each sentence, only the highest match was selected. The dataset consists of roughly 1 million unique sentence pairs, and was randomly split into 80% for training, 10% for validation and 10% for testing.

	Train dataset		**Validation dataset**		**Test dataset**	
	Complex	**Simple**	**Complex**	**Simple**	**Complex**	**Simple**
Rows	847,625	847,625	105,953	105,953	105,954	105,954
Words	13,146,230	11,834,560	1,642,853	1,481,354	1,644,951	1,479,748
Avg. Word Count	15.51	13.96	15.51	13.98	15.53	13.97
Characters (+ spaces)	82,908,326	70,281,815	10,356,023	8,796,525	10,363,504	8,785,257
Avg. Character Count	97.81	82.92	97.74	83.02	97.81	82.92

Table 1: Detailed overview of the statistics of the train dataset, validation dataset, and LEESPLANKSS test dataset. (Avg. = Average)

Table 1 presents statistics about the dataset. On average, complex sentences have a higher average word count compared to their simple counterparts, indicating that complex sentences are generally longer which is in line with (Bulté et al. 2018). The averages of character count further demonstrate that complex sentences also have more characters per sentence compared to simpler ones.

We use the Dutch ASSET dataset (Seidl and Vandeghinste 2024) to compare results of the model across test sets originating from different datasets. Table 2 shows the statistics of this dataset.

	Train dataset		**Validation dataset**		**ASSET Testset**	
	Complex	**Simple**	**Complex**	**Simple**	**Complex**	**Simple**
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	7,292	6,116
Characters (+ 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 2: Detailed overview of the statistics of the LEESPLANKSS train dataset, LEESPLANKSS validation dataset, and ASSET test dataset.

15. `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

3.2 Control Tokens

Ten control tokens (five from literature and five new) are calculated and prepended to all source sentences before training takes place. Table 3 shows an example from the training set of ten control tokens and their respective complex and simple sentences. The first two rows (control tokens and original sentence) of this table are concatenated and passed to the T5 model, which is fine-tuned to generate the last row. Control token values are rounded to the nearest 0.05. This binning approach helps the model to learn the effect of each control token (Li et al. 2022). During validation, a hyperparameter search is used to find the best combination of control token values for the validation set as a whole. Subsequently, these fixed values are prepended to each source sentence in the test set. As stated at the end of section 2, using fixed values from the hyperparameter search is not ideal. This will be addressed later on.

Control Token values	WLR_0.7 CLR_0.7 LR_0.55 WRR_1.1 DTDR_0.75 CCR_0.6 DTRLR_0.75 NSR_1.7 PR_0.3 PPR_1.55
Original	De Schoterlandseweg is een lange weg van minstens 20 kilometer, die van Heerenveen via Mildam - Nieuwehorne - Oudehorne - Jubbega - Hoornsterszwaag tot aan Donkerbroek doorloopt.
Simple	De Schoterlandseweg is een lange weg die van de stad Heerenveen naar het plaatsje Donkerbroek loopt.

Table 3: Training set example of control tokens and their respective complex and simple sentences

We first present the control tokens that were also used by Seidl and Vandeghinste (2024). **Word Length Ratio (WLR)** is the ratio of the number of words in the target sentence to the number of words in the source sentence. **Character Length Ratio (CLR)** is the ratio of the number of characters in the target sentence to the number of characters in the source sentence. Seidl and Vandeghinste (2024) implemented the adapted version of Sheang et al. (2022), which successfully substitutes long words with shorter alternatives or rephrases the text. **Word Rank Ratio (WRR)** represents the ratio between the logarithms of the word ranks in the target and source sentences. A rank is equivalent to the frequency of a word in the language. It gives an indication of how the complexity of the words in the source and target sentences compare. (Seidl and Vandeghinste 2024) implemented the version of (Sheang et al. 2022). Ranks are based on Dutch Common Crawl data.[16] **Levenshtein Similarity Ratio (LR)** measures how similar the source and target sentences are at the character level using the Levenshtein (Levenshtein 1965) similarity metric. It calculates the minimum number of single-character edits (insertions, deletions, or substitutions) needed to change one string into the other. Like Seidl and Vandeghinste (2024), we use the normalised LevenshteinRatio (Martin et al. 2020), (Menta and García-Serrano 2022) and (Sheang et al. 2022). **Dependency Tree Depth Ratio (DTDR)** is the ratio of the maximum depth of the dependency tree of the target sentence to the maximum depth of the dependency tree of the source sentence. It indicates how complex the syntactic structure is in the source compared to the target. Seidl and Vandeghinste (2024) implemented the Dutch model pipeline 'nl_core_news_sm' from Spacy[17].

Besides the control tokens from literature, we experimented with five new control tokens.Nouns play a crucial role in forming simple sentences as they help identify people, places, things, or ideas. Singular nouns, such as *child*, specifically refer to one entity, ensuring clarity and precision in communication. **Noun Singular Ratio (NSR)** measures the ratio of the number of singular nouns in a simple sentence to the number of singular nouns in a complex sentence. By controlling for NSR, we hope to better understand how the presence of singular nouns influences the overall complexity of sentences. Prepositions are used to express relationships between nouns/pronouns and

16. https://www.commoncrawl.org/
17. https://spacy.io/models/nl

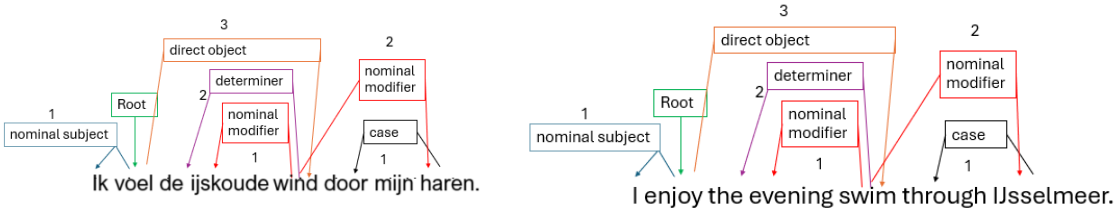

Figure 1: Examples of sentence with dependency tree relation lengths

other elements in a sentence. In the language development of individuals learning a new language, prepositions tend to emerge later (Barak et al. 2020). In simple sentences, sentence length is shorter and prepositions are often omitted.(Barak et al. 2020). **Preposition Ratio (PR)** measures the ratio of the number of prepositions in a simple sentence to the number of prepositions in a complex sentence. By measuring and controlling for PR, we hope to better understand how the presence of prepositions influences the overall complexity of sentences.

For the next control feature, we make a distinction between three degrees of complexity/difficulty in part-of-speech (POS) tags, each provided with a certain amount of penalty points. The **POS Penality Ratio (PPR)** measures the ratio of the number of penalty points of all POS tags occurring in a simple sentence to the number of penalty points in a complex sentence. Words with certain POS tags are considered difficult, which we aim to avoid in the output. Examples of difficult verb POS tags are the past participle, the past singular and plural, and the present plural. Another POS tag that indicates a difficulty is the subordinating conjunction. We give these a $Penalty = 3$. As easy POS tags we consider common nouns and personal pronouns, and prepositions are already controlled. Therefore, these get a $Penalty = 1$. The remaining POS tags are considered of intermediate difficulty and get a $Penalty = 2$.

Average dependency tree relation length is a measure of the average linear distance between two linguistic units that have a syntactic relationship within a sentence (Jiang and Liu 2015) and is related to the syntactic complexity of a sentence. Simplified sentences should have a lower average dependency tree relation length. The interrelations among dependency tree widths, heights, and sentence lengths are explored in detail by Zhang and Liu (2018) who analysed the relation between dependency tree relation length and text complexity. **Dependency Tree Relation Length Ratio (DTRLR)** measures the complexity of sentence structures by comparing the average distances of relationships (dependencies) in a simple sentence to those in a complex sentence.

An example is shown in Figure 1. The length of a dependency tree relation, represented by the number, refers to the distance between two related words in a sentence. The distance is dependent on the exact structural representation. **Copy Control Ratio (CCR)** quantifies the degree to which a sentence is simplified through the direct replication of tokens from the original text, similar to Maddela et al. (2021). To quantify the degree of token alteration between a source sentence and its simplified counterpart, we examine how many tokens in the source sentence appear in the simplified sentence and divide by the total number of tokens in the complex sentence. In this calculation, we count all tokens, not just the unique tokens.

3.3 Fine-tuning the T5 model

HuggingFace offers a wide selection of pre-trained large language models. Among these, one notable example is the T5 model, short for Text-to-Text Transfer Transformer, developed by Google (Raffel et al. 2020). This approach involves feeding the model text as input and training it to generate some target text, which allows for the use of the same model, loss function, and hyperparameters across

diverse tasks (Raffel et al. 2020). According to Seidl and Vandeghinste (2024), the Dutch variant of the T5 model[18] is the best model for Dutch text simplification.

We utilized the beam search algorithm with specific parameters shown in Table 4 to optimize text generation quality.[19] [20]

Parameter	Value
Maximum number of new tokens	64
Minimum number of new tokens	3
`do_sample`	False
Number of beams	5
Number of beam groups	5
Temperature	0
Repetition penalty	1.5
Diversity penalty	0.1
Fine-tuning model	T5
Maximum input tokens	512
Maximum generated tokens	128

Table 4: Beam search parameters

The pre-trained Dutch T5 model was fine-tuned on our dataset. The preparation for fine-tuning includes calculating the control token values to get the sentence dependent values for each sentence pair in the training set. The model with 10 control tokens was trained and validated using cross-entropy loss as the evaluation metric. Training and validation losses were tracked to monitor the performance and generalisation of the model. After 2 epochs, the training loss and the validation loss flattened. Finetuning parameters are presented in Table 5.

Parameter	Value
Batch Size	16
Number of Epochs	2
Learning Rate	1×10^{-4}
Gradient Accumulation Steps	1
Weight Decay	0.01
Optimizer	Adafactor
Warmup Steps	25
Dataset size	800,000

Table 5: Training Parameters

3.4 Hyperparameter search

The hyperparameter search not only involves optimizing the hyperparameters' values but also selecting an effective subset of control tokens. For example, given 10 control tokens, we can evaluate the performance with random values for those 10 control tokens to find the best values. However, we can also take a random subset of 1-9 control tokens and assign random values to those. By exploring hundreds of subsets and their randomly assigned values, we can identify both the optimal hyperparameter values and the most effective subset of control tokens, ensuring that the model achieves the best possible performance. Hyperparameter searches are done with and without selecting subsets

18. `https://huggingface.co/yhavinga/t5-base-dutch`
19. Repetition penalty refers to repetition within the output.
20. The maximum input tokens is too high, but it does no harm (except in terms of memory usage).

to compare the results. The best model that takes subsets into account will be referred to as the "*trained model with the 4 best control tokens*" in the results section, as a configuration with 4 control tokens performed the best. 3 hyperparameter searches were conducted. In the first one, subset selection was enabled. In the last 2, subset selection was disabled. The first two experiments started with 10 control tokens, while the last experiment started with only 7. The reason for choosing 7 control tokens is that the other 3 turned out not to be useful in their current implementation. The results of each hyperparameter search are explained in section 4. The control token values of the 3 hyperparameter searches are shown in Table 6.

Condition	WLR	CLR	LR	WRR	DTDR	CCR	DTRLR	NSR	PR	PPR
4 control tokens	-	0.35	0.45	1.15	-	0.45	-	-	-	-
10 control tokens	0.3	0.35	0.45	1.05	0.4	0.3	0.85	1.75	0.85	0.95
7 control tokens	1.0	0.75	0.4	0.35	0.75	0.1	0.85	-	-	-

Table 6: The best control token values during the hyperparameter search with 10 and 7 control tokens

In Table 14 in the appendix, example sentences are shown in the following order: complex, reference, output of no control token model and the control token model with the best and two alternative control token combinations. The different hyperparameter search combinations have different simplification results. For example, the complex sentence "*De spin leeft op de bodem en maakt geen web.*" [E: The spider lives on the soil and makes no web.] , simplified by the model with the first combination of 7 control tokens results in "*Deze spin maakt geen web.*" [E: This spider makes no web.] compared to the second combination of 7 control tokens "*Deze spin maakt geen web, maar een web.*" [E: This spider makes no web, but a web.], and the third combination "*De spin maakt geen web op de grond.*" [E: The spider makes no web on the ground.]. Some sentences contain the essence only, while others contain additional (sometimes wrong) information. To compare the different control token values, evaluation metrics will be discussed in Section 4.

3.5 Sentence dependent control token prediction with BERTje

Previous methods used fixed values for control tokens (by means of hyperparameter search) for all sentences at test time, without considering the complexity of individual source sentences. Instead of fixed values, we take into account the characteristics and complexity of the difficult sentences by using a transformer architecture. Agrawal and Carpuat (2023) show that this improves the quality and controllability of the simplified outputs compared to using standardised control values. However, the potential of transformers to predict these values remains untested. We investigate how adjusting control values at the sentence level with BERTje, based on each input instance and the desired output complexity, affects the simplification.

We employ BERTje (de Vries et al. 2019) for a regression task in this setup. Initially, the BERT tokenizer and model are loaded, and a regression head is defined to extend the BERT model's capability to predict continuous values. This head consists of a linear layer that takes the [CLS] token's hidden state as input. The model is trained to predict the actual control token values given a complex sentence. The ground truth labels can easily be calculated given the complex sentence and its simple counterpart.

For fine-tuning, the model is trained over 2 epochs with a learning rate of 1e-5. The training process utilizes a batch size of 16 and the AdamW optimizer, with a linear learning rate scheduler applied to manage learning rate adjustments over the training steps. The loss function used for regression is Mean Squared Error (MSELoss).

Hyperparameter tuning was carried out using Weights & Biases,[21] a platform for tracking evaluation results and model weights, offering visualisation options and hyperparameter sweeps to train

21. `https://wandb.ai/site/`

models with different hyperparameter values for analysis.The model is configured with the settings illustrated in Table 7.

Parameter	Value
Decoding	Beam search
Do Sampling	False
Temperature	0
Repetition penalty	1.5
Maximum new tokens	64
Minimum new tokens	3

Table 7: Parameters for hyperparameter search

3.6 Automatic Evaluation

The performance of the fine-tuned simplification model was quantitatively evaluated using several automatic evaluation metrics: SARI, BLEURT, BERTScore, ChrF, FKGL and a single metric that combines all of the above.

To compare with prior studies, we use SARI, a simplification-specific metric, considering the original input along with reference simplifications (Xu et al. 2016). Despite the fact that SARI has a weak correlation with human judgement, other research recommends the use of this metric to compare with prior studies, in this case to compare with the results of Seidl and Vandeghinste (2024).

BLEURT (Sellam et al. 2020) was used instead of Bilingual Evaluation Understudy (BLEU) (Papineni et al. 2001) due to its combination of BLEU and BERTScore (Zhang et al. 2020) features. BLEURT compares the generated text to the reference text using a combination of pre-trained language models and synthetic data.

In addition to BLEURT, we calculated the BERTScore (Zhang et al. 2020) to further evaluate the outputs, as it focuses specifically on measuring the semantic similarity between the generated and reference texts.

In comparison to BLUE, the translation outputs of ChrF (CHaRacter-level F-score) have higher performance (Popović 2015). ChrF is a machine translation evaluation metric that assesses the similarity between a machine translation output and a reference translation by comparing character n-grams rather than word n-grams.

FKGL is used to grade the complexity of the sentence, Flesch-Kincaid Grade Level (FKGL) measurements were applied. FKGL computes the complexity level of a standalone piece of text or sentence, by taking into account the number of vowel-sound units in a word and the sentence length (Ondov et al. 2022, Kincaid et al. 1975). While higher values indicate better translations for the previously mentioned metrics, lower values of FKGL indicate simpler translations while higher values indicate more difficult-to-read translations.

Finally, a metric that combines all of the above is used to get an overall score of the simplification. It combines all of the above, with equal weights per metric, into a single value from 0 to 1. For this metric, higher scores indicate better overall simplifications.

4. Results

4.1 Experiments

Experiment 1 evaluates the results of the simplification process using 10 control tokens on the LEESPLANKSS test dataset. The control token values of the Trained with BERT model are predicted as described in Section 3.5, while the fixed control token values of the other models are results

from multiple hyperparameter searches, as described in Section 3.4. At last, the control tokens with their values of the different models are illustrated in Table 8.

Models and values	WLR	CLR	LR	WRR	DTDR	CCR	DTRLR	NSR	PR	PRR
Trained with 10 CT	0.3	0.35	0.45	1.05	0.4	0.3	0.85	1.75	0.85	0.95
Trained with 5 old CT	0.3	0.35	0.45	1.05	0.4	-	-	-	-	-
Trained with 5 new CT	-	-	-	-	-	0.3	0.85	1.75	0.85	0.95

Table 8: The different models with control tokens and their values

The results shown in Table 9 indicate a large difference in performance between the untrained and trained models. The untrained model has the lowest scores across all evaluation metrics. FKGL was not measured for this model, because the output sentences were mostly unfinished, badly structured sentences. This reduces the relevance of the FKGL metric. The trained model without control tokens achieved the best scores across 3 of the 5 metrics; although, using the BERT model with 10 control tokens demonstrates similar performance, closely matching with the best results. Compared to the trained model with 10 control tokens (values in Table 6), the BERT model with 10 control tokens achieves better scores across all evaluation metrics.

Using only the 5 old control tokens (WLR, CLR, LR, WRR, DTDR) showed slightly higher performance compared to using 10 control tokens. Among all the trained models, our trained model with only the 5 new control tokens performs the worst.

Models	BLEURT	ChrF	BERTScore	SARI	FKGL	Combined[22]
Untrained	0.27	22.71	0.62	36.80	-	0.50
Trained without CT	0.76	59.58	0.89	61.63	5.12	0.72
Trained with 10 CT	0.72	58.76	0.86	56.74	6.68	0.67
Trained with BERT (10 CT)	**0.75**	**60.03**	**0.88**	**62.21**	**5.62**	**0.71**
Trained with 5 old CT	0.74	59.04	0.87	57.76	5.92	0.69
Trained with 5 new CT	0.69	58.41	0.85	50.04	7.94	0.64

Table 9: Results of Experiment 1 with 10 CT on LEESPLANKSS dataset. (CT = Control Tokens)

Experiment 2 uses only 7 control tokens (5 existing and 2 new), with the Dependency Tree Relation Length Ratio (DTRLR) and Copy Control Ratio (CCR) introduced as new control tokens. The Noun Singular Ratio (NSR), the Preposition Ratio (PR), and the POS Penalty Ratio (PPR) were removed. The control tokens with their values of the different models are illustrated in Table 10.

Models and values	WLR	CLR	LR	WRR	DTDR	CCR	DTRLR	NSR	PR	PRR
Trained with 7 CT	1.0	0.75	0.4	0.35	0.75	0.1	0.85	-	-	-
Trained with 2 new CT	-	-	-	-	-	0.1	0.85	-	-	-
Trained with 5 old CT	0.3	0.35	0.45	1.05	0.4	-	-	-	-	-
Trained with 4 best CT	-	0.35	0.45	1.15	-	0.45	-	-	-	-

Table 10: The different models with control tokens and their values

In Experiment 2, as in Experiment 1, a difference in performance is observed between the untrained and trained models. The untrained model has the lowest scores across all evaluation metrics (Table 11). Again, the trained model without control tokens achieves the highest scores across almost all metrics, demonstrating a similar performance to the BERT model with 7 control tokens.

As shown in Table 11, the BERT model with 7 control tokens outperforms the trained model with fixed control token values (cf. Table 6). Remarkably, this BERT model with 7 control tokens

22. The combined score is the average of the normalized and rescaled scores of the 5 metrics.

similarly resulted in almost exactly the same results as the BERT model with 10 control tokens in Table 9. Among all the trained models, our trained model with the 7 control tokens performs the worst on all metrics but SARI.

In comparison to the trained model using 7 control token values, the trained model using only 2 new control tokens showed strong performance in all metrics but SARI, where the score dropped to 49.62, and the combined metric (0.66).

To compare only the most important models, only the best result from each hyperparameter search is shown in Table 11. This is the best result from a hyperparameter search with all 7 control tokens, the 5 old control tokens, the 2 new control tokens and all 7 control tokens with subset selection. These 4 control tokens, resulting from the hyperparameter search with subset selection, achieved higher scores for BLEURT, ChrF, and SARI compared to the trained model with 5 old control tokens.

Models	BLEURT	ChrF	BERTScore	SARI	FKGL	Combined
Untrained	0.27	22.71	0.62	36.80	0.0	0.50
Trained without CT	0.76	59.58	0.89	61.63	5.12	0.72
Trained with 7 CT	0.67	45.16	0.84	53.35	4.05	0.66
Trained with BERT (7CT)	**0.75**	**60.77**	**0.88**	**62.34**	**5.61**	**0.71**
Trained with 2 new CT	0.74	58.94	0.86	49.62	6.72	0.66
Trained with 5 old CT	0.72	54.92	0.87	56.82	4.92	0.69
Trained with 4 best CT	0.73	57.42	0.87	58.48	5.76	0.69

Table 11: Results of Experiment 2 with 7 CT on LEESPLANKSS dataset.

Experiment 3 presented in Table 13 highlights the performance of different models evaluated on the ASSET test dataset[23] of Seidl and Vandeghinste (2024). In this experiment, BLEURT and the combined metric were omitted due to out of memory issues. The control tokens with their values of the different models are illustrated in Table 12.

Models and values	WLR	CLR	LR	WRR	DTDR	CCR	DTRLR	NSR	PR	PRR
Trained with 10 CT	0.3	0.35	0.45	1.05	0.4	0.3	0.85	1.75	0.85	0.95
Trained with 7 CT	0.35	1.95	0.7	0.45	0.65	0.4	1.4	-	-	-
Trained with 2 new CT	-	-	-	-	-	0.4	1.4	-	-	-
Trained with 5 old CT	0.6	0.7	0.6	0.55	0.75	-	-	-	-	-
Trained with 4 best CT	1.5	1.05	-	-	1.2	-	0.5	-	-	-

Table 12: The different models with control tokens and their values

The untrained model achieves the lowest scores across all metrics. Again, the trained models have higher results compared to the untrained model across all metrics. There is no clear winner in this experiment, as multiple models demonstrate similar performance. Models with higher ChrF scores have worse FKGL scores and vice versa. The best scores of the evaluation metrics were all achieved by a different model. Depending on which metric is deemed to be more important, one might have a preference for one specific model. The model trained with 4 best CT has a slight edge over the other models. However, the model with 7 CT, as well as the model with 2 new CT, demonstrates similar performance. The model with 10 CT demonstrates the worst performance among all the models, followed shortly by the model without CT and the BERT model.

Comparing the second experiment (Table 11) with the last experiment (Table 13), the models evaluated on the LEESPLANKSS test dataset generally have better scores than those models evaluated on the ASSET test dataset across multiple metrics. For instance, the trained model with 7 control tokens on LEESPLANKSS highest ChrF score in the second experiment (60.77) is achieved

23. https://github.com/tsei902/simplify_dutch/tree/main/resources/datasets/asset

by the trained model with BERT model (7 CT), whereas in the last experiment, a ChrF score of 59.49 is achieved in the same model. This comes as no surprise, given the increased complexity of sentences in the ASSET dataset.

Models	ChrF	BERTScore	SARI	FKGL
Untrained	46.77	0.65	36.28	-
Trained without CT	58.79	0.89	42.80	**6.28**
Trained with 10 CT	58.31	0.87	42.54	7.21
Trained with 7 CT	66.30	0.90	**43.79**	8.14
Trained with BERT model (7 CT)	59.49	0.88	43.11	6.92
Trained with 2 new CT	**72.00**	0.90	42.73	9.58
Trained with 5 old CT	69.07	0.90	42.73	9.58
Trained with 4 best CT	70.97	**0.92**	43.52	8.25

Table 13: Results with translated ASSET testset created by Seidl and Vandeghinste (2024)

To determine the impact of increasing the amount of training sentence pairs on the model performance, a T5 model with the 4 best CT was trained multiple times on a subset of the LEESPLANKSS until no further improvements on a fixed LEESPLANKSS validation set and evaluated on ASSET. By starting with a very low number of training samples and gradually adding more samples, a threshold can be identified at which additional data no longer greatly improves performance. ASSET was selected as a test set because these results indicate at which threshold the model no longer generalizes to unseen data of higher complexity of at least one different source. Figures 2 to 5 demonstrates that increasing the LEESPLANKSS training dataset size beyond 400 does not result in improvements for most evaluation metrics. An exception is observed with the SARI metric (Figure 2), where the data indicates that the threshold for optimal performance is 100,000.

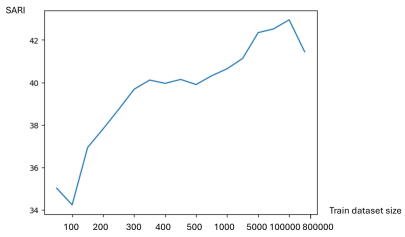

Figure 2: SARI scores

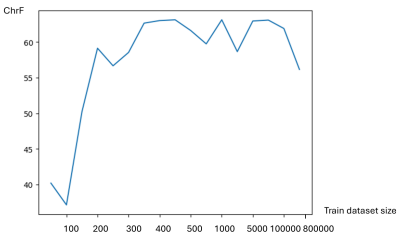

Figure 3: Chrf scores

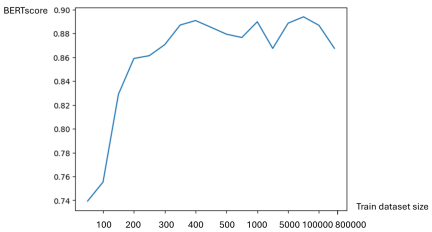

Figure 4: Bert scores

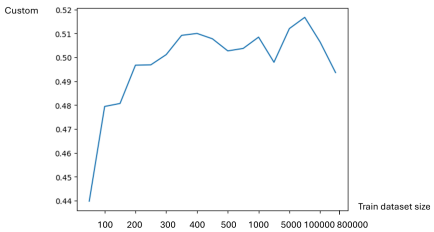

Figure 5: Custom metric

5. Discussion

This study compares the performance of models trained with different configurations of control tokens. The results of experiment 1 indicate that control tokens generally do not improve the simplifications on the LEESPLANKSS dataset, unless they have sentence dependent values. The model trained without control tokens performed exceptionally well but incorporating the BERT method to predict the 10 control tokens values yielded the best SARI and ChrF scores, highlighting the effectiveness of sentence-dependent control token values in language simplification tasks. Overall, the choice of training method and control tokens plays a crucial role in the performance of the models on the LEESPLANKSS dataset.

Experiment 2 with 7 control tokens instead of 10 indicates similar results, where the model without CT and the BERT model demonstrate the highest performances. Going from 10 to 7 control tokens, the predictions of the fixed control token model have lower quality according to all metrics except FKGL, indicating that it is making much simpler sentences than before. Interestingly, both BERT models (7 versus 10 control tokens) have similar results, suggesting that the three new tokens (PPR, NSR, and PR) do not improve the results. However, another reason for this lack of improvement could be that these three control token values are more difficult to predict by the BERT model.

Overall, the results demonstrate that training with different configurations of control tokens can influence model performance. The models trained with BERT generally outperformed those models with fixed CT values, indicating the importance of leveraging both pre-trained models and control tokens for improving text generation quality.

Remarkably, trained models without control tokens consistently achieved high scores across most metrics in both experiments on LEESPLANKSS dataset, suggesting that the dataset's inherent features are well-captured without additional token control. This is different from the results obtained by using ASSET as the test set, where using CT generally has a positive impact on the evaluation metrics (Table 13).

First, overall the SARI scores are lower and the FKGL scores are higher. The fine-tuned models are not capable of generalizing well enough to overcome this difference in sentence difficulty. However, compared to previous research (Seidl and Vandeghinste 2024), the SARI scores are generally higher across all fine-tuned models, indicating that the LEESPLANKSS training dataset is of better quality than the dataset that was used in their research. There is a significant difference in most evaluation metrics between the 2 test sets. A probable reason for this is the difference in distribution of control token values. While CLR values of the ASSET test set are mostly below 1, CLR values of the LEESPLANKSS test set show a normal distribution centered around 1. This difference causes the use of static control tokens while evaluating on ASSET to be less problematic for some models. This could also be the reason why the BERT model does not improve performance in this experiment, as static control token values perform better for this dataset.

The findings from this study underscore several critical insights into the effectiveness of using (new) control tokens for Dutch sentence simplification using the T5 large language model. Since simplified sentences can sometimes be longer or shorter than the original sentences, it is impractical to use fixed values for control tokens. For example, if CLR has a fixed value of 0.35, the model is inclined to produce simple sentences that have a length of only 30% of the complex sentences. As the CLR distribution of our dataset resembles a normal distribution centred around 1, this will not result in correct simplifications most of the time . This reinforces the necessity for a more dynamic approach that adapts to the characteristics of individual sentences, which our novel method aims to address.

Furthermore, we observed that using 800,000 sentence pairs of training data compared to 400 shows only a minor difference. This finding aligns with previous research indicating that neural networks can learn effectively from relatively small datasets and generalize to new examples given sufficient tuning (Zhang et al. 2018). This implies that for tasks like sentence simplification, which

involve transforming complex sentences into simpler ones while retaining essential meaning, the marginal gains from large datasets are limited.

Limitations

A limitation of the study is the reliance on automatic evaluation metrics. Although these metrics, such as BLEURT and SARI, provide valuable quantitative insights, they may not fully capture the qualitative aspects of text simplification (Xu et al. 2016). Human evaluation remains the gold standard for assessing text simplification quality, as it can capture subtle nuances that automated metrics might miss. Nonetheless, the combination of automatic evaluation metrics offers a useful and reproducible tool to assess and compare models during development.

Secondly, our synthetic LEESPLANKSS dataset consists of selected parallel sentence pairs of the LEESPLANK dataset via sentence transformers. This method sometimes resulted in poor quality sentence pairs. The original dataset is on paragraph-level, while our dataset is on sentence-level. For instance:

- The complex sentence "Het vliegtuigbedrijf Fokker is naar hem genoemd." is simplified to "Hij was zo goed in het bouwen van vliegtuigen dat er zelfs een bedrijf naar hem is vernoemd, het Fokker vliegtuigbedrijf.
- The complex sentence "Er zijn veel wandel- en fietspaden in dit gebied." is simplified to "Daar kun je lekker wandelen of fietsen, want er zijn veel paden.".

The simplification of these sentences, originating from more complex paragraphs, results in a different interpretation. This approach is inadequate as it introduces a subjective interpretation to what should remain an objective observation.

Future work

This study opens opportunities for future research. One area of exploration could be the integration of human evaluations alongside automatic metrics to capture qualitative aspects of text simplification more effectively.

Another aspect to discuss and to investigate is experimental deletion of the Noun Singular Ratio (NSR), the Preposition Ratio (PR), and the POS Penalty Ratio (PPR). The NSR and PR are deleted in the second experiment, because complex sentences inherently contain more nouns and prepositions due to their length, making the effect of these tokens similar with the Word Length Ratio (WLR) token. To counter this, sentence length should be taken into account. Additionally, the POS Penalty Ratio (PPR) was excluded because it did not produce the hoped result and needs too much added experimentation and tuning to keep it in the set of control tokens.

Moreover, the BERT model used in this research is scalable and can be adapted to other languages and contexts. Furthermore, this method can be improved by modifying the training process. Currently during training, the parameters of the BERT model are updated based on the Mean Square Error loss applied to the difference between the predicted and reference control tokens. However, the ground truth control token values do not necessarily produce the best results, because the BERT and T5 models are trained separately. The following training technique has potential to improve these control token value predictions:

1. Predict control tokens for a complex sentence using BERT
2. Provide the predicted values with the complex sentence as input to the T5 model
3. Compare the predicted simple sentence to the reference and calculate cross-entropy loss
4. Use this loss to update the weights of the BERT model

By training in this way, the BERT model will predict control token values that actually lead to good simplifications because the predictions are based on the reference simple sentence.

6. Conclusion

Automated text simplification provides easier text to language learners, non-native speakers, individuals with neurocognitive disorders, and people with intellectual disabilities. This paper examined 5 new control tokens with a Dutch T5 large language model, and predicts sentence-dependent control token values with a Dutch BERT model.

Previous methods used fixed values for control tokens without considering the complexity of individual difficult source sentences. Instead of fixed values, the characteristics and complexity of the difficult sentences were taken into account. The introduction of the 5 new control tokens has not proven to enhance the model's ability to simplify sentences. The results of the models evaluated on the ASSET test dataset differ from the evaluation scores obtained by evaluating on the LEESPLANKSS test set. Both datasets have a different distribution of control token values. While CLR values of the ASSET test set are mostly below 1, CLR values of the LEESPLANKSS test set show a normal distribution centered slightly below 1 (meaning that the average simple sentence contains roughly the same words than its complex counterpart).

This study opens avenues for future research. One area of exploration could be the integration of human evaluations alongside automatic metrics to capture qualitative aspects of text simplification more effectively. Another area could be optimizing the BERT model. The current BERT approach is scalable and can be adapted to other languages and contexts. In addition, this method can be improved by modifying the training process.

Appendix A. Example Simplifications

Model	Sentence
Complex	Dutch: De spin leeft op de bodem en maakt geen web. English: The spider lives on the soil and makes no web.
Reference	Dutch: Deze spin maakt geen web en leeft op de grond. English: This spider makes no web and lives on the ground.
NO CT	Dutch: Deze spin maakt geen web maar leeft op de grond. English: This spider makes no web but lives on the ground.
7 CTs (1)	Dutch: Deze spin maakt geen web. English: This spider makes no web.
7 CTs (2)	Dutch: Deze spin maakt geen web, maar een web. English: This spider makes no web, but a web.
7 CTs (3)	Dutch: Deze spin maakt geen web op de grond. English: The spider makes no web on the ground.
Complex	Dutch: In 1960 leidde hij de campagne die Johnson de Democratische nominatie voor het presidentschap zou moeten opleveren. English: In 1960, he led the campaign that was to give Johnson the Democratic nomination for president.
Reference	Dutch: In 1960 hielp hij met een campagne om Johnson de belangrijkste kandidaat van de Democratische partij voor het presidentschap te maken. English: In 1960, he helped lead a campaign to make Johnson the Democratic party's leading candidate for the presidency.
NO CT	Dutch: In 1960 was hij de leider van een campagne om Johnson te selecteren voor de Democratische partij. English: In 1960 he was the leader of a campaign to select Johnson for the Democratic party.
7 CTs (1)	Dutch: In 1960 was hij de leider van een campagne om Johnson te vertegenwoordigen. English: In 1960, he was the leader of a campaign to represent Johnson.
7 CTs (2)	Dutch: In 1960 leidde hij een campagne om Johnson te selecteren voor de Democratische partij. English: In 1960, he led a campaign to select Johnson for the Democratic party.
7 CTs (3)	Dutch: In 1960 leidde hij een campagne voor Johnson. English: In 1960, he led a campaign to represent Johnson.
Complex	Dutch: Aan het einde van de 19e eeuw werden er in en rond Noordwolde zo'n 200,000 stoelen per jaar gemaakt, welke voor het grootste gedeelte per trein werden geëxporteerd via voormalig station Peperga. English: At the end of the 19th century some 200,000 chairs a year were made in and around Noordwolde, most of them being exported by train via the former Peperga station.
Reference	Dutch: Aan het eind van de 19e eeuw maakte Noordwolde wel 200,000 stoelen per jaar. English: By the end of the 19th century, Noordwolde was making as many as 200,000 chairs a year.
NO CT	Dutch: Aan het eind van de 19e eeuw maakten ze wel 200,000 stoelen per jaar in Noordwolde en omgeving. English: By the end of the 19th century, they were making as many as 200,000 chairs a year in Noordwolde and surrounding areas.
7 CTs (1)	Dutch: Aan het eind van de 1800s maakten ze ongeveer 200.000 stoelen per jaar, vooral in Noordwolde. English: By the late 1800s they were making about 200,000 chairs a year, mostly in Noordwolde.
7 CTs (2)	Dutch: Aan het eind van de 1800s maakten ze in Noordwolde ongeveer 200,000 stoelen per jaar, en de meeste werden per trein naar een ander station gebracht. English: By the end of the 1800s they were making about 200,000 chairs a year in Noordwolde, and most of them were taken by train to another station.
7 CTs (3)	Dutch: In de 19e eeuw maakten ze ongeveer 200,000 stoelen per jaar in Noordwolde en de omgeving. English: In the 19th century, they made about 200,000 chairs a year in Noordwolde and the surrounding area.
(1)	WLR 1.0, CLR 0.75, LR 0.4, WRR 0.35, DTDR 0.75, CCR 0.1, DTRL 0.85
(2)	WLR 1.6, CLR 0.3, LR 0.55, WRR 1.15, DTDR 0.45, CCR 0.4, DTRL 1.95
(3)	WLR 0.45, CLR 0.35, LR 0.25, WRR 1.2, DTDR 0.45, CCR 0.85, DTRL 0.3

Table 14: Examples of output sentences with 7 control token values

References

Agrawal, Sweta and Marine Carpuat (2023), Controlling pre-trained language models for grade-specific text simplification, *in* Bouamor, Houda, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, pp. 12807–12819. https://aclanthology.org/2023.emnlp-main.790.

Alva-Manchego, Fernando, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia (2020), ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations, *in* Jurafsky, Dan, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *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.

Barak, Libby, Scott Cheng-Hsin Yang, Chirag Rank, and Patrick Shafto (2020), Modeling second language preposition learning, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Brysbaert, Marc, Michaël Stevens, Simon De Deyne, Wouter Voorspoels, and Gert Storms (2014), Norms of age of acquisition and concreteness for 30,000 dutch words, *Acta Psychologica* **150**, pp. 80–84. https://www.sciencedirect.com/science/article/pii/S0001691814000985.

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.

Coster, William and David Kauchak (2011), Simple english wikipedia: a new text simplification task, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, Association for Computational Linguistics, USA, p. 665–669.

de Vries, Wietse, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim (2019), Bertje: A dutch bert model. https://arxiv.org/abs/1912.09582.

Jiang, Jingyang and Haitao Liu (2015), The effects of sentence length on dependency distance, dependency direction and the implications–based on a parallel english–chinese dependency treebank, *Language Sciences* **50**, pp. 93–104. https://www.sciencedirect.com/science/article/pii/S0388000115000418.

Kew, Tannon and Sarah Ebling (2022), Target-level sentence simplification as controlled paraphrasing, *in* Štajner, Sanja, Horacio Saggion, Daniel Ferrés, Matthew Shardlow, Kim Cheng Sheang, Kai North, Marcos Zampieri, and Wei Xu, editors, *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Virtual), pp. 28–42. https://aclanthology.org/2022.tsar-1.4.

Kincaid, J. Peter, Robert P. Jr. Fishburne, Richard L. Rogers, and Brad S. Chissom (1975), Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. https://stars.library.ucf.edu/istlibrary/56.

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.

Li, Zihao, Matthew Shardlow, and Saeed Hassan (2022), An investigation into the effect of control tokens on text simplification, *in* Štajner, Sanja, Horacio Saggion, Daniel Ferrés, Matthew Shardlow, Kim Cheng Sheang, Kai North, Marcos Zampieri, and Wei Xu, editors, *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Virtual), pp. 154–165. https://aclanthology.org/2022.tsar-1.14.

Maddela, Mounica, Fernando Alva-Manchego, and Wei Xu (2021), Controllable text simplification with explicit paraphrasing, *in* Toutanova, Kristina, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *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, *in* Calzolari, Nicoletta, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *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, *in* Calzolari, Nicoletta, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *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.

Menta, Antonio Garuz and Ana M. García-Serrano (2022), Controllable sentence simplification using transfer learning, *Conference and Labs of the Evaluation Forum*. https://api.semanticscholar.org/CorpusID:251471835.

Ondov, Brian, Kush Attal, and Dina Demner-Fushman (2022), A survey of automated methods for biomedical text simplification, *Journal of the American Medical Informatics Association* **29** (11), pp. 1976–1988. https://doi.org/10.1093/jamia/ocac149.

OpenAI (2023), Gpt-4 technical report. https://openai.com/research/gpt-4.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2001), Bleu: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Association for Computational Linguistics.

Popović, Maja (2015), chrF: character n-gram F-score for automatic MT evaluation, *in* Bojar, Ondřej, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Association for Computational Linguistics, Lisbon, Portugal, pp. 392–395. https://aclanthology.org/W15-3049.

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, *Journal of Machine Learning Research* **21** (140), pp. 1–67. http://jmlr.org/papers/v21/20-074.html.

Reimers, Nils and Iryna Gurevych (2019), Sentence-bert: Sentence embeddings using siamese bert-networks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics. http://arxiv.org/abs/1908.10084.

Saggion, Horacio (2022), 1114Text Simplification, *The Oxford Handbook of Computational Linguistics*, Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199573691.013.52.

Scarton, Carolina and Lucia Specia (2018), Learning simplifications for specific target audiences, *in* Gurevych, Iryna and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Melbourne, Australia, pp. 712–718. https://aclanthology.org/P18-2113.

Seidl, Theresa and Vincent Vandeghinste (2024), Controllable sentence simplification in dutch, *Computational Linguistics in the Netherlands Journal* **13**, pp. 31–61. https://clinjournal.org/clinj/article/view/171.

Sellam, Thibault, Dipanjan Das, and Ankur Parikh (2020), BLEURT: Learning robust metrics for text generation, *in* Jurafsky, Dan, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 7881–7892. https://aclanthology.org/2020.acl-main.704.

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 & Knowledge Engineering* **117**, pp. 264–289. https://www.sciencedirect.com/science/article/pii/S0169023X17304974.

Sheang, Kim Cheng, Daniel Ferrés, and Horacio Saggion (2022), Controllable lexical simplification for English, *in* Štajner, Sanja, Horacio Saggion, Daniel Ferrés, Matthew Shardlow, Kim Cheng Sheang, Kai North, Marcos Zampieri, and Wei Xu, editors, *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Virtual), pp. 199–206. https://aclanthology.org/2022.tsar-1.19.

Van de Velde, Charlotte, Vincent Vandeghinste, Bram Vanroy, and KU Leuven. Faculteit Ingenieurswetenschappen. Opleiding Master of Artificial Intelligence (Leuven) degree granting institution (2023), Automatic sentence-level simplification for dutch.

Vanackere, Nick and Vincent Vandeghinste (2022), *Building a comparable corpus between easy-to-read Dutch Wablieft and De Standaard*, Master's thesis, KU Leuven. Faculteit Ingenieurswetenschappen.

Vandeghinste, Vincent, Adéline Muller, Thomas François, and Orphée Declercq (2021), Easy Language in Belgium, *in* Lindholm, Camilla and Ulla Vanhatalo, editors, *Handbook of Easy Languages in Europe*, Frank & Timme, Berlin, pp. 53 – 90.

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.

Vlantis, Daniel, Iva Gornishka, and Shuai Wang (2024), Benchmarking the simplification of Dutch municipal text, *in* Calzolari, Nicoletta, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, ELRA and ICCL, Torino, Italia, pp. 2217–2226. https://aclanthology.org/2024.lrec-main.199.

Wubben, Sander, Antal van den Bosch, and Emiel Krahmer (2012), Sentence simplification by monolingual machine translation, *in* Li, Haizhou, Chin-Yew Lin, Miles Osborne, Gary Geunbae Lee, and Jong C. Park, editors, *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.

Zhang, Hongxin and Haitao Liu (2018), Interrelations among dependency tree widths, heights and sentence lengths, *in* Jiang, Jingyang and Haitao Liu, editors, *Quantitative Analysis of Dependency Structures*, De Gruyter Mouton, pp. 31–52.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi (2020), Bertscore: Evaluating text generation with bert. https://arxiv.org/abs/1904.09675.

Zhang, Xingxing and Mirella Lapata (2017), Sentence simplification with deep reinforcement learning, *in* Palmer, Martha, Rebecca Hwa, and Sebastian Riedel, editors, *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.

Zhang, Zhirui, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen (2018), Style transfer as unsupervised machine translation, *arXiv.org*, Cornell University Library, arXiv.org, Ithaca.