*Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4689–4698
Marseille, 11–16 May 2020

# Controllable Sentence Simplification

**Louis Martin**[1,2,3] **Éric Villemonte de la Clergerie**[2] **Benoît Sagot**[2] **Antoine Bordes**[1]
Facebook AI Research[1], Inria[2], Sorbonne Université[3].
6 Rue Ménars, 75002 Paris[1]; 2 rue Simone Iff, 75012 Paris[2].
louismartin@fb.com, {benoit.sagot, eric.de_la_clergerie}@inria.fr, antoine.bordes@fb.com

**Abstract**

Text simplification aims at making a text easier to read and understand by simplifying grammar and structure while keeping the underlying information identical. It is often considered an all-purpose generic task where the same simplification is suitable for all; however multiple audiences can benefit from simplified text in different ways. We adapt a discrete parametrization mechanism that provides explicit control on simplification systems based on Sequence-to-Sequence models. As a result, users can condition the simplifications returned by a model on attributes such as length, amount of paraphrasing, lexical complexity and syntactic complexity. We also show that carefully chosen values of these attributes allow out-of-the-box Sequence-to-Sequence models to outperform their standard counterparts on simplification benchmarks. Our model, which we call ACCESS (as shorthand for AudienCe-CEntric Sentence Simplification), establishes the state of the art at 41.87 SARI on the WikiLarge test set, a +1.42 improvement over the best previously reported score.

**Keywords:** Text Simplification, Sequence-to-Sequence models, ACCESS

## 1. Introduction

In Natural Language Processing, the Text Simplification task aims at making a text easier to read and understand. Text simplification can be beneficial for people with cognitive disabilities such as aphasia (Carroll et al., 1998), dyslexia (Rello et al., 2013) and autism (Evans et al., 2014) but also for second language learners (Xia et al., 2016) and people with low literacy (Watanabe et al., 2009). The type of simplification needed for each of these audiences is different. Some aphasic patients struggle to read sentences with a high cognitive load such as long sentences with intricate syntactic structures, whereas second language learners might not understand texts with rare or specific vocabulary. Yet, research in text simplification has been mostly focused on developing models that generate a single generic simplification for a given source text with no possibility to adapt outputs for the needs of various target populations.

In this paper, we propose a controllable simplification model that provides explicit ways for users to manipulate and update simplified outputs as they see fit. This work only considers the task of *Sentence Simplification* (SS) where the input of the model is a single source sentence and the output can be composed of one sentence or split into multiple. Our work builds upon previous work on controllable text generation (Kikuchi et al., 2016; Fan et al., 2017; Scarton and Specia, 2018; Nishihara et al., 2019) where a Sequence-to-Sequence (Seq2Seq) model is modified to control attributes of the output text. We tailor this mechanism to the task of SS by considering relevant attributes of the output sentence such as the output length, the amount of paraphrasing, lexical complexity, and syntactic complexity. To this end, we condition the model at train time, by feeding parameter tokens representing these attributes along with the source sentence as additional inputs.

Our contributions are the following: (1) We adapt a parametrization mechanism to the specific task of Sentence Simplification by conditioning on relevant attributes; (2) We show through a detailed analysis that our model can indeed control the considered attributes, making the simplifications potentially able to fit the needs of various end audiences; (3) With careful calibration, our controllable parametrization improves the performance of out-of-the-box Seq2Seq models leading to a new state-of-the-art score of 41.87 SARI (Xu et al., 2016) on the WikiLarge benchmark (Zhang and Lapata, 2017), a +1.42 gain over previous scores, without requiring any external resource or modified training objective.

## 2. Related Work

### 2.1. Sentence Simplification

Text simplification has gained increasing interest through the years and has benefited from advances in Natural Language Processing and notably Machine Translation.

In recent years, SS was largely treated as a monolingual variant of machine translation (MT), where simplification operations are learned from complex-simple sentence pairs automatically extracted from English Wikipedia and Simple English Wikipedia (Zhu et al., 2010; Wubben et al., 2012).

Phrase-based and Syntax-based MT was successfully used for SS (Zhu et al., 2010) and further tailored to the task using deletion models (Coster and Kauchak, 2011) and candidate reranking (Wubben et al., 2012). The candidate reranking method by Wubben et al. (2012) favors simplifications that are most dissimilar to the source using Levenshtein distance. The authors argue that dissimilarity is a key factor of simplification.

Lately, SS has mostly been tackled using Seq2Seq MT models (Sutskever et al., 2014). Seq2Seq models were either used as-is (Nisioi et al., 2017) or combined with reinforcement learning thanks to a specific simplification reward (Zhang and Lapata, 2017), augmented with an external simplification database as a dynamic memory (Zhao et al., 2018) or trained with multi-tasking on entailment and paraphrase generation (Guo et al., 2018).

This work builds upon Seq2Seq as well. We prepend additional inputs to the source sentences at train time, in the form of plain text special tokens. Our approach does not require any external data or modified training objective.

## 2.2. Controllable Text Generation

Conditional training with Seq2Seq models was applied to multiple natural language processing tasks such as summarization (Kikuchi et al., 2016; Fan et al., 2017), dialog (See et al., 2019), sentence compression (Fevry and Phang, 2018; Mallinson et al., 2018) or poetry generation (Ghazvininejad et al., 2017).
Most approaches for controllable text generation are either decoding-based or learning-based.

**Decoding-based methods** Decoding-based methods use a standard Seq2Seq training setup but modify the system during decoding to control a given attribute. For instance, the length of summaries was controlled by preventing the decoder from generating the End-Of-Sentence token before reaching the desired length or by only selecting hypotheses of a given length during the beam search (Kikuchi et al., 2016). Weighted decoding (i.e. assigning weights to specific words during decoding) was also used with dialog models (See et al., 2019) or poetry generation models (Ghazvininejad et al., 2017) to control the number of repetitions, alliterations, sentiment or style.

**Learning-based methods** On the other hand, learning-based methods condition the Seq2Seq model on the considered attribute at train time, and can then be used to control the output at inference time. Kikuchi et al. (2016) explored learning-based methods to control the length of summaries, e.g. by feeding a target length vector to the neural network. They concluded that learning-based methods worked better than decoding-based methods and allowed finer control on the length without degrading performances. Length control was likewise used in sentence compression by feeding the network a length countdown scalar (Fevry and Phang, 2018) or a length vector (Mallinson et al., 2018). (Ficler and Goldberg, 2017) concatenate a context vector to the hidden state of each time step of their recurrent neural network decoder. This context vector represents the controlled stylistic attributes of the text, where an embedding is learnt for each attribute value. (Hu et al., 2017) achieved controlled text generation by disentangling the latent space representations of a variational auto-encoder between the text representation and its controlled attributes such as sentiment and tense. They impose the latent space structure during training by using additional discriminators.
Our work uses a simpler approach: we condition the generation process by concatenating plain text special tokens to the source text. This method only modifies the source data and not the training procedure. Such mechanism was used to control politeness in MT (Sennrich et al., 2016), to control summaries in terms of length, of news source style, or to make the summary more focused on a given named entity (Fan et al., 2017). Scarton and Specia (2018) and Nishihara et al. (2019) similarly showed that adding special tokens at the beginning of sentences can improve the performance of Seq2Seq models for SS. Plain text special tokens were used to encode attributes such as the target school grade-level (i.e. understanding level) and the type of simplification operation applied between the source and the ground truth simplification (identical, elaboration, one-to-many, many-to-one). Our work goes further by using a more diverse set of parameter tokens that represent specific grammatical attributes of the text simplification process. Moreover, we investigate the influence of those parameter tokens on the generated simplification in a detailed analysis.

## 3. Adding Parameter Tokens to Seq2Seq

In this section we present ACCESS, our approach for **A**udien**C**e-**CE**ntric **S**entence **S**implification. We want to control the process of Sentence Simplification using explicit parameter tokens. We first identify attributes that cover important aspects of the simplification process and then find explicit parameter tokens to represent each of those attributes. Parametrization is then achieved by conditioning a Seq2Seq model on those parameter tokens.

### 3.1. Controlled attributes

Based on previous findings, we identify four attributes related to the process of text simplification: amount of compression, amount of paraphrasing, lexical complexity and syntactic complexity,.

- **Amount of compression:** The amount of compression is directly dependent on the length of sentences which is itself very correlated to simplicity (Martin et al., 2019), and is one of the two variables used in FKGL (Kincaid et al., 1975). It also accounts for the amount of content that is preserved between the source and target text, and can therefore control the simplicity-adequacy trade-off that is witnessed in text simplification (Schwarzer and Kauchak, 2018).

- **Paraphrasing:** Paraphrasing is an important aspect for good text simplification systems (Wubben et al., 2012), especially because it allows the user from choosing if he prefers very safe simplifications (i.e. close to the source) or to try and simplify the input more at the cost of more mistakes when using imperfect systems. The amount of paraphrasing was also shown to correlate with human jugdment of meaning preservation and simplicity sometimes even more than traditional metrics such as BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016).

- **Lexical and Syntactic complexity:** (Shardlow, 2014) identified lexical simplification and syntactic simplification as core components of SS systems, which often decomposes there approach into these two subcomponents. Audiences also have different simplification needs along these two attributes. In order to understand a text correctly, second language learner will require a text with less complicated words. On the other hand, some specific types of aphasia will make people struggle more with complex syntactic structures, intricated clauses, and long sentence, thus requiring syntactic simplification.

Other more specific attributes could be considered such as the tense or the use passive-active voice. We only consider the previous attributes for simplicity and leave the rest for future work. We don't consider "readability" measured with FKGL because it is just a linear combination of other attributes, namely sentence length and word complexity.

## 3.2. Explicit parameter tokens

For each of the four aforementioned attributes, we choose an explicit "proxy" parameter token that can be computed using the source and simplified sentence and used as a plain text token. We describe these for explicit parameter tokens in this subsection.

- **NbChars:** character length ratio between source sentence and target sentence (compression level). This parameter token accounts for sentence compression, and content deletion. Previous work showed that simplicity is best correlated with length-based metrics, and especially in terms of number of characters (Martin et al., 2019). The number of characters indeed accounts for the lengths of words which is itself correlated to lexical complexity.
- **LevSim:** normalized character-level Levenshtein similarity (Levenshtein, 1966) between source and target. LevSim quantifies the amount of modification operated on the source sentence (through paraphrasing, adding and deleting content).
- **WordRank:** as a proxy to lexical complexity, we compute a sentence-level measure, that we call *WordRank*, by taking the third-quartile of log-ranks (inverse frequency order) of all words in a sentence. We subsequently divide the *WordRank* of the target by that of the source to get a ratio. Word frequencies have shown to be the best indicators of word complexity in the Semeval 2016 task 11 (Paetzold and Specia, 2016).
- **DepTreeDepth:** maximum depth of the dependency tree of the source divided by that of the target (we do not feed any syntactic information other than this ratio to the model). This parameter token is designed to approximate syntactic complexity. Deeper dependency trees indicate dependencies that span longer and possibly more intricate sentences. DepTreeDepth proved better in early experiments over other candidates for measuring syntactic complexity such as the maximum length of a dependency relation, or the maximum inter-word dependency flux.

We parametrize a Seq2Seq model on a given attribute of the target simplification, e.g. its length, by prepending a special token at the beginning of the source sentence. The special token value is the ratio[1] of this parameter token calculated on the target sentence with respect to its value on the source sentence. For example when trying to control the number of characters of a generated simplification, we compute the compression ratio between the number of characters in the source and the number of characters in the target sentence (see Table 1 for an illustration). Ratios are discretized into bins of fixed width of 0.05 in our experiments and capped to a maximum ratio of 2. Special tokens are then included in the vocabulary (40 unique values per parameter token). At inference time, we just set the ratio to a fixed value for all samples[2]. For instance, to get simplifications that are 80% of the source length, we prepend the token <NbChars_0.8> to each source sentence. This fixed ratio can be user-defined or automatically set. In our setting, we choose fixed ratios that maximize the SARI on the validation set.

| | |
|---|---|
| Source | *<NbChars_0.3> <LevSim_0.4> He settled in London , devoting himself chiefly to practical teaching .* |
| Target | *He teaches in London .* |

Table 1: Example of parametrization on the number of characters. Here the source and target simplifications respectively contain 71 and 22 characters which gives a compression ratio of 0.3. We prepend the <NbChars_0.3> token to the source sentence. Similarly, the Levenshtein similarity between the source and the sentence is 0.37 which gives the <LevSim_0.4> special token after bucketing.

## 4. Experiments

### 4.1. Experimental Setting

**Architecture details** We train a Transformer model (Vaswani et al., 2017) using the FairSeq toolkit (Ott et al., 2019). Our architecture is the base architecture from (Vaswani et al., 2017). We used an embedding dimension of 512, fully connected layers of dimension 2048, 8 attention heads, 6 layers in the encoder and 6 layers in the decoder. Dropout is set to 0.2. We use the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and a learning rate of $lr = 0.00011$. We add label smoothing with a uniform prior distribution of $\epsilon = 0.54$. We use early stopping when SARI does not increase for more than 5 epochs. We tokenize sentences using the NLTK NIST tokenizer and preprocess using SentencePiece (Kudo and Richardson, 2018) with 10k vocabulary size to handle rare and unknown words. For generation we use beam search with a beam size of 8. [3]

**Training and evaluation datasets** Our models are trained and evaluated on the **WikiLarge dataset** (Zhang and Lapata, 2017) which contains 296,402/2,000/359 samples (train/validation/test). WikiLarge is a set of automatically aligned complex-simple sentence pairs from English Wikipedia (EW) and Simple English Wikipedia (SEW). It is compiled from previous extractions of EW-SEW (Zhu et al., 2010; Woodsend and Lapata, 2011; Kauchak, 2013). Its validation and test sets are taken from Turkcorpus (Xu

[1]Early experiments showed that using a ratio instead of an absolute value allowed finer control on the respective attributes.

[2]We did not investigate predicting ratios on a per sentence basis as done by Scarton and Specia (2018), and leave this for future work. End-users can nonetheless choose the target ratios as they see fit, for each source sentence.

[3]Code and pretrained models are released with an open-source license at `https://github.com/facebookresearch/access`.

et al., 2016), where each complex sentence has 8 human simplifications created by Amazon Mechanical Turk workers. Human annotators were instructed to only paraphrase the source sentences while keeping as much meaning as possible. Hence, no sentence splitting, minimal structural simplification and little content reduction occurs in this test set (Xu et al., 2016). We are not able to use the Newsela dataset (Xu et al., 2015) because of legal constraints related to its limited public availability. The Newsela dataset can only be accessed by signing a one year Data Sharing Agreement and comes with a restrictive non-commercial license. Additionally, all publications using the dataset need to be sent in advance to Newsela for approval. This limited public availability also prevents the research community from agreeing on a public train/validation/test split which hampers reproducibility of results.

**Evaluation metrics** We evaluate our methods with **FKGL** (Flesch-Kincaid Grade Level) (Kincaid et al., 1975) to account for simplicity and **SARI** (Xu et al., 2016) as an overall score. FKGL is a commonly used metric for measuring readability however it should not be used alone for evaluating systems because it does not account for grammaticality and meaning preservation (Wubben et al., 2012). It is computed as a linear combination of the number of words per simple sentence and the number of syllables per word:

$$FKGL = 0.39\frac{nb\ words}{nb\ sentences} + 11.8\frac{nb\ syllables}{nb\ words} - 15.59$$

On the other hand SARI compares the predicted simplification with both the source *and* the target references. It is an average of F1 scores for three $n$-gram operations: additions, keeps and deletions[4]. For each operation, these scores are then averaged for all $n$-gram orders (from 1 to 4) to get the overall F1 score.

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

$$f_{ope}(n) = \frac{2 \times p_{ope}(n) \times r_{ope}(n)}{p_{ope}(n) + r_{ope}(n)}$$

$$F_{ope} = \frac{1}{k} \sum_{n=[1,..,k]} f_{ope}(n)$$

$$SARI = \frac{F_{add} + F_{keep} + F_{del}}{3}$$

We compute FKGL and SARI using the EASSE python package for SS (Alva-Manchego et al., 2019). We do not use BLEU because it is not suitable for evaluating SS systems (Sulem et al., 2018). BLEU is also misleading because it favors models that do not modify the source sentence (Xu et al., 2016) on TurkCorpus. For instance copying the source sentence in place of simplification gives a BLEU of 99.37 on WikiLarge.

[4]Following Zhang and Lapata (2017), our SARI implementation includes deletion recall to match previous work.

| WikiLarge **(test)** | SARI ↑ | FKGL ↓ |
|---|---|---|
| PBMT-R | 38.56 | 8.33 |
| Hybrid | 31.40 | **4.56** |
| SBMT+PPDB+SARI | 39.96 | 7.29 |
| DRESS-LS | 37.27 | 6.62 |
| Pointer+Ent+Par | 37.45 | — |
| NTS+SARI | 37.25 | — |
| NSELSTM-S | 36.88 | — |
| DMASS+DCSS | 40.45 | 8.04 |
| **ACCESS:** $\text{NbChars}_{0.95}$ + $\text{LevSim}_{0.75}$ + $\text{WordRank}_{0.75}$ | **41.87** | 7.22 |

Table 2: Comparison to the literature. We report the results of the model that performed the best on the validation set among all runs and parametrizations. The ratios used for parametrizations are written as subscripts.

## 4.2. Overall Performance

Table 2 compares our best model to state-of-the-art methods:

**PBMT-R** (Wubben et al., 2012)
Phrase-Based MT system with candidate reranking. Dissimilar candidates are favored based on their Levenshtein distance to the source.

**Hybrid** (Narayan and Gardent, 2014)
Deep semantics sentence representation fed to a monolingual MT system.

**SBMT+PPDB+SARI** (Xu et al., 2016)
Syntax-based MT model augmented using the PPDB paraphrase database (Pavlick et al., 2015) and fine-tuned towards SARI.

**DRESS-LS** (Zhang and Lapata, 2017)
Seq2Seq trained with reinforcement learning, combined with a lexical simplification model.

**Pointer+Ent+Par** (Guo et al., 2018)
Seq2Seq model based on the pointer-copy mechanism and trained via multi-task learning on the Entailment and Paraphrase Generation tasks.

**NTS+SARI** (Nisioi et al., 2017)
Standard Seq2Seq model. The second beam search hypothesis is selected during decoding; the hypothesis number is an hyper-parameter fine-tuned with SARI.

**NSELSTM-S** (Vu et al., 2018)
Seq2Seq with a memory-augmented Neural Semantic Encoder, tuned with SARI.

**DMASS+DCSS** (Zhao et al., 2018)
Seq2Seq integrating the simple PPDB simplification database (Pavlick and Callison-Burch, 2016) as a dynamic memory. The database is also used to modify the loss and re-weight word probabilities to favor simpler words.

We select the model with the best SARI on the validation set and report its score on the test set. This model uses three

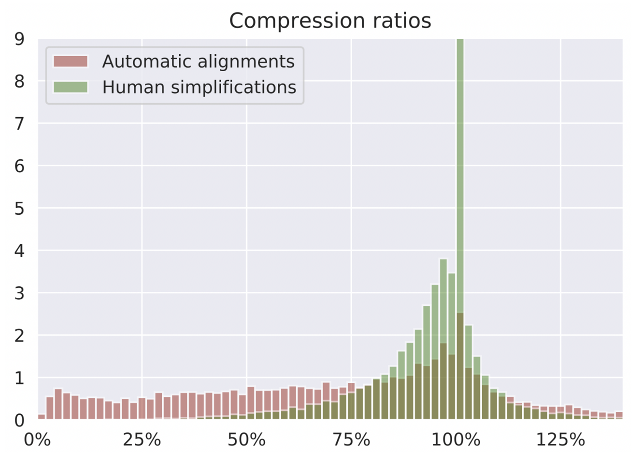


Figure 1: Density distribution of the **compression ratios** between the source sentence and the target sentence. The automatically aligned pairs from WikiLarge train set are spread (red) while human simplifications from the validation and test set (green) are gathered together with a mean ratio of 0.93 (i.e. nearly no compression).

parameter tokens out of four: NbChars$_{0.95}$, LevSim$_{0.75}$ and WordRank$_{0.75}$ (optimal target ratios in subscript).
ACCESS scores best on SARI (41.87), a significant improvement over previous state of the art (40.45), and third to best FKGL (7.22). The second and third models in terms of SARI, DMASS+DCSS (40.45) and SBMT+PPDB+SARI (39.96), both use the external resource Simple PPDB (Pavlick and Callison-Burch, 2016) that was extracted from 1000 times more data than what we used for training. Our FKGL is also better (lower) than these methods. The Hybrid model scores best on FKGL (4.56) i.e. they generated the simplest (and shortest) sentences, but it was done at the expense of SARI (31.40).
Parametrization encourages the model to rely on explicit aspects of the simplification process, and to associate them with the parameter tokens. The model can then be adapted more precisely to the type of simplification needed. In WikiLarge, for instance, the compression ratio distribution is different than that of human simplifications (see Figure 1). The NbChars parameter token helps the model decorrelate the compression aspect from other attributes of the simplification process. This parameter token is then adapted to the amount of compression required in a given evaluation dataset, such as a true, human simplified SS dataset. Our best model indeed worked best with a NbChars target ratio set to 0.95 which is the closest bucketed value to the compression ratio of human annotators on the WikiLarge validation set (0.93).

## 5. Ablation Studies

In this section we investigate the contribution of each parameter token to the final SARI score of ACCESS. Table 3 reports scores of models trained with different combinations of parameter tokens on the WikiLarge validation set (2000 source sentences, with 8 human simplifications each). We combined parameter tokens using greedy forward selection; at each step, we add the parameter token

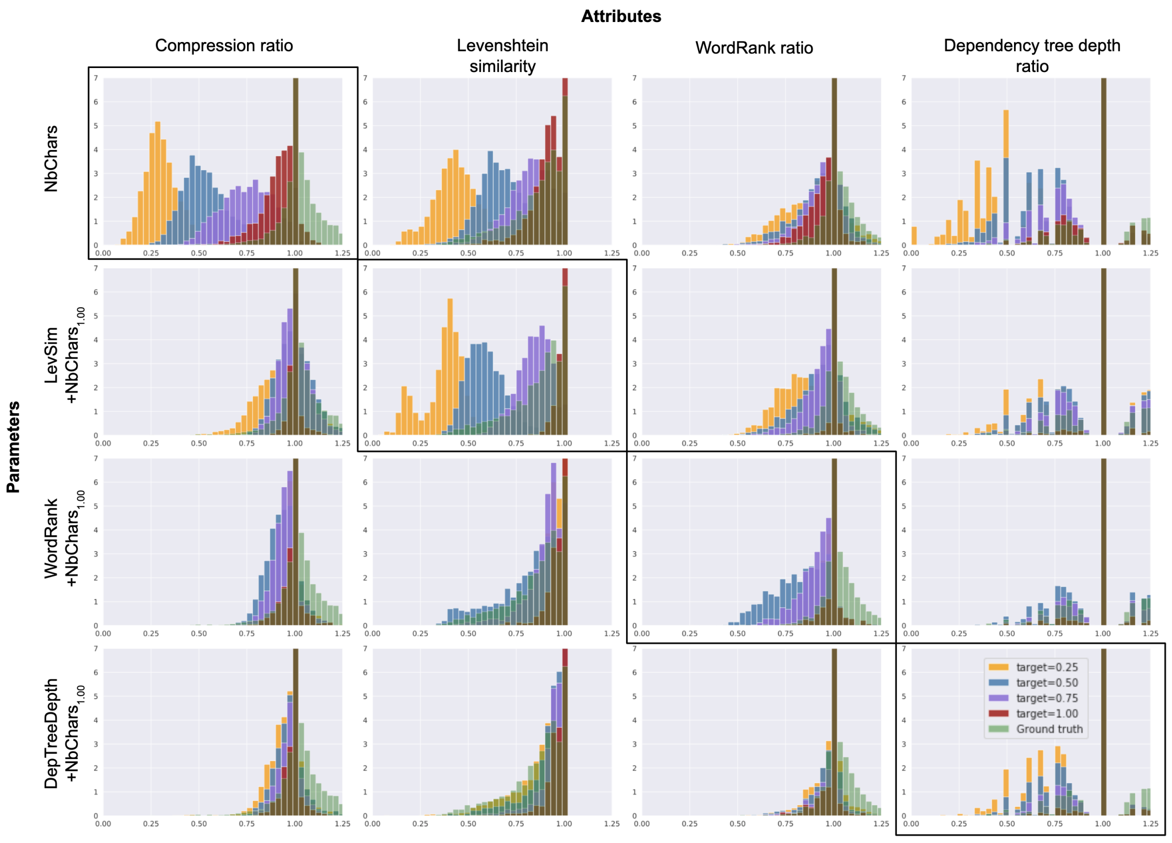
| WikiLarge **(validation)** | SARI ↑ | FKGL ↓ |
|---|---|---|
| Transformer | $37.06 \pm 0.25$ | $7.66 \pm 0.42$ |
| +DepTreeDepth | $37.72^* \pm 0.18$ | $7.64 \pm 0.22$ |
| +NbChars | $37.94^* \pm 0.09$ | $7.87 \pm 0.15$ |
| +LevSim | $38.29^* \pm 0.66$ | $7.53 \pm 0.21$ |
| +WordRank | $39.35^* \pm 0.25$ | $7.61 \pm 0.19$ |
| +WordRank+LevSim | $41.1^* \pm 0.14$ | $6.86^* \pm 0.17$ |
| +WordRank+LevSim +NbChars | $\mathbf{41.29}^* \pm 0.27$ | $7.25^* \pm 0.26$ |
| *all* | $41.03^* \pm 0.39$ | $\mathbf{6.72}^* \pm 0.39$ |

Table 3: Ablation study on the parameters using greedy forward selection. We report SARI and FKGL on WikiLarge **validation** set. Each score is a mean over 10 runs with a 95% confidence interval. Scores with $^*$ are statistically significantly better than the Transformer baseline (p-value $< 0.01$ for a Student's T-test).

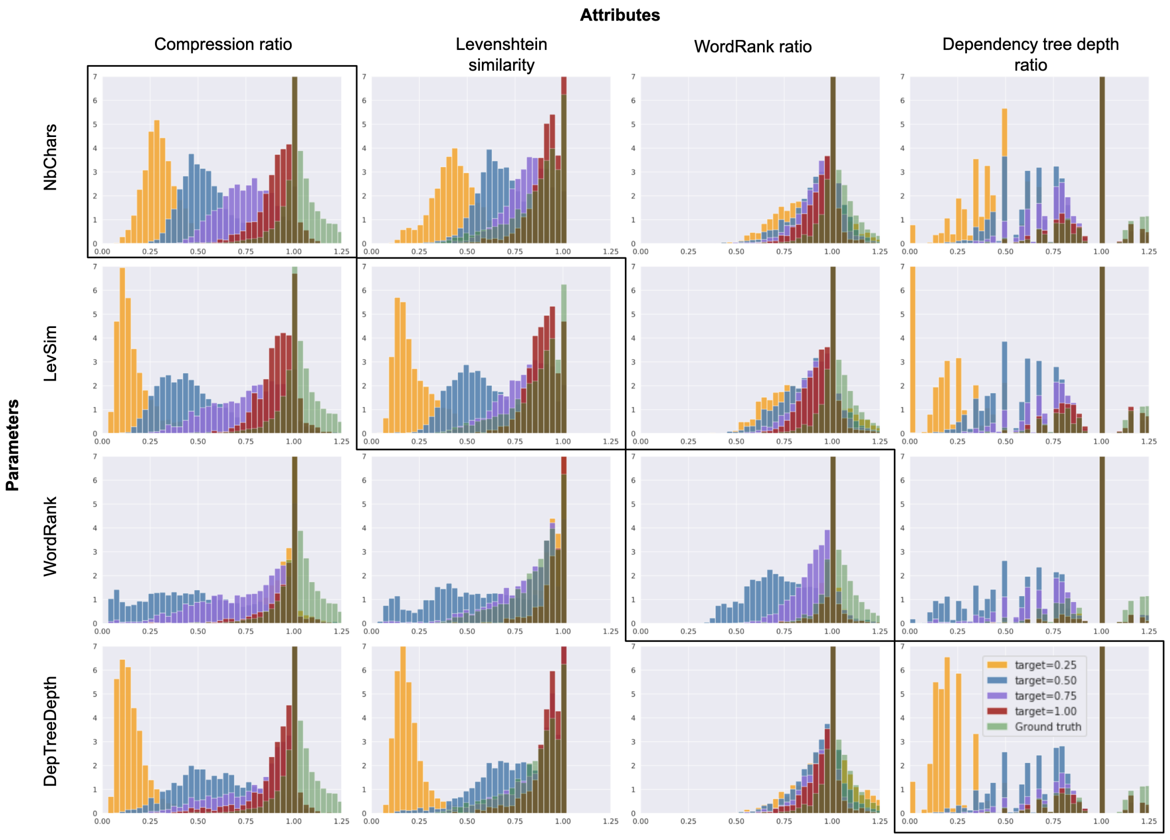
leading to the best performance when combined with previously added parameter tokens.
With only one parameter token, WordRank proves to be best (+2.28 SARI over models without parametrization). As the WikiLarge validation set mostly contains small paraphrases, it seems natural that the parameter token linked to lexical simplification increases the performance the most.
LevSim (+1.23) is the second best parameter token. This confirms the intuition that hypotheses that are more dissimilar to the source are better simplifications, as claimed in (Wubben et al., 2012; Nisioi et al., 2017).
There is little content reduction in the WikiLarge validation set (see Figure 1), thus parameter tokens that are closely related to sentence length will be less effective. This is the case for the NbChars and DepTreeDepth parameter tokens (shorter sentences, will have lower tree depths): they bring more modest improvements, +0.88 and +0.66.
The performance boost is nearly additive at first when adding more parameter tokens (WordRank+LevSim: +4.04) but saturates quickly with 3+ parameter tokens. In fact, no combination of 3 or more parameter tokens gets a statistically significant improvement over the WordRank+LevSim setup (p-value $<$ 0.01 for a Student's T-test). This indicates that parameter tokens are not all useful to improve the scores on this benchmark, and that they might be not independent from one another. The addition of the DepTreeDepth as a final parameter token even decreases the SARI score slightly, most probably because the considered validation set does not include sentence splitting and structural modifications.

## 6. Analysis of Parameter Tokens' Influence

Our goal is to give the user control over how the model will simplify sentences on four important attributes of SS: length, paraphrasing, lexical complexity and syntactic complexity. To this end, we introduced four parameter tokens: NbChars, LevSim, WordRank and DepTreeDepth. Even though the parameter tokens improve the performance in terms of SARI, it is not sure whether they have the desired



(a) With the NbChars$_{1.00}$ constraint.



(b) Without the NbChars$_{1.00}$ constraint.

Figure 2: Influence of each parameter token on the corresponding attributes of the output simplifications. **Rows represent parameter tokens** (each model is trained either only with one parameter token or with one parameter token and the NbChars$_{1.00}$ constraint), **columns represent output attributes** of the predictions and **colors represent the fixed target ratio** of the parameter token (yellow=0.25, blue=0.50, violet=0.75, red=1.00, green=Ground truth). We plot the results on the 2000 validation sentences. Figure 2a uses the NbChars$_{1.00}$ constraint, whereas Figure 2b doesn't.

| Target parameter tokens | Sentence |
|---|---|
| **Source** | Some trails are designated as nature trails , and are used by people learning about the natural world . |
| **NbChars$_{1.00}$** | Some trails are **called** nature trails , and are used by people about the natural world . |
| **NbChars$_{0.75}$** | Some trails are **called** nature trails , and are used by people about the natural world . |
| **NbChars$_{0.50}$** | Some trails are used by people about the natural world . |
| **NbChars$_{0.25}$** | Some trails are used by people . |
| **LevSim$_{1.00}$**+NbChars$_{1.00}$ | Some trails are designated as nature trails , and are used by people learning about the natural world . |
| **LevSim$_{0.75}$**+NbChars$_{1.00}$ | Some trails are **made for** nature trails **. They** are used by people **who learn** about the natural world . |
| **LevSim$_{0.50}$**+NbChars$_{1.00}$ | **The** trails **that** are used by people learning about the natural world **, because the trails are good trails** . |
| **LevSim$_{0.25}$**+NbChars$_{1.00}$ | **Mechanical** trails **( also known as " trail trail " or " trails " )** are trails **that** are used **for trails** . |
| **WordRank$_{1.00}$**+NbChars$_{1.00}$ | Some trails are designated as nature trails , and are used by people learning about the natural world . |
| **WordRank$_{0.75}$**+NbChars$_{1.00}$ | Some trails are **called** nature trails , and are used by people learning about the natural world . |
| **WordRank$_{0.50}$**+NbChars$_{1.00}$ | Some trails are **known** as nature trails , and are used by people **as well as by people who are in** the world . |
| **WordRank$_{0.25}$**+NbChars$_{1.00}$ | Some trails are **also called** nature trails , and are used by people learning about the natural world . |
| **DepTreeDepth$_{1.00}$**+NbChars$_{1.00}$ | Some trails are designated as nature trails , and are used by people learning about the natural world . |
| **DepTreeDepth$_{0.75}$**+NbChars$_{1.00}$ | Some trails are designated as nature trails **. They** are used by people learning about the natural world . |
| **DepTreeDepth$_{0.50}$**+NbChars$_{1.00}$ | Some trails are designated as nature trails **. They** are used by people learning about the natural world . |
| **DepTreeDepth$_{0.25}$**+NbChars$_{1.00}$ | Some trails are designated as nature trails **. They** are used by people **to learn** about the natural world . |
| **Source** | Iron Maiden , released on April 14 , 1980 , is the debut album by heavy metal band Iron Maiden . |
| **NbChars**$_{1.00}$ | Iron Maiden , released on April 14 , 1980 , is the debut album by heavy metal band Iron Maiden . |
| **NbChars**$_{0.75}$ | Iron Maiden is the debut album by heavy metal band Iron Maiden . |
| **NbChars**$_{0.50}$ | Iron Maiden , released on April 14 , 1980 . |
| **NbChars**$_{0.25}$ | Iron Maiden **was** released on April 14 , 1980 . |
| **LevSim**$_{1.00}$+NbChars$_{1.00}$ | Iron Maiden , released on April 14 , 1980 , is the debut album by heavy metal band Iron Maiden . |
| **LevSim**$_{0.75}$+NbChars$_{1.00}$ | Iron Maiden is the debut album by heavy metal band Iron Maiden . **It was released on April 14 , 1980 .** |
| **LevSim**$_{0.50}$+NbChars$_{1.00}$ | Iron Maiden is the debut album by heavy metal band Iron Maiden . **It was released on April 14 , 1980 .** |
| **LevSim**$_{0.25}$+NbChars$_{1.00}$ | Iron Maiden is **an** album by **American** heavy metal band Iron Maiden . **It was released on April 14 , 1980 .** |
| **WordRank**$_{1.00}$+NbChars$_{1.00}$ | Iron Maiden is the **first** album **released** by heavy metal band Iron Maiden . **It was released in 1980 .** |
| **WordRank**$_{0.75}$+NbChars$_{1.00}$ | Iron Maiden **is a first album by the band Iron Maiden . It was** released on April 14 , 1980 . |
| **WordRank**$_{0.50}$+NbChars$_{1.00}$ | Iron Maiden is **a city of** the **state of Arkansas in the United States of America** . |
| **WordRank**$_{0.25}$+NbChars$_{1.00}$ | Iron Maiden **is a first album** released **by the band Iron Maiden . It was released** on April 14 , 1980 . |
| **DepTreeDepth**$_{1.00}$+NbChars$_{1.00}$ | Iron Maiden , released on April 14 , 1980 , is the **first** album by heavy metal band Iron Maiden . |
| **DepTreeDepth**$_{0.75}$+NbChars$_{1.00}$ | Iron Maiden is **a first** album by **British** heavy metal band Iron Maiden . **It was released on April 14 , 1980 .** |
| **DepTreeDepth**$_{0.50}$+NbChars$_{1.00}$ | Iron Maiden is **an** album by **British** heavy metal band Iron Maiden . **It was released on April 14 , 1980 .** |
| **DepTreeDepth**$_{0.25}$+NbChars$_{1.00}$ | Iron Maiden **was** released on April 14 , 1980 **. It was released in** Iron Maiden **on April 14 , 1980** . |
| **Source** | Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy . |
| **NbChars**$_{1.00}$ | Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy . |
| **NbChars**$_{0.75}$ | Nocturnes is an orchestral composition in three movements by the French composer Claude **Debus** . |
| **NbChars**$_{0.50}$ | Nocturnes is an orchestral composition in three movements . |
| **NbChars**$_{0.25}$ | Nocturnes is an orchestral composition . |
| **LevSim**$_{1.00}$+NbChars$_{1.00}$ | Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy . |
| **LevSim**$_{0.75}$+NbChars$_{1.00}$ | Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy . |
| **LevSim**$_{0.50}$+NbChars$_{1.00}$ | Nocturnes is **a piece of music for orchestra that was composed** by **a** French composer **called** Claude Debussy . |
| **LevSim**$_{0.25}$+NbChars$_{1.00}$ | **Claude Debussy was a** French composer **who wrote music for the orchestra when he was 17 years old** . |
| **WordRank**$_{1.00}$+NbChars$_{1.00}$ | Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy . |
| **WordRank**$_{0.75}$+NbChars$_{1.00}$ | Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy . |
| **WordRank**$_{0.50}$+NbChars$_{1.00}$ | Nocturnes is **a piece of music** by the French composer Claude Debussy . |
| **WordRank**$_{0.25}$+NbChars$_{1.00}$ | Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy . |
| **DepTreeDepth**$_{1.00}$+NbChars$_{1.00}$ | Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy . |
| **DepTreeDepth**$_{0.75}$+NbChars$_{1.00}$ | Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy . |
| **DepTreeDepth**$_{0.50}$+NbChars$_{1.00}$ | Nocturnes is an orchestral composition by the French composer Claude Debussy **in three movements** . |
| **DepTreeDepth**$_{0.25}$+NbChars$_{1.00}$ | Nocturnes is **a French orchestra . It was started by Claude Debussy** in three movements . |
| **Source** | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **NbChars**$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **NbChars**$_{0.75}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) . |
| **NbChars**$_{0.50}$ | **This** means that it is very dark in colouring ( darker than soot ) . |
| **NbChars**$_{0.25}$ | It is an F-type asteroid . |
| **LevSim**$_{1.00}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **LevSim**$_{0.75}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) **made up of carbonate metal** . |
| **LevSim**$_{0.50}$+NbChars$_{1.00}$ | F-type **asteroids can be made up of darker than soot ( darker than soot )** **, or darker** ( darker than soot ) **, or dark ( darker )** . |
| **LevSim**$_{0.25}$+NbChars$_{1.00}$ | **IAUC 2003 September 6 ( naming the moon ) was discovered by Eros in 2005 by E. H. E. E. J. E. J. J. J. J. J. J. J. J. R. J.** [...] |
| **WordRank**$_{1.00}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **WordRank**$_{0.75}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a **made of carbonate** . |
| **WordRank**$_{0.50}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a **very dark made up of** . |
| **WordRank**$_{0.25}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **DepTreeDepth**$_{1.00}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **DepTreeDepth**$_{0.75}$+NbChars$_{1.00}$ | It is an F-type asteroid , which means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **DepTreeDepth**$_{0.50}$+NbChars$_{1.00}$ | It is an F-type asteroid **. It** means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |
| **DepTreeDepth**$_{0.25}$+NbChars$_{1.00}$ | It is an F-type asteroid **. It** means that it is very dark in colouring ( darker than soot ) with a carbonaceous composition . |

Table 4: Influence of parameter tokens on example sentences. Each source sentence is simplified with models trained with each of the four parameter tokens with varying target ratios; modified words are in bold. The NbChars$_{1.00}$ constraint is added for LevSim, WordRank and DepTreeDepth.

effect on their associated attribute. In this section we investigate to what extent each parameter token controls the generated simplification. We first used separate models, each trained with a single parameter token to isolate their respective influence on the output simplifications. However, we witnessed that with only one parameter token, the effect of LevSim, WordRank and DepTreeDepth was mainly to reduce the length of the sentence (Figure 2b). Indeed, shortening the sentence will decrease the Levenshtein similarity, decrease the WordRank (when complex words are deleted) and decrease the dependency tree depth (shorter sentences have shallower dependency trees). Therefore, to clearly study the influence of those parameter tokens, we also add the NbChars parameter token during training, and set its ratio to 1.00 at inference time, as a constraint toward not modifying the length.

Figure 2a highlights the cross influence of each of the four parameter tokens on their four associated attributes. Parameter tokens are successively set to ratios of 0.25 (yellow), 0.50 (blue), 0.75 (violet) and 1.00 (red); the ground truth is displayed in green. Plots located on the diagonal show that parameter tokens control their respective attributes (e.g. NbChars affects the compression ratio...), although not with the same effectiveness.

The histogram located at (row 1, col 1) shows the effect of the NbChars parameter token on the compression ratio of the predicted simplifications. The resulting distributions are centered on the 0.25, 0.5, 0.75 and 1 target ratios as expected, and with little overlap. This indicates that the lengths of predictions closely follow what is asked of the model. Table 4 illustrates this with an example. The NbChars parameter token affects Levenshtein similarity: reducing the length decreases the Levenshtein similarity. Finally, NbChars has a marginal impact on the WordRank ratio distribution, but clearly influences the dependency tree depth. This is natural considered that the depth of a dependency tree is very correlated with the length of the sentence.

The LevSim parameter token also has a clear cut impact on the Levenshtein similarity (row 2, col 2). The first example in Table 4 highlights that LevSim increases the amount of paraphrasing in the simplifications. With an extreme target ratio of 0.25, the model outputs ungrammatical and meaningless predictions, thus indicating that the choice of a target ratio is important for generating proper simplifications.

WordRank and DepTreeDepth do not seem to control their respective attribute as well as NbChars and LevSim according to Figure 2a. However we witness more lexical simplifications when using the WordRank ratio than with other parameter tokens. In Table 4's first example, "designated as" is simplified by "called" or "known as" with the WordRank parameter token. Equivalently, DepTreeDepth splits the source sentence in multiple shorter sentences in Table 4's first example. WordRank and DepTreeDepth parameter tokens therefore have the desired effect.

## 7. Conclusion

This paper showed that explicitly conditioning Seq2Seq models on parameter tokens such as length, paraphrasing, lexical complexity or syntactic complexity increases their performance significantly for sentence simplification. We confirmed through an analysis that each parameter token has the desired effect on the generated simplifications. In addition to being easy to extend to other attributes of text simplification, our method paves the way toward adapting the simplification to audiences with different needs.

## 8. Bibliographical References

Alva-Manchego, F., Martin, L., Scarton, C., and Specia, L. (2019). Easse: Easier automatic sentence simplification evaluation. *arXiv preprint arXiv:1908.04567.*

Carroll, J., Minnen, G., Canning, Y., Devlin, S., and Tait, J. (1998). Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.

Coster, W. and Kauchak, D. (2011). Learning to simplify sentences using wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, pages 1–9. Association for Computational Linguistics.

Evans, R., Orasan, C., and Dornescu, I. (2014). An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 131–140.

Fan, A., Grangier, D., and Auli, M. (2017). Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217.*

Fevry, T. and Phang, J. (2018). Unsupervised sentence compression using denoising auto-encoders. *arXiv preprint arXiv:1809.02669.*

Ficler, J. and Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633.*

Ghazvininejad, M., Shi, X., Priyadarshi, J., and Knight, K. (2017). Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48.

Guo, H., Pasunuru, R., and Bansal, M. (2018). Dynamic multi-level multi-task learning for sentence simplification. *arXiv preprint arXiv:1806.07304.*

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Kauchak, D. (2013). Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, volume 1, pages 1537–1546.

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. (2016). Controlling output length in neural encoder-decoders. *arXiv preprint arXiv:1609.09552.*

Kincaid, J. P., Fishburne Jr., R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Mallinson, J., Sennrich, R., and Lapata, M. (2018). Sentence compression for arbitrary languages via multilingual pivoting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2453–2464.

Martin, L., Humeau, S., Mazaré, P.-E., Bordes, A., de La Clergerie, É. V., and Sagot, B. (2019). Reference-less quality estimation of text simplification systems. *arXiv preprint arXiv:1901.10746*.

Narayan, S. and Gardent, C. (2014). Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 435–445.

Nishihara, D., Kajiwara, T., and Arase, Y. (2019). Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266.

Nisioi, S., Štajner, S., Ponzetto, S. P., and Dinu, L. P. (2017). Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Paetzold, G. and Specia, L. (2016). Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pavlick, E. and Callison-Burch, C. (2016). Simple ppdb: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 143–148.

Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2015). Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 425–430.

Rello, L., Baeza-Yates, R., Bott, S., and Saggion, H. (2013). Simplify or help?: text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, page 15. ACM.

Scarton, C. and Specia, L. (2018). Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718.

Schwarzer, M. and Kauchak, D. (2018). Human evaluation for text simplification: The simplicity-adequacy tradeoff.

See, A., Roller, S., Kiela, D., and Weston, J. (2019). What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.

Shardlow, M. (2014). A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Sulem, E., Abend, O., and Rappoport, A. (2018). Bleu is not suitable for the evaluation of text simplification.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vu, T., Hu, B., Munkhdalai, T., and Yu, H. (2018). Sentence simplification with memory-augmented neural networks. *arXiv preprint arXiv:1804.07445*.

Watanabe, W. M., Junior, A. C., Uzêda, V. R., Fortes, R. P. d. M., Pardo, T. A. S., and Aluísio, S. M. (2009). Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 29–36. ACM.

Woodsend, K. and Lapata, M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the conference on empirical methods in natural language processing*, pages 409–420. Association for Computational Linguistics.

Wubben, S., Van Den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

Xia, M., Kochmar, E., and Briscoe, T. (2016). Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22.

Xu, W., Napoles, C., Pavlick, E., Chen, Q., and Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Zhang, X. and Lapata, M. (2017). Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

Zhao, S., Meng, R., He, D., Andi, S., and Bambang, P. (2018). Integrating transformer and paraphrase rules for sentence simplification. *arXiv preprint arXiv:1810.11193*.

Zhu, Z., Bernhard, D., and Gurevych, I. (2010). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics.

## 9. Language Resource References

Xu, Wei and Callison-Burch, Chris and Napoles, Courtney. (2015). *Problems in current text simplification research: New data can help*.

Xu, Wei and Napoles, Courtney and Pavlick, Ellie and Chen, Quanze and Callison-Burch, Chris. (2016). *Optimizing statistical machine translation for text simplification*.

Zhang, Xingxing and Lapata, Mirella. (2017). *Sentence simplification with deep reinforcement learning*.