A Non-Autoregressive Edit-Based Approach to Controllable Text Simplification

Sweta Agrawal, Weijia Xu, Marine Carpuat
Department of Computer Science
University of Maryland
{sweagraw, weijia, marine}@cs.umd.edu

Abstract

We introduce a new approach for the task of Controllable Text Simplification, where systems rewrite a complex English sentence so that it can be understood by readers at different grade levels in the US K-12 system. It uses a non-autoregressive model to iteratively edit an input sequence and incorporates lexical complexity information seamlessly into the refinement process to generate simplifications that better match the desired output complexity than strong autoregressive baselines. Analysis shows that our model's local edit operations are combined to achieve more complex simplification operations such as content deletion and paraphrasing, as well as sentence splitting.

1 Introduction

Text simplification (TS) aims to automatically rewrite text so that it is easier to read. What makes text simple depends on its target audience (Xu et al., 2015): replacing complex or specialized terms with simpler synonyms might be helpful for non-native speakers (Petersen and Ostendorf, 2007; Allen, 2009) whereas restructuring text into short sentences with simple words might better match the literacy skills of children (Watanabe et al., 2009). Studies of simplification tools for deaf or hard-of-hearing users also show that they prefer lexical simplification to be applied on-demand (Alonzo et al., 2020). Yet, research in TS has mostly focused on developing models that generate a generic simplified output for a given source text (Xu et al., 2015; Zhang and Lapata, 2017; Alva-Manchego et al., 2020). We contrast this Generic TS with Controllable TS which specifies desired output properties.

Prior work has addressed Controllable TS for either high-level properties, such as the target reading grade level for the entire text (Scarton and Specia, 2018; Nishihara et al., 2019), or low-level properties, such as the compression ratio or the nature of the simplification operation to use (Mallinson and Lapata, 2019; Martin et al., 2020; Maddela et al., 2020). Specifying the desired reading grade level might be more intuitive for lay users. However, it provides only weak control over the nature of simplification. As illustrated in Table 1, simplifying text to different grade levels results in diverse edits. To rewrite the grade 10 original for grade 5, the complex text is split into two sentences and paraphrased. When simplifying for grade 3, phrases are further simplified, and content is entirely deleted.

Grade	Text
10	Tesla is a maker of electric cars, which do not need gas and can be charged by being plugged into a wall socket.
5	Tesla cars can be charged by being plugged **in, like a phone. They** do not need any gas.
3	Tesla **builds** cars **that** do not need gas.

Table 1: Simplified text changes depending on the reading grade level of the target audience. The bold font highlights changes compared to the grade 10 version.

In this work, we adopt the intuitive framing for Controllable TS where the desired reading grade level is given as input, while providing fine-grained control on simplification by incorporating lexical complexity signals into our model. We adopt a non-autoregressive sequence-to-sequence model (Xu and Carpuat, 2020) that iteratively refines an input sequence to reach the desired degree of simplification and seamlessly integrate lexical complexity.

Unlike commonly used autoregressive (AR) models for simplification (Specia, 2010; Nisioi et al., 2017; Zhang and Lapata, 2017; Wubben et al., 2012; Scarton and Specia, 2018; Nishihara et al., 2019; Martin et al., 2020; Jiang et al., 2020, among others), our model relies on explicit edit

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3757–3769

operations. It therefore has the potential of modeling the simplification process more directly than AR models which need to learn to copy operations implicitly. Unlike existing edit-based models for simplification which rely on pipelines of independently trained components (Alva-Manchego et al., 2017; Malmi et al., 2019; Mallinson et al., 2020), our model is trained end-to-end via imitation learning and thus learns to apply sequences of edits to transform the original source into the final simplified text. Furthermore, our approach does not require a custom architecture for simplification: it repurposes a non-autoregressive (NAR) model introduced for Machine Translation (MT) and can seamlessly incorporate lexical complexity information derived from data statistics in the initial sequence to be refined.

Based on extensive experiments on the Newsela English corpus, we show that our approach generates simplified outputs that match the target reading grade level better than strong AR baselines. Further analysis shows that the model learns complex editing operations such as sentence splitting, substitution and paraphrasing, and content deletion and applies these operations accordingly to match the complexity of the desired grade level.

2 An Edit-based approach for Controllable TS

Task We frame Controllable TS as follows: given a complex text c and a target grade level g_t, the task consists in generating a simplified output s that is appropriate for grade level g_t.

Approach Our approach, illustrated in Figure 1, is based on EDITOR (Xu and Carpuat, 2020), a NAR Transformer model where the decoder layer is used to apply a sequence of edits on an initial input sequence (possibly empty). The edits are of two types: (1) reposition and (2) insertion. The reposition layer predicts the new position of each token (including deletions). The insertion layer has two components: the first layer predicts the number of placeholders to be inserted and the fill-in layer generates the actual target tokens for each placeholder. At each iteration, the model applies a reposition operation followed by insertion to the current input. This is repeated until two consecutive iterations return the same output, or a preset maximum number of operations is reached. We tailor EDITOR for the task of Controllable TS as follows:

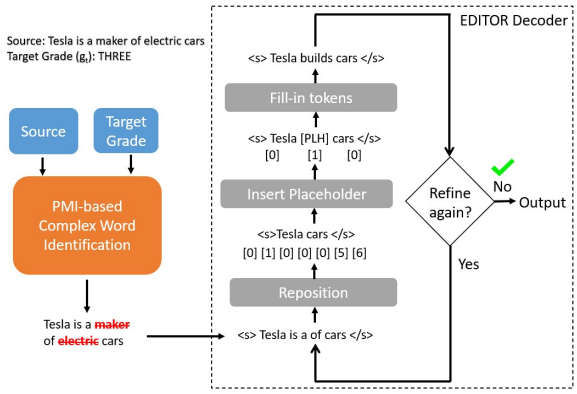


Figure 1: EDITOR iteratively refines a version of the input where words predicted to be too complex for 3rd grade readers have been deleted.

Control tokens The target complexity g_t is encoded as a special token added at the start of the input sequence. As in prior work with autoregressive models (Scarton and Specia, 2018; Nishihara et al., 2019), this token acts as a side-constraint, gets encoded in the encoder hidden states as any other vocabulary token, and informs hypothesis generation through the source-target attention mechanism.

Lexical complexity signals We automatically identify the source words that are too complex for the target grade and delete them from the initial sequence to be refined by EDITOR. This simple strategy provides finer-grained guidance to the simplification process than the sequence-level side-constraint, while leaving the EDITOR model the flexibility to rewrite the output without constraints. We quantify the relatedness between each vocabulary word (w) and grade-level (g) using their Pointwise Mutual Information (PMI) in the newsela corpus (Nishihara et al., 2019; Kajiwara, 2019):

$$PMI(w, g) = log\frac{p(w|g)}{p(w)} \quad (1)$$

Here, $p(w|g)$ is the probability that word w appears in sentences of grade level g and $p(w)$ is the probability of word w in the entire training corpus.

While the desired grade level g_t is known in the task, we automatically predict the complexity g_s of each source sentence s_i using the Automatic Readability Index (ARI; Senter and Smith (1967)). The initial decoding sequence $\hat{s_i}$ takes the source sequence and deletes all words that are strongly related to the source grade level and unlikely to be found in text of the target grade level, with the

exception of named entities:

$$\hat{s_i} = \{w | w \in s_i \wedge \quad \sim (PMI(w, g_s) > 0 \wedge PMI(w, g_t) < 0 \wedge \ w \notin E_i)\} \quad (2)$$

where, E_i represents the set of entities in the source sequence s_i. Our approach contrasts with prior work where PMI has been used in the loss to reward the generation of target grade-specific words for Controllable TS (Nishihara et al., 2019) or to exclude complex words from the decoding vocabulary using hard constraints for Generic TS (Kajiwara, 2019). Our approach combines lexical complexity information from both the source and target grade level more flexibly. Starting from $\hat{s_i}$ as an initial sequence, EDITOR can still delete further content to match the target grade level, insert new words to fix fluency and preserve the original meaning, and has the flexibility to re-generate tokens that were incorrectly dropped from the initial sequence.

Training to generate & refine EDITOR uses imitation learning to learn an appropriate sequence of edit operations to generate the output sequence by efficiently exploring the large space of valid edit sequences that can reach a reference output. A roll-in policy is used to generate sequences to be refined and a roll-out policy is then used to estimate cost-to-go for all possible actions given the roll-in sequences. The model is trained to choose actions that minimizes the cost-to-go estimates from the roll-in sequences to the true reference by comparing the model actions to the oracle actions generated by the Levenshtein edit distance algorithm. The roll-in sequences are stochastic mixtures of the initial sequences and outputs of the insertion and reposition modules given an initial sequence. The initial sequence is generated by applying random word dropping (Gu et al., 2019) and random word shuffle (Lample et al., 2018) with a probability of 0.5 and maximum shuffle distance of 3 to either the target sequence for MT tasks (Xu and Carpuat, 2020) or to the source sequence for Automatic Post Editing (Gu et al., 2019). For Controllable TS, we combine both, training EDITOR to generate text based on the corrupted *target sequence* first, and then fine-tuning the model for refinement based on the corrupted *source sequence* next.

3 Experimental Settings

3.1 Data

The Newsela website provides high quality data to study text simplification (Xu et al., 2015). It consists of news articles rewritten by professional editors for students in different grade levels. We use English Newsela samples as extracted by Agrawal and Carpuat (2019) since their process preserves grade level information for each segment. We restrict the length of each segment to be between 5 and 80 resulting in 470k/2k/19k for training, development and test sets respectively. We pre-process the dataset using Moses tools for normalization, and truecasing. We refer to the resulting dataset as **newsela-grade**. We further segment tokens into subwords using a joint source-target byte pair encoding model with $32,000$ operations. We use spacy[1] to identify entities in the source sequence.

3.2 Model configurations

Architecture We adopt the base Transformer architecture (Vaswani et al., 2017) with d_{model} = 512, d_{hidden} = 2048, n_{heads} = 8, n_{layers} = 6, and $p_{dropout}$ = 0.1 for all our models. We add dropout to embeddings (0.1) and label smoothing (0.1). AR models are trained with the Adam optimizer with a batch size of 4096 tokens. Training stops after 8 checkpoints without improvement of validation perplexity. We decode with a beam size of 5 for the AR models. All NAR models are trained using Adam with initial learning rate of 0.0005 and a batch size of 16,000 tokens. We select the best checkpoint based on validation perplexity. Grade side-constraints are defined using a distinct special token for each grade level (from 2 to 12). All models are implemented using the Fairseq toolkit.

Preliminary We establish that our Transformer architecture choice is strong on the more standard Generic TS task, as it performs comparably to the state-of-the-art[2] (Jiang et al., 2020) on the Newsela-Auto corpus (Table 2).[3]

Experimental Conditions We compare our approach, i.e.,"NAR + PMI-based initialization", described in Section 2 to three auto-regressive baselines for Controllable TS:

1. **AR** is a Transformer model which uses grade

[1] https://spacy.io/

[2] The Bert-initialized Transformer has parameters d_{model} = 768, d_{hidden} = 3072, n_{heads} = 12, n_{layers} = 12, and $p_{dropout}$ = 0.1. The encoder and decoder follow the BERT-base architecture. The encoder is initialized with a pre-trained checkpoint and the decoder is randomly initialized.

[3] This corpus contains complex-simple pairs extracted from 1,506 articles for training, 188 for validation and 188 for testing for Generic TS.

	SARI	add-F1	keep-F1	del-P
Results as reported in Jiang et al. (2020)				
EditNTS	35.8	2.4	29.4	**75.6**
Transformer-BERT	36.6	4.5	31.0	74.3
AR Transformer (ours)	36.1	3.8	**33.5**	71.1

Table 2: Generic TS Evaluation on Newsela-Auto: our Transformer baseline is comparable to SOTA models.

level tokens as side constraints (Scarton and Specia, 2018).

2. **AR + PMI-based constraints** is an AR Transformer model which incorporates lexical complexity information as hard constraints during decoding (Kajiwara, 2019): complex words are excluded from beam search using the dynamic beam allocation algorithm (Post and Vilar, 2018). While this approach was introduced for Generic TS, we adapt it to Controllable TS by defining hard constraints using the same criteria as for deleting words in initial sequences for EDITOR (Section 2).

3. **AR + PMI weighted loss** (Nishihara et al., 2019) is an AR Transformer model trained with a loss that weights words based on their PMI values with the desired target grade level.

3.3 Automatic Evaluation Metrics

We evaluate the output of the models using the following text simplification evaluation metrics:

SARI (Xu et al., 2016) measures lexical simplification based on the words that are added, deleted and kept by the systems by comparing system output against references and against the input sentence. It computes the F1 score for the n-grams that are added (add-F1). The model's deletion capability is measured by the F1 score for n-grams that are kept (keep-F1) and precision for the deleted n-grams (del-P) [4].

Pearson's correlation coefficient (PCC) measures the strength of the linear relationship between the complexity of our system outputs and the complexity of reference outputs. We estimate the reading grade level of the system outputs and reference text using the ARI score.

[4] https://github.com/cocoxu/simplification

Adjacency ARI Accuracy represents the percentage of sentences where the system output grade level is within 1 grade of the reference text according to the ARI score (Heilman et al., 2008).

Mean Squared Error (MSE) between the predicted ARI grade level of the system output and the desired target grade level (Scarton and Specia, 2018; Nishihara et al., 2019).

4 Evaluation of Controllable TS

4.1 Automatic Evaluation

Table 3 summarizes the automatic evaluation of our approach on Controllable TS: `our approach`, "NAR + PMI-based initialization", improves all metrics—SARI, PCC, ARI-accuracy and MSE—compared to the AR baselines. It also outperforms the `AR + PMI-based constraints` baseline across all metrics except MSE which oversimplifies the source text by always deleting the complex tokens, as shown by a decrease in keep-F1 (-6.1) and improved del-P (+4.6). This results in lower MSE but worse PCC and ARI-Accuracy. By contrast, `our approach` uses lexical complexity information to provide an initial canvas and yields simplified sentences that match the desired target complexity better than the AR baselines. This is reflected in the higher SARI obtained by the `PMI-based initialization` baseline relative to the `Source`, which represents outputs generated by deleting complex tokens from the source text and hence by itself is not a well-formed. `AR + PMI weighted loss` performs comparably to the `AR` baseline across all the metrics except PCC, which could be due to PMI values being a relatively noisy signal at the token level during training, especially for the target grade levels where the data is scarce.[5]

We further compare `our approach` with the model that uses the `Oracle-keep` sequence, i.e., tokens from the source sequence that are present in the target sequence. As expected, the oracle significantly outperforms all models that do not have access to the reference, further confirming EDITOR's ability to make good use of the provided initial sequence. More interestingly, our method for identifying grade-specific complex tokens (Equation 2) achieves a recall of 91.3% and precision of

[5] We note that the improvement in SARI reported in prior work (Nishihara et al. (2019): +0.15) is within the confidence interval (+0.5) of the `AR` baseline (Table 3).

Model	SARI ↑	keep-F1 ↑	add-F1 ↑	del-P ↑	%PCC ↑	%ARI-Acc ↑	MSE ↓	%Unchanged
Source	22.5	67.7	0.0	0.0	63.2	29.4	4.92	100.0
Reference	91.1	98.9	88.1	86.3	100.0	100.0	1.91	10.6
PMI-based initialization	37.1	60.5	1.3	49.3	61.4	26.4	3.69	3.4
AR	38.7 ±0.5	68.3 ±0.3	4.6 ±0.3	43.2 ±1.4	73.0 ±0.3	37.1 ±0.4	3.39 ±0.24	36.2
+ PMI-based constraints	38.3 ±0.1	62.2 ±0.7	4.9 ±0.3	47.8 ±0.3	69.1 ±0.2	35.0 ±0.5	2.21 ±0.22	12.1
+ PMI weighted loss	38.5 ±0.5	68.2 ±0.3	4.5 ±0.3	42.9 ±1.3	72.4 ±0.2	36.6 ±0.5	3.32 ±0.21	35.9
NAR	39.1 ±0.1	66.7 ±0.1	3.1 ±0.1	47.6 ±0.4	73.1 ±0.1	36.4 ±0.0	3.56 ±0.04	17.7
+ PMI-based initialization **(our approach)**	39.7 ±0.1	66.5 ±0.1	3.5 ±0.1	49.0 ±0.4	73.7 ±0.0	38.1 ±0.1	3.30 ±0.05	16.0
Oracle-keep	41.8 ±0.3	70.0 ±0.1	5.0 ±0.1	50.3 ±0.7	75.6 ±0.3	41.8 ±0.3	2.97 ±0.06	16.9

Table 3: Automatic evaluation results on Newsela-Grade test set: our approach outperforms AR baselines on SARI, PCC and ARI accuracy.

(Grade 12) Source:		The researchers **analyzed** a **national** injury **database** operated by the Consumer Product Safety Commission. Their study covers the early years of **commercial zip lines**, which now number more than 200 nationwide.
(Grade 6) Reference:		To come up with their findings, researchers **studied** national figures on injuries. The figures were provided by the Consumer Product Safety Commission, an agency that studies product safety. The study covers the early years of commercial zip lines. There are now more than 200 of these nationwide.
Our Approach	Initial Sequence	The researchers a injury operated by the Consumer Product Safety Commission. Their study covers the early years of which now number more than 200 nationwide.
Iter 1:	*Reposition*	The ~~researchers~~ a ~~injury~~ by the Consumer Product Safety Commission. ~~Their~~ study covers the early years ~~of which now number~~ more than 200 nationwide.
	Insert Placeholder	The [plh] [plh] a by the Consumer Product Safety Commission. [plh] study covers the early years [plh] [plh] [plh] [plh] [plh] [plh] more than 200 nationwide.
	Fill − in tokens	The researchers studied a by the Consumer Product Safety Commission. The study covers the early years of commercial zip lines. It now more than 200 nationwide.
Iter 2:	*Reposition*	The researchers studied ~~a~~ by the Consumer Product Safety Commission. The study covers the early years of commercial zip lines. It now more than 200 nationwide.
	Insert Placeholder	The researchers studied [plh] by the Consumer Product Safety Commission. The study covers the early years of commercial zip lines. It now [plh] more than 200 nationwide.
	Fill − in tokens	The researchers studied database by the Consumer Product Safety Commission. The study covers the early years of commercial zip lines. It now has more than 200 nationwide.
	no further actions	**[Terminate]**
Final Output:		The researchers studied database by the Consumer Product Safety Commission. The study covers the early years of commercial zip lines. It now has more than 200 nationwide.

Figure 2: Our approach substitutes "analyzed" correctly as well as splits the source sentence into two simple sentences to generate a simplified output that matches the lexical complexity of the desired grade-level 6. The tokens identified as complex using the proposed method in the source are bold.

76.4% with the oracle on the development set, indicating that the initial sequences contain appropriate vocabulary. Table 3 shows that `our approach` partially closes the gap in performance with the oracle by using this modified source sequence as opposed to the original source sequence (`NAR`).

Figure 2 illustrates the refinement process that generates the simplified output. Reposition and insertion operations are used in consecutive steps to perform complex editing operations (e.g., sentence splitting and lexical substitution) , which requires that the model learns to perform these operations sequentially. Furthermore, `our approach` recovers the tokens that were incorrectly identified as complex and thus deleted in the initial sequence, highlighting the benefits of the flexible refinement process.

4.2 Human evaluation

We randomly sample 60 source sentences from the Newsela-Grade dataset, among sources that are simplified toward four distinct grade levels (∼240 examples). For each of these target grades, we obtain ratings of system outputs and reference from

Model	SARI ↑	%PCC ↑	%ARI-Acc ↑	MSE ↓	Iteration	%Unchanged
our approach	40.2 ±0.0	73.9 ±0.3	36.1 ±0.2	3.73 ±0.03	2.32 ±0.04	17.6
−PMI-based Initialization	39.2 ±0.1	73.1 ±0.9	34.5 ±0.4	4.01 ±0.06	2.23 ±0.06	19.2
−Finetune	37.6 ±0.4	63.7 ±0.5	28.0 ±0.1	5.12 ±0.01	1.14 ±0.00	20.6
−Src Initialization	37.2 ±0.2	72.2 ±0.3	34.4 ±0.4	3.59 ±0.09	2.13 ±0.07	34.1
−Joint	39.7 ±0.1	68.1 ±1.7	31.6 ±0.2	4.70 ±0.05	1.00 ±0.00	11.0
Single Iteration	40.3 ±0.2	72.6 ±0.1	35.7 ±0.5	3.99 ±0.01	1.00 ±0.00	15.0
Gold Source Grade	40.2 ±0.1	74.1 ±0.5	36.5 ±0.3	3.71 ±0.04	2.33 ±0.03	17.3

Table 4: Ablation analysis on model design choices for our approach on Newsela-Grade development set.

	Meaning	Grammar	Simplicity		
	Mean	Mean	Mean	Abs. Diff ↓	Adj. Acc ↑
Reference	2.763	**3.193**	5.325	-	-
AR	**2.803**	3.171	5.157	2.035	0.533
our approach	2.647	3.081	5.310	**1.895**	**0.575**

Table 5: Human Evaluation Results: our approach generates output that match the reference judgements better than the AR baseline.

five Amazon Mechanical Turk workers. Following prior annotation protocols (Jiang et al., 2020), we ask workers to rate outputs on three dimensions: a) is the output grammatical? [0-4] b) to what extent is the meaning expressed in the original sentence preserved in the output? [0-4] and c) how simplified is the output with respect to the original source sentence? [0-10]. Different from prior work, we use a 10-point scale for evaluating simplicity to map the rating resolution to the gold grade differences. The detailed instructions provided to the workers are in the Appendix B.

We compute the absolute difference ("AbsDiff") in the simplicity ratings between the reference and the system output by the same annotator, and aggregate over all examples and all ratings. Table 5 shows that our outputs are closer to the reference according to the simplicity judgements than the AR system outputs. The "Mean" ratings indicate that the two models make different tradeoffs: where the AR model under-simplifies the source sentence and preserves the meaning, our approach almost matches the mean simplicity of the reference at the cost of lower meaning preservation. Our outputs are also less grammatical than those of the AR model and the references, probably due to the independence assumptions made by the non-autoregressive model. The Adjacency Accuracy, representing the percentage of system outputs within a difference of one rating with the reference, is also higher for our approach relative to the AR model.

4.3 Ablation Experiments

Table 4 summarizes the impact of the design choices described in Section 2: Removing lexical information (`-PMI-based Initialization`) hurts both SARI and the grade specific metrics. Further, using the baseline EDITOR model that is trained only to generate, without `fine-tuning` for refinement, significantly hurts the performance across the board. In that setting, EDITOR never learns to delete tokens from the source, but only learns to delete tokens inserted by the model. Using EDITOR to generate the output from scratch instead (`-Src Initialization`) recovers the performance on SARI and grade specific metrics but fails to match the performance of our approach. This shows that fine-tuning for refinement and providing initial sequences informed by lexical complexity are both key to the performance of the EDITOR for Controllable TS.

We also compare our approach with the variant of the model that is trained to perform reposition independent of the insertion operation (`-Joint`), similar to Mallinson et al. (2020). Even though this variant is able to match SARI, the difference in grade-specific metrics is significant, showing the benefits of joint training of the insertion and reposition components.

Iterative refinement helps match the target grade better than single step refinement as suggested by ARI Accuracy, MSE and PCC. Figure 3 shows the number of iterations of refinement performed by our approach as the function of desired target grade level: simplifying to lower grade levels (2 or 3) requires on average 1 additional refinement step

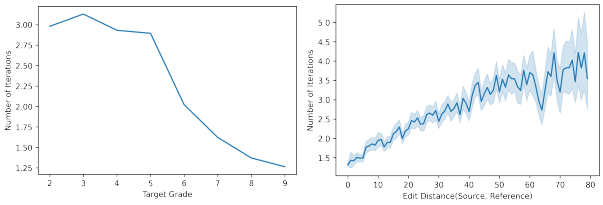


Figure 3: Our approach requires more number of iterations when simplifying to a lower grade level. The number of iterations performed by the model monotonically increases with the edit distance between source and reference.

than simplifiying to grade 8 or 9. This suggests that the iterative process helps simplification when the gap between source and target grades is wider.

Method	Precision	Recall
Target Only	76.4	80.2
Source (ARI) Only	**76.6**	78.2
Source (ARI) + Target	76.4	**91.3**

Table 6: Using both source and target grade to filter complex words yields maximum overlap with the set of tokens that are preserved from the source in the reference on the Newsela-Grade development set.

Finally, we verify that using ARI to estimate the complexity of the source is effective. Replacing the ARI predictions with the gold-standard grade-level improves the grade-specific metrics only marginally. Table 6 further shows the advantage of combining source and target grade information when identifying complex tokens (Equation 2) over using source or target grade only.

5 Analysis

Per Grade Analysis How does our model compare against the AR baselines for each target grade-levels? Figure 4a and 4b show the SARI and Adjacency Accuracy bucketed by target grade level. We observe that our approach achieves comparable or higher accuracy than the AR baselines for all grades except 2 and 3. Further analysis suggests that this is due to samples where the source grade level is 12, and where our approach deletes words too aggressively to simplify for the large grade gap (Figure 4c and 4d).

Model Edit operations We compare the number of edit operations performed by our model and the oracle Levenshtein Edit Distance (Section 2) when simplifying to different target grade levels. Figure 5 shows that the number of operations performed by our approach to generate its output track the number of oracle Levenshtein edits overall. The main differences are that our approach performs more than twice as many repositions than the oracle (5c) for grades 4 and above which suggest that the sequence of operations performed is suboptimal. Furthermore our approach overdeletes words for target grade levels lower than 4 (5b), and performs fewer insertions than the oracle (5a). We turn to manual analysis to shed more light on these results.

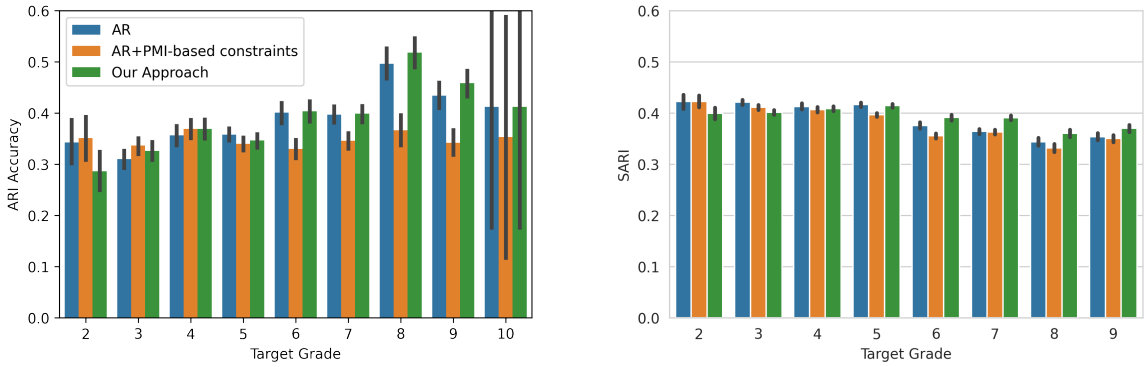
Simplification Operations Table 7 reports a manual annotation of the simplification operations observed for 50 randomly sampled segments, using an operation taxonomy from prior work (Xu et al., 2015; Jiang et al., 2020). Our approach performs content deletion in 7.5% more sentences than needed to generate the references. At the same time, it performs fewer insertions – in particular, our approach is unable to generate the elaborations and explanations found in the Newsela references (Srikanth and Li, 2020). This would require knowledge-based reasoning, which is beyond the capacity of the current model. However, our approach can model sentence splitting and substitution, which often require a sequence of insertion/deletion/reposition operations to be performed sequentially.

Type	% reference	% output
Lexical Substitution	25.0	17.5
Deletion	25.0	32.5
Reordering/Paraphrasing	35.0	20.0
Splitting	27.5	15.0
Content Elaboration	10.0	0.0
Unchanged	22.5	37.5

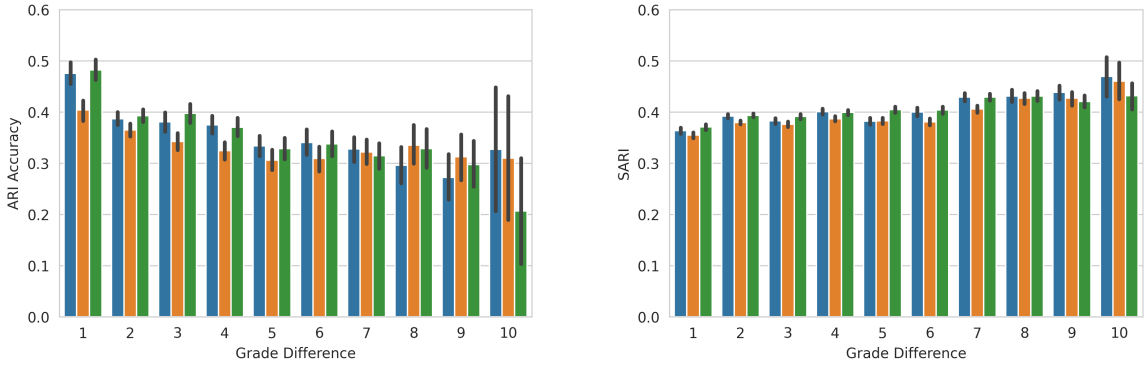
Table 7: Simplification Operations observed in the reference and output by our approach in 50 randomly sampled examples from the Newsela-Grade dataset.

6 Related Work

AR Models for TS Generic TS is often framed as machine translation where an autoregressive sequence-to-sequence model learns to model simplification operations implicitly from pairs of complex-simple training samples (Specia, 2010; Nisioi et al., 2017; Zhang and Lapata, 2017; Wubben et al., 2012; Scarton and Specia, 2018; Nishihara et al., 2019; Martin et al., 2020; Jiang

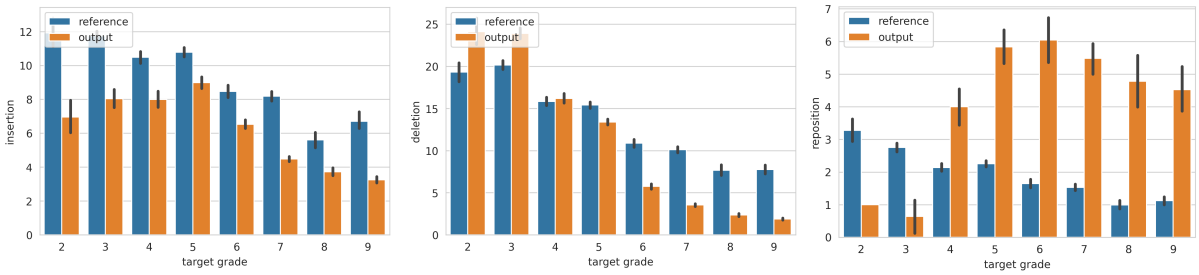


(a) ARI Accuracy (b) SARI



(c) ARI Accuracy (d) SARI

Figure 4: Analysis of automatic metrics for different target grade levels on the Newsela-Grade development set: our approach achieves higher or comparable SARI and ARI scores compared to the AR baselines for all grade levels except 2 or 3.



(a) Insertion (b) Deletion (c) Reposition

Figure 5: Edit operations accumulated over iterations for different target grade levels relative to the reference.

et al., 2020). There have been efforts at controlling a different aspect of the simplified output, such as controlling for a specific grade-level (Scarton and Specia, 2018; Nishihara et al., 2019) or employing lexical or syntactic constraints (Mallinson and Lapata, 2019; Martin et al., 2020), where the complexity of a word is either determined by its frequency or by manually tagging the tokens at inference time. We instead use the association of a word with the grade-level to define lexical constraints automatically. Furthermore, these models lack interpretability in terms of the type of operations performed, and need to generate the entire output sequence from scratch thus potentially wasting capacity in learning copying operations.

Edit-based Generic TS Recent work incorporates edit operations into neural text simplifications more directly. These approaches rely on custom multi-step architectures. They first learn to tag the source token representing the type of edit operations to be performed, and then use a secondary model for in-filling new tokens or executing the edit operation. The tagging and editing model are either

trained independently (Alva-Manchego et al., 2017; Malmi et al., 2019; Kumar et al., 2020; Mallinson et al., 2020) or jointly (Dong et al., 2019). By contrast, we use a single model trained end-to-end to generate sequences of edit operations to transform the entire source sequence.

Lexical Complexity for TS Nishihara et al. (2019) introduced a training loss for Controllable TS that weights words that frequently appear in the sentences of a specific grade-level. By contrast, we use lexical complexity information to define the initial sequence for refinement, which does not require any change to the model architecture nor to the training process. For Generic TS, Kajiwara (2019) used complex words as negative constrained for decoding with an autoregressive model. By contrast our approach provides more flexibility to the model which results in better outputs in practice.

Non-autoregressive Seq2Seq Models They have primarily been used to speed up Machine Translation by allowing parallel edit operations on the output sequence (Lee et al., 2018; Gu et al., 2018; Ghazvininejad et al., 2019; Stern et al., 2019; Chan et al., 2020; Xu and Carpuat, 2020). Refinement approaches have been used to incorporate terminology constraints in machine translation, including as hard (Susanto et al., 2020) and soft constraints (Xu and Carpuat, 2020). They have also shown promise for Automatic Post Editing (APE) (Gu et al., 2019; Wan et al., 2020) , and grammatical error correction (Awasthi et al., 2019). In this work, we show that they are a good fit to incorporate lexical complexity information for Controllable TS.

7 Conclusion

We introduced an approach that repurposes a non-autoregressive sequence-to-sequence model to incorporate lexical complexity signals in Controllable TS. An extensive empirical study showed that our approach generates simplified outputs that better match the desired target-grade complexity than AR models. Analysis revealed promising directions for future work, such as improving grammaticality while encouraging tighter control on complexity by better aligning the model's atomic edit operations with more complex simplification operations.

Acknowledgments

We thank Eleftheria Briakou, Pranav Goel, Aquia Richburg, Alexander Hoyle, Suraj Nair, Susmija Jabbireddy, the anonymous reviewers and the members of the CLIP lab at UMD for their helpful and constructive comments.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1549–1564.

David Allen. 2009. A study of the role of relative clauses in the simplification of news texts for learners of english. *System*, 37(4):585–599.

Oliver Alonzo, Matthew Seita, Abraham Glasser, and Matt Huenerfauth. 2020. Automatic text simplification tools for deaf and hard of hearing adults: Benefits of lexical simplification and providing users with autonomy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA. Association for Computing Machinery.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4251–4261.

William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pages 1403–1413. PMLR.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6114–6123.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11179–11189. Curran Associates, Inc.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the third workshop on innovative use of NLP for building educational applications*, pages 71–79. Association for Computational Linguistics.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2020. Controllable text simplification with explicit paraphrasing. *arXiv preprint arXiv:2010.11004*.

Jonathan Mallinson and Mirella Lapata. 2019. Controllable sentence simplification: Employing syntactic and lexical constraints. *arXiv preprint arXiv:1910.04387*.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5057–5068.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 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, Vancouver, Canada. Association for Computational Linguistics.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324.

Carolina Scarton and Lucia Specia. 2018. Learning Simplifications for Specific Target Audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 712–718.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, CINCINNATI UNIV OH.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language*, pages 30–39. Springer.

Neha Srikanth and Junyi Jessy Li. 2020. Elaborative simplification: Content addition and explanation generation in text simplification. *arXiv preprint arXiv:2010.10035*.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pages 5976–5985. PMLR.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

David Wan, Chris Kedzie, Faisal Ladhak, Marine Carpuat, and Kathleen McKeown. 2020. Incorporating terminology constraints in automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1193–1204.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 29–36.

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

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, 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*, 4:401–415.

Weijia Xu and Marine Carpuat. 2020. Editor: an edit-based transformer with repositioning for neural machine translation with soft lexical constraints. *arXiv preprint arXiv:2011.06868*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594.

A Appendix

A.1 Dataset Statistics

Table 8 provides the statistics of grade pair distribution in the Newsela-Grade dataset.

A.2 Implementation Details

We train all our models on two GeForce GTX 1080Ti GPUs. The average training time for a single seed of AR model is ∼8-9 hrs and for the EDITOR model is ∼20-22 hrs. Fine-tuning EDITOR takes additional 4-5 hrs.

B Human Annotation

Quality Control We set the location restriction to the United States to control for the quality of annotations. The correlation between the target grade levels and the simplicity ratings of the reference text is 0.582, which suggest that workers do rank simpler output higher than a relatively complex reference of the same source sentence.

Compensation We compensate the Amazon Mechanical Turk workers at a rate of $\$0.03$ per HIT.

Instructions We provide the following instructions to the Amazon Mechanical Turk workers to evaluate generated simplified sentences.

Meaning You are given one sentence and 3 rewrites of the same sentence. Carefully read the instructions provided and then use the sliders to indicate the extent to which the meaning expressed in the original sentence is preserved in the rewrites (Agirre et al., 2016).

Score	Category
4	they convey the same key idea
3	they convey the same key idea but differ in some unimportant details
2	they share some ideas but differ in important details
1	they convey different ideas on the same topic
0	Completely different from the first sentence

Grammar You are given three sentences. Carefully read the instructions provided and then use the sliders to indicate the extent to which each of the sentence is grammatical (Heilman et al., 2014).

Score	Category
4	Perfect: The sentence is native-sounding.
3	Comprehensible: The sentence may contain one or more minor grammatical errors
2	Somewhat Comprehensible: The sentence may contain one or more serious grammatical errors,
1	Incomprehensible: The sentence contains so many errors that it would be difficult to correct
0	Other/Incomplete This sentence is incomplete

Simplicity You are given one sentence and 3 rewrites of the same sentence. Carefully read the instructions provided and use the sliders to indicate how simple is each of the rewrite as compared to the original sentence (0: not simplified at all, 10: most simplified). We provide the following examples for your reference.

Src / Tgt	2	3	4	5	6	7	8	9	10
3	2488	0	0	0	0	0	0	0	0
4	466	7737	0	0	0	0	0	0	0
5	2080	18143	22888	0	0	0	0	0	0
6	1742	6952	20041	20212	0	0	0	0	0
7	545	7857	13556	31297	10315	0	0	0	0
8	557	3277	12557	16301	21457	11241	0	0	0
9	106	4338	4714	18143	4384	28690	2016	0	0
10	6	33	218	306	367	277	386	134	0
11	0	0	15	19	11	16	28	0	0
12	1039	6320	17703	32361	27144	39143	28545	29261	82

Table 8: Number of text segments per grade level pair in the Newsela-Grade dataset.

Original sentence: Craig and April Likhite drove to Chicago from Evanston with their 10-year-old son, Cade, because they wanted to see history made with other fans as close to Wrigley Field as possible.	
Rewrites:	**Simplicity:**
1. Craig and April Likhite drove to Chicago. They wanted to see history made with other fans as close to Wrigley Field as possible. *Explanation:* long and complex sentence has been split into two simple sentences, complex words are dropped	7
2. Craig and April Likhite to Chicago with their son Cade. *Explanation:* drastic content deletion	8
3. Craig and April Likhite drove to Chicago from Evanston with their 10-year-old son, Cade. They wanted to see history made with other fans as close to Wrigley Field as possible. *Explanation:* long sentence is split into two simple sentences	4
4. Craig and April Likhite drove to Chicago because they wanted to see history made with other fans as close to Wrigley Field as possible. *Explanation:* paraphrasing and deletion	6

Table 9: Example illustrating ratings for simplified rewrites of an originally complex sentence.