Target-Level Sentence Simplification as Controlled Paraphrasing

Anonymous ACL submission

Abstract

Automatic text simplification aims to reduce the linguistic complexity of a text in order to make it easier to understand and more accessible. However, simplified texts are consumed by a diverse array of target audiences and what may be appropriately simplified for one group of readers may differ considerably for another. In this work we investigate a novel formulation of sentence simplification as paraphrasing with controlled decoding, which aims to alleviate the major burden of relying on large amounts of in-domain parallel training data, while at the same time allowing for modular and adaptive simplification. According to a range of automatic metrics, our approach performs competitively against baselines that prove more difficult to adapt to the needs of different target audiences or require complex-simple parallel data.

1 Introduction

Sentence simplification (SS) aims to reduce the linguistic complexity of a sentence while still preserving its meaning in order to make a text easier to understand and to make texts more accessible to a wider array of potential readers (Bingel and Søgaard, 2016; Sikka and Mago, 2020). These readers may include children and adults with low literacy levels, cognitive impairments, or a lack of specialist knowledge in certain topics, as well as non-native language learners and even downstream natural language applications (Stajner, 2021; Saggion, 2017). However, the notion of exactly what constitutes simplified text is highly subjective and may differ considerably between different readers. Thus it is important to accommodate the needs of specific target audiences.

SS has been spurred on by performance gains in neural sequence-to-sequence (seq2seq) language generation methods that improve on earlier rule-based approaches (Wubben et al., 2012; Zhang and Lapata, 2017). However, fully supervised seq2seq approaches require a large amount of parallel training data that is both high in quality and diverse in order to derive robust and generalisable models (Koehn and Knowles, 2017). This poses a significant challenge for text simplification across the board as suitable training data is often scarce.

For these reasons, much work has focused on reducing the dependence on sentence-level parallel training data either by focusing on lexical simplification (Glavaš and Štajner, 2015; Kriz et al., 2018), structural simplification (Niklaus et al., 2019; Garain et al., 2019; Narayan et al., 2017; Gao et al., 2021), or both through text editing (Omelianchuk et al., 2021; Dong et al., 2019). Others have highlighted the commonality between SS and paraphrasing and aimed to exploit this relationship to bootstrap seq2seq-based simplification (Martin et al., 2020; Maddela et al., 2021).

We follow this line of work and investigate an alternative framing of SS as the task of controlled paraphrasing. We train a large-scale paraphrase model capable of producing high-quality and diverse paraphrases and combine it with FUDGE (Yang and Klein, 2021) for controlled decoding in order to steer the paraphrase generation towards a specific target-level for text simplification. Our experiments show that this proves to be an effective approach for generating simplified sentences for different target audiences without requiring any parallel sentence data.

2 Background & Motivation

As pointed out by Stajner (2021), text simplification systems should be developed to support a variety of target populations and would thus benefit from a modular approach that allows for easy customisation and adaption. Meanwhile, a major hurdle for popular neural-based approaches is the collection of appropriate sentence-aligned parallel training data, which inhibits the development of robust systems (Laban et al., 2021). Recently, however, large general-purpose text generation models have demonstrated impressive performance in both conditional and unconditional generation tasks (Radford et al., 2019; Lewis et al., 2020). Along with this, there has been considerable work done exploring ways to better control the outputs of large generation models in order to achieve certain communicative goals (Dathathri et al., 2020; Krause et al., 2021; Liu et al., 2021; Yang and Klein, 2021; Pascual et al., 2021). We see a clear link between these recent developments and the challenges associated with SS and set out to investigate a modular approach suitable for simplifying content for different target audiences without requiring any complex-simple parallel data for model training.

3 Method

Given a complex source sentence, the goal is to translate it into a simplified target sequence[1] that preserves its meaning. Following the seq2seq framework, we generate a target sequence X conditioned on the source sentence I.[2] The probability of the sequence $P(X)$ is computed as the probability of the ith token conditioned on the input source sequence I all previously generated tokens,

$$P(X) = \prod_{i=1}^{n} P(x_i|I, x_{1:i-1}). \tag{1}$$

In order to ensure that the generated target sequence is appropriately simplified for a specific target-level, we employ the controlled decoding method FUDGE (FUture Discriminators for GEneration) (Yang and Klein, 2021), which has been shown to be effective for poetry couplet generation, topic-controlled generation and controlling formality in machine translation. FUDGE introduces a lightweight classifier $\mathcal{B}$ into the generation process of any autoregressive generation model $\mathcal{G}$, modifying Equation 1 through a Bayesian factorisation of the target sequence:

$$P(x_i|I, x_{1:i-1}, a) \propto P(a|x_{1:i})P(x_i|I, x_{1:i-1}), \tag{2}$$

where a is the target attribute being controlled for. This factorisation is especially appealing given today's popular pre-trained generation models, since, as long as $\mathcal{B}$ and $\mathcal{G}$ share the same tokenisation, it only requires access to $\mathcal{G}$'s output logits at each timestep, making the system highly modular and adaptable. For further details on FUDGE, we refer the reader to Yang and Klein (2021).

[1]Since an appropriate simplified formulation may consist of multiple shorter sentences we refer to it as a sequence.

[2]For consistency, we borrow the notation used in Yang and Klein (2021).

	# articles	# manually aligned sentences			
		Simp-1	Simp-2	Simp-3	Simp-4
train	1,862	-	-	-	-
train	35	1,341	1,245	1,042	841
test	10	365	353	309	256
valid	5	180	163	134	87

Table 1: Newsela English corpus articles and their *manually* aligned sentences from Jiang et al. (2020) for Simp-0 to Simp-l.

4 Experimental Setup

4.1 Data

We conduct our experiments on the Newsela corpus of simplified news articles[3]. In its current form, the corpus contains 1,912 English news articles that have been professionally re-written according to readability guidelines for children at multiple grade levels (Xu et al., 2015). Article versions range from Simp-0 to Simp-4, with the former referring to the original unsimplified article, suitable for upper secondary school grades, and the latter indicating the simplest versions, suitable for lower primary school grades.

While Newsela provides complex-simple alignments at the document level, it must be emphasised that this alignment is not a requirement for our SS approach with FUDGE. That said, we reason that it is beneficial as it ensures that examples used to train the attribute classifiers (henceforth FUDGEs) cover the same domain. Consequently, each FUDGE must learn to distinguish between complex and simple text based on relevant characteristics such as the vocabulary and grammatical structures used rather than relying on differences in topical content, which could be misleading (Kumar et al., 2019).

Evaluation data For automatic evaluation purposes, however, alignments on the sentence level are a must. To this end, we make use of the manually aligned test and validation splits provided by Jiang et al. (2020). Setting aside all sentence pairs from these splits ensures that no unwanted data leakage occurs. An overview of the corpus and manually aligned sentence pairs is provided in Table 1.

[3]https://newsela.com/data/

4.2 FUDGE for Sentence Simplification

To apply FUDGE on target-level SS, we train a classifier for *each* target level, i.e., $\mathcal{B}_{Simp-l}$, and combine them with the same underlying generator model $\mathcal{G}$. Following Yang and Klein (2021), each FUDGE is trained as a binary predictor on labelled subsequences of complex (Simp-0) and simple (Simp-l) texts. Since SS often involves breaking down a long complex sentence into smaller atomic sentences (Honeyfield, 1977), we make use of the paragraph structure available in the Newsela corpus and train each FUDGE to predict labels on subsequences pertaining to consecutive sentences. This ensures that the FUDGE's predictions do not unduly bias the generation of the end of sentence symbol '</S>' after producing sentence-final punctuation.

As the underlying generator, $\mathcal{G}$, we fine-tune BART-large on 1.4 million paraphrase sentence pairs mined from the web.[4] To ensure a fair comparison to previous state-of-the-art, we use the exact same training data as Martin et al. (2021) and aim to keep training hyperparameters as consistent as possible (detailed information on the training settings and the paraphrase corpus is given in Appendix B). Combining the predictions from $\mathcal{G}$ and $\mathcal{B}$ makes use of a single weight parameter λ. For our experiments, we derive suitable values for each target-level by sweeping over possible whole number values in the range [0,10] and select the best according to SARI on the held-out validation set (see Appendix C).

4.3 Baselines

We compare our approach to two recently proposed techniques for controlled SS.

MUSS Martin et al. (2021) leverage large-scale paraphrase data to fine-tune BART-large in combination with the ACCESS control method for simplification (Martin et al., 2020). ACCESS relies on four special tokens which are prepended to each source sequence indicating length, N-gram similarity, lexical and syntactic complexity ratios between the source and target sequences. At inference time, these special tokens act as control knobs for simplification. Following Martin et al. (2021), we derive the best special token values through a parameter search on the same held-out validation set as used to set λ for FUDGE models (see Appendix B.4).

SUPER Following Scarton and Specia (2018) and Spring et al. (2021), we also train a level-aware supervised baseline with a special token indicating the target level (e.g., `<L3>` = Simp-3) prepended to each source sentence. For a fair comparison, we initialise this model from the same BART-large checkpoint as the other two models and fine-tune on the manually aligned sentence pairs for all Newsela levels simultaneously. This amounts to a low resource setting with a total of 4,469 training instances.

PARA In addition, we also compare to a straight-forward paraphrase generated by our underlying generation model $\mathcal{G}$ with no control.

4.4 Evaluation Metrics

Reliably evaluating SS is an open challenge (Alva-Manchego et al., 2021). However, a range of both reference-based and reference-less automatic metrics have been proposed (Martin et al., 2018). We make use of the open-source EASSE package (Alva-Manchego et al., 2019), which implements relevant metrics such as SARI, BERTScore, Flesch-Kincaid Grade Level (FKGL) and a host of quality evaluation measures for more fine-grained analysis of the simplifications generated (see Appendix A for more details).

5 Results & Discussion

Table 2 presents the results of our experiments on the Newsela corpus. According to SARI, our primary metric, SS with FUDGE outperforms both MUSS and supervised baselines for all simplification levels except for Simp-4, where the supervised method performs surprisingly well. This result is consistent with the findings from Spring et al. (2021), where this simple labelling approach proved most effective for simplifying ordinary German to A1-level German, despite it being the target level with the least amount of parallel data in both studies. At lower simplification levels, this model has a strong tendency to copy the inputs.

MUSS produces suitable simplifications according to FKGL, yet this model also tends to summarise the input, as shown by the lower compression ratio scores and a higher proportion of deleted tokens. This information loss causes model outputs to diverge from the ground truth reference sequences and appears to be appropriately penalised by BERTScore. Meanwhile, FUDGE achieves

[4] In theory, it could be possible to avoid fine-tuning the generator all together, but initial experiments showed that the probability distribution of the off-the-shelf BART model is far too peaked for FUDGE's predictions to have any effect.

Method	SARI	BERTScore	FKGL	Comp. ratio	Sent. splits	Lev. sim.	Copies	Add prop.	Del prop.
Target Level: Simp-1			7.97	1.01	1.19	0.90	0.44	0.10	0.10
PARA	36.61	81.68	9.15	0.97	1.02	**0.89**	**0.18**	**0.08**	**0.11**
MUSS	35.69	75.95	**7.75**	0.81	1.00	0.84	0.01	0.07	0.24
SUPER	32.49	88.19	9.36	**0.99**	**1.04**	0.99	0.89	0.01	0.01
$\mathcal{B}_{Simp-1}$	36.10	80.45	8.81	0.94	1.01	0.88	0.13	0.07	0.13
Target Level: Simp-2			6.41	0.98	1.42	0.82	0.23	0.17	0.20
PARA	35.01	73.53	9.12	**0.97**	1.02	0.89	**0.18**	0.08	0.11
MUSS	36.57	65.91	**7.27**	0.78	1.03	0.75	0.00	**0.15**	0.35
SUPER	31.12	78.22	8.88	0.99	1.10	0.98	0.80	0.02	0.03
$\mathcal{B}_{Simp-2}$	38.32	70.75	7.42	0.96	**1.25**	**0.84**	0.08	0.12	**0.17**
Target Level: Simp-3			4.91	0.92	1.55	0.73	0.13	0.24	0.31
PARA	30.87	65.06	9.09	0.98	1.01	0.89	**0.18**	0.08	0.11
MUSS	38.05	56.03	**5.19**	0.62	1.01	**0.68**	0.00	0.12	0.45
SUPER	37.89	66.60	6.65	**0.93**	1.34	0.90	0.48	0.06	0.13
$\mathcal{B}_{Simp-3}$	39.56	61.46	6.44	1.00	**1.45**	0.81	0.02	**0.20**	**0.20**
Target Level: Simp-4			3.40	0.85	1.79	0.65	0.09	0.30	0.43
PARA	25.61	56.21	9.41	0.98	1.01	0.89	0.18	0.08	0.11
MUSS	39.63	51.73	5.61	0.65	1.04	**0.68**	0.00	0.13	**0.44**
SUPER	43.22	55.00	5.09	**0.78**	**1.45**	0.74	0.24	0.12	0.32
$\mathcal{B}_{Simp-4}$	37.03	49.60	**4.60**	1.02	2.14	0.76	**0.00**	**0.28**	0.28

Table 2: Target-level results on the Newsela corpus. For reference-based metrics (SARI, BERTScore), where higher values are better, we highlight systems according to their performance. For FKGL and reference-less quality evaluation metrics we embolden the systems which perform closest to the level-specific references (provided in the intermediary rows).

lower BERTScores than both the supervised and paraphrase baselines, where it appears to reward outputs that make fewer modifications to the source sentence, as indicated by the higher degree of copying. In contrast to the baselines, FUDGE demonstrates a higher rate of sentence splitting and additions, which is of particular advantage for SS for certain target audiences. That said, manual inspection of the model outputs shows that not all additions and sentence splits are warranted and that these could be degenerative artefacts, such as unnecessary repetitions or hallucinations (see tables in Appendix F for examples). Comparing FUDGE against the paraphrase baseline without control clearly shows the strong positive influence of FUDGE for SS.

Since simplifying with FUDGE is performed actively during decoding and decisions are informed by the currently generated prefix $x_{1:i-1}$, this approach is not *guaranteed* to transform the input text. This is an important consideration for SS as oftentimes not all parts of a sentence need to be simplified (Garbacea et al., 2021). Thus, given a well-trained model, FUDGE performs simplification operations only when appropriate.

SS with FUDGE also makes use of a single hyperparameter λ which controls the contribution from $\mathcal{B}$. In contrast, MUSS requires setting an appropriate continuous value for each of the four control tokens to attain a suitable simplification. These are not only difficult to determine for each target level (see Appendix D), but the way in which these tokens interact with each other is also unclear (Martin et al., 2020).

6 Conclusion & Future Work

We have explored a modular and adaptable approach to SS by reframing it as a controlled paraphrasing task. We used FUDGE (Yang and Klein, 2021) to steer the generation of paraphrastic target sequences toward different target levels. This modular approach to SS is comparable to state-of-the-art methods according to automatic metrics. In future work we aim to conduct a more detailed human evaluation in order to better understand the qualitative differences between these approaches, as well as applying our method to larger textual units beyond sentences.

References

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. 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): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un)suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Joachim Bingel and Anders Søgaard. 2016. Text simplification as tree labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 337–343, Berlin, Germany. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *arXiv:1912.02164 [cs]*.

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.

Yanjun Gao, Ting-Hao Huang, and Rebecca J. Passonneau. 2021. ABCD: A graph framework to convert complex sentences to a covering set of simple sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3919–3931, Online. Association for Computational Linguistics.

Avishek Garain, Arpan Basu, Rudrajit Dawn, and Sudip Kumar Naskar. 2019. Sentence Simplification using Syntactic Parse trees. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, pages 672–676, Mathura, India. IEEE.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable Prediction of Text Complexity: The Missing Preliminaries for Text Simplification. *arXiv:2007.15823 [cs]*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China. Association for Computational Linguistics.

John Honeyfield. 1977. Simplification. *TESOL Quarterly*, 11(4):431–440.

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*, pages 7943–7960, Online. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification using paraphrases and context-based lexical substitution. 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 207–217, New Orleans, Louisiana. Association for Computational Linguistics.

Sachin Kumar, Shuly Wintner, Noah A. Smith, and Yulia Tsvetkov. 2019. Topics to avoid: Demoting latent confounds in text classification. 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 4153–4163, Hong Kong, China. Association for Computational Linguistics.

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

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

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553, Online. Association for Computational Linguistics.

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.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2021. MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases. *arXiv:2005.00352 [cs]*.

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

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. DisSim: A discourse-aware syntactic text simplification framework for English and German. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 504–507, Tokyo, Japan. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanskyi. 2021. Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Horacio Saggion. 2017. *Automatic Text Simplification*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

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)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Thomas Scialom, Louis Martin, Jacopo Staiano, Éric Villemonte de la Clergerie, and Benoît Sagot. 2021. Rethinking Automatic Evaluation in Sentence Simplification. *arXiv:2104.07560 [cs]*.

Punardeep Sikka and Vijay Mago. 2020. A Survey on Text Simplification. *arXiv:2008.08612 [cs]*.

Nicolas Spring, Annette Rios, and Sarah Ebling. 2021. Exploring German multi-level text simplification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1339–1349, Held Online. INCOMA Ltd.

Sanja Stajner. 2021. Automatic text simplification for social good: Progress and challenges. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652, Online. Association for Computational Linguistics.

Teerapaun Tanprasert and David Kauchak. 2021. Flesch-kincaid is not a text simplification evaluation metric. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14, Online. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.

A Evaluation Metrics for Sentence Simplification

Simplicity SARI is intended to measure simplicity by considering N-gram overlap between the source sentence, model output and one or more reference sentences. It rewards model outputs that involve edit operations such as deletions, additions and copies which correspond with the provided references.

Fluency and meaning preservation BERTScore uses BERT's contextualised representations to compute the similarity between tokens in the model output and one or more references. It has been shown to correlate better than BLEU for assessing meaning preservation and fluency in SS (Scialom et al., 2021).

Readability Flesch-Kincaid Grade Level (FKGL) is often used as a proxy for estimating text simplicity without a reference. Originally developed for grading technical materials for military personnel, it considers surface-level statistics such as word and sentence length to provide a single score. However, these scores should be interpreted carefully as it has recently been shown that this metric can be mislead by degenerate and disfluent outputs (Tanprasert and Kauchak, 2021).

Quality Evaluation Measures For a more fine-grained analysis of model outputs, we also report quality estimation measures which are computed between the source sentence and the model's output. These include the compression ratio, Levenshtein similarity, average number of sentence splits performed, exact copies between source and target, and the proportion of added and deleted N-grams.

B Settings used for Model Training and Inference

B.1 Resources

Model training and inference experiments were performed on NVIDIA GeForce GTX TITAN X GPUs (12GB).

B.2 Training Generation Models

For our underlying generator model $\mathcal{G}$ and the level-aware supervised baseline, we fine-tune BART-large using Hugging Face's Transformers library[5]

[5] https://github.com/huggingface/transformers

(Wolf et al., 2020). Training parameters used for $\mathcal{G}$ aim to replicate the settings used by Martin et al. (2021) who trained their models using Fairseq[6]. For the level-aware supervised baseline, we aim to replicate the settings used by Spring et al. (2021) who trained their models with Sockeye[7]. Note, in contrast to the paraphrase model, the effective batch size and maximum training steps for this model are considerably smaller to account for the differences in the size of the relevant training data (1.4M paraphrase sentence pairs vs. 4k aligned simplifications).

Paraphrase Model $\mathcal{G}$	
hyperparameter	value
max src length	1024
max tgt length	256
eff. batch size	64
learning rate	3e-05
weight decay	0.01
optim	adamw_hf
adam betas	0.9 - 0.999
adam epsilon	1e-8
lr scheduler	polynomial
warmup steps	500
label smoothing	0.1
max steps	20000
num beams for pred	4
optim metric	loss
Level-Aware Supervised Model	
hyperparameter	value
max src length	256
max tgt length	128
eff. batch size	16
learning rate	3e-05
weight decay	0.01
optim	adamw_hf
adam betas	0.9 - 0.999
adam epsilon	1e-8
lr scheduler	polynomial
warmup steps	500
label smoothing	0.1
max steps	5000
num beams for pred	4
optim metric	rouge1

Table 3: Hyperparameters for training generation models

B.3 Training FUDGE Classifiers

Our FUDGE classifiers $\mathcal{B}_{simp-l}$ are unidirectional three-layer LSTM-based RNNs with hidden layer dimensionality of 512. These settings differ slightly from the original implementation by Yang and Klein (2021), who learn slightly smaller classifiers for their tasks. The embedding matrix is constructed to cover the vocabulary of the underlying generator model and token embeddings are initialised using 300d pre-trained GloVe embeddings (`glove-wiki-gigaword-300`) (Pennington et al., 2014). For certain wordpieces and rare words that are OOV in GloVe, we initialise their embeddings randomly.

B.4 Inference

For all models except MUSS we run inference with beam search (k=5). A manual inspection of the model outputs revealed that our underlying paraphraser $\mathcal{G}$ showed a tendency to produce repetitions in the target sequence. To counter this, we set the repetition penalty equal to 1.2 when performing inference with $\mathcal{G}$. All other inference hyperparameters use the default values set in Hugging Face. For each source sentence in the test set, we generate the top five model hypotheses according to the model and select the first non-empty string as the final model output.

FUDGE has two hyperparameters which need to be set at inference time. The first is a weight λ that controls the strength of $\mathcal{B}$'s contribution, while the second aims to keep the cost associated with classifying all possible continuations at each decoding timestep down by limiting the computation to the top-k predictions at each step. Our experiments showed that λ is indeed useful for controlling the degree of simplification and finding a suitable λ is important for getting the best target-level simplifications (see Figure 1). Meanwhile using different pre-selection top-k values (e.g., [50, 200]) had almost no effect on the resulting generation sequence when using argmax decoding techniques such as beam search. Therefore, we follow the recommendation by Yang and Klein (2021), and fix the pre-selection top-k=200.

For MUSS, we kept inference settings the same as the default set by Martin et al. (2021). The only differences are the control token values used for performing inference on each of the Newsela simplification levels, which we derive via a parameter sweep over 50 items from the respective development set. Table 4 shows the relevant values used.

[6] https://github.com/huggingface/transformers
[7] https://github.com/awslabs/sockeye

	Comp. Ratio	Leven-shtein Sim.	Word Rank Ratio	Dep. Tree Depth Ratio
Simp-1	0.30	0.99	0.54	1.45
Simp-2	0.75	0.82	0.94	0.22
Simp-3	0.52	0.85	0.45	0.62
Simp-4	0.47	0.79	0.43	0.42

Table 4: Values used for target-level inference on the Newsela English corpus with MUSS

C Parameter Sweep for FUDGE

We search for the optimum λ value for each combination of Newsela simplification levels and each target-level FUDGE on 50 sentences from the manually aligned validation set (Jiang et al., 2020). Table 2 shows the resulting SARI scores. For our experiments, we selected the best scoring λs for each simplification level and its corresponding FUDGE (i.e., plots along the diagonal). For instances where more than one possible λ delivers good results, we select the lowest λ value > 0 (marked with a vertical dotted line).

It is clear from this figure that cross-matching target simplification levels with FUDGEs trained on a different target level would also yield good, and in some cases even better, results according to SARI (e.g., target-level Simp-2 with $\mathcal{B}_{Simp-3}$). This is likely due to it being easier for the classifier to correctly distinguish between the positive (simple) and negative (complex) classes when the stylistic differences between simplification levels are larger. Indeed, ROC-AUC scores for each target-level classifier on the respective test sets increase from 0.67 to 0.96 going from Simp-1 to Simp-4, indicating that FUDGEs trained on higher simplification levels are better at distinguishing between the classes.

D ACCESS Attributes on Newsela Corpus

Deciding on optimal attribute values for target-level simplification with ACCESS is non-trivial. We computed the ratio scores on source-target pairs from the manually aligned training split from Jiang et al. (2020) for all four simplification levels of the Newsela English corpus. Figure 2 shows that for most attributes, the largest density is on a value of 1.0, which would indicate no difference between the source and target. For many attribute values, the distributions are also relatively wide and flat indicating that there could be many potentially valid values, especially for the higher simplification levels (e.g., Simp-2 - Simp-4).

E Ablation Experiment

Unlike a fully-supervised seq2seq approach, FUDGE for SS does not require parallel complex-simple sentence pairs for training. Instead, SS with FUDGE relies on contrastive instances to train its target-level classifiers. Such data is significantly easier to collect from comparable, contrastive, or even 'monolingual' corpora, e.g., language learning materials (Vajjala and Lučić, 2018), information from government websites or news articles produced specifically for certain target groups which are available for in a variety of languages[8].

However, an open question remains as to how much data is required to train a suitable classifier. While this may depend heavily on the target-level simplified text both in topical and stylistic features, we examined this question for Newsela's Simp-4 target level. In contrast to the main experiments, here, we set FUDGE's $\lambda = 1.0$ (i.e., the minimum amount of influence). Figure 3 depicts the relationship between the amount of contrastive data used to train $\mathcal{B}_{Simp-l}$ and the resulting automatic metrics.

For metrics that consider simplification, a strong positive correlation can be seen, indicating that the amount of contrastive data helps considerably to get the best performance. However, even small amounts of contrastive data can already be effective in steering the generations towards the target attribute.

F Output Examples

The tables below provide randomly sampled examples of model outputs for each target-level in the Newsela English corpus. We colour parts of the simplified texts based on the edit operations applied to the source text. Blue indicates additions or explanations not in the source text. Green is used to highlight lexical and punctuation substitutions. Yellow shows operations on contractions (either creating or deconstructing). Pink indicates phrases that have been truncated or lexical deletions from the source text. Violet is used for larger paraphrastic segments or positionally shuffled phrases. Undesirable repetitions or hallucinations are italicised.

[8]For example, Ligetil from the Danish Broadcasting Corporation (https://www.dr.dk/ligetil/) and Japan's News Web Easy (https://www3.nhk.or.jp/news/easy/)

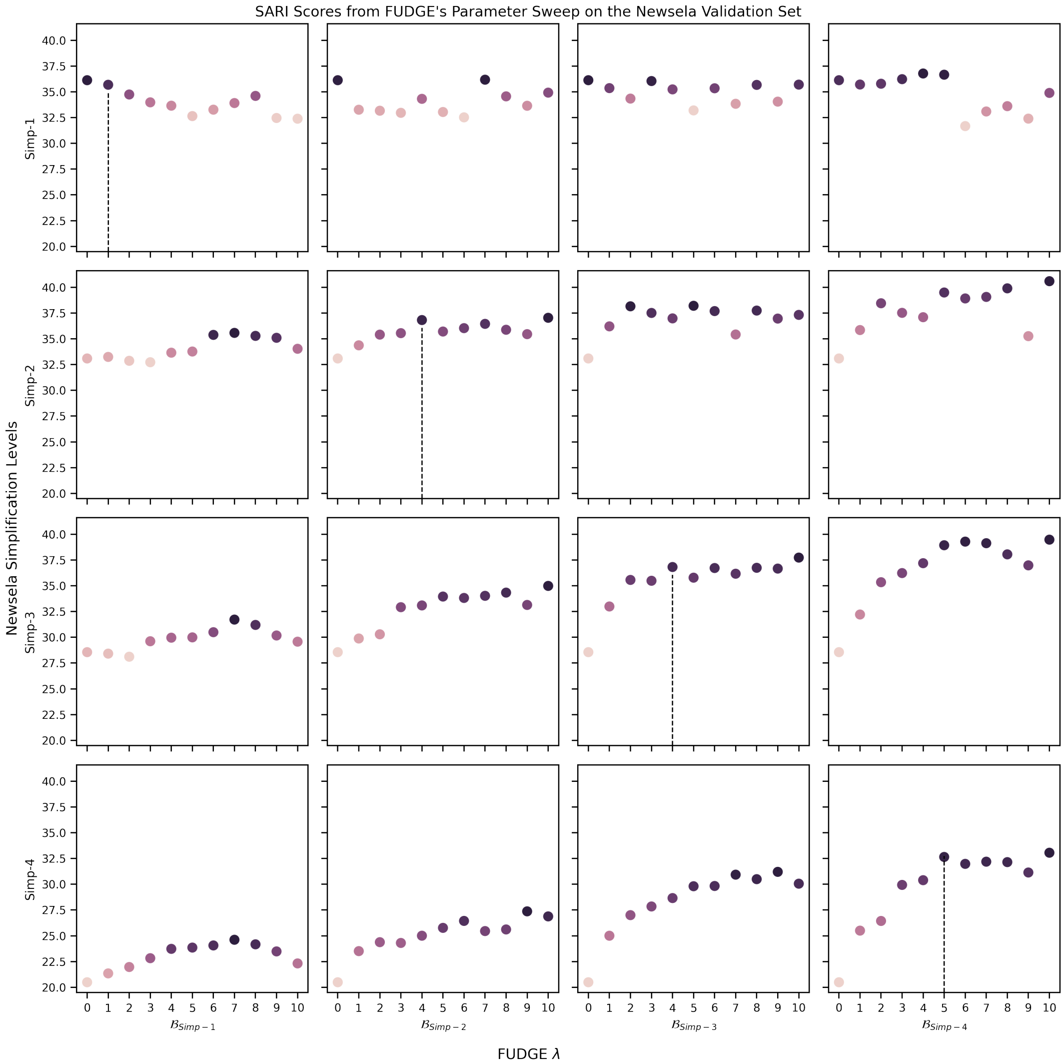

Figure 1: SARI scores from parameter sweep over different λ values for FUDGE at inference time.

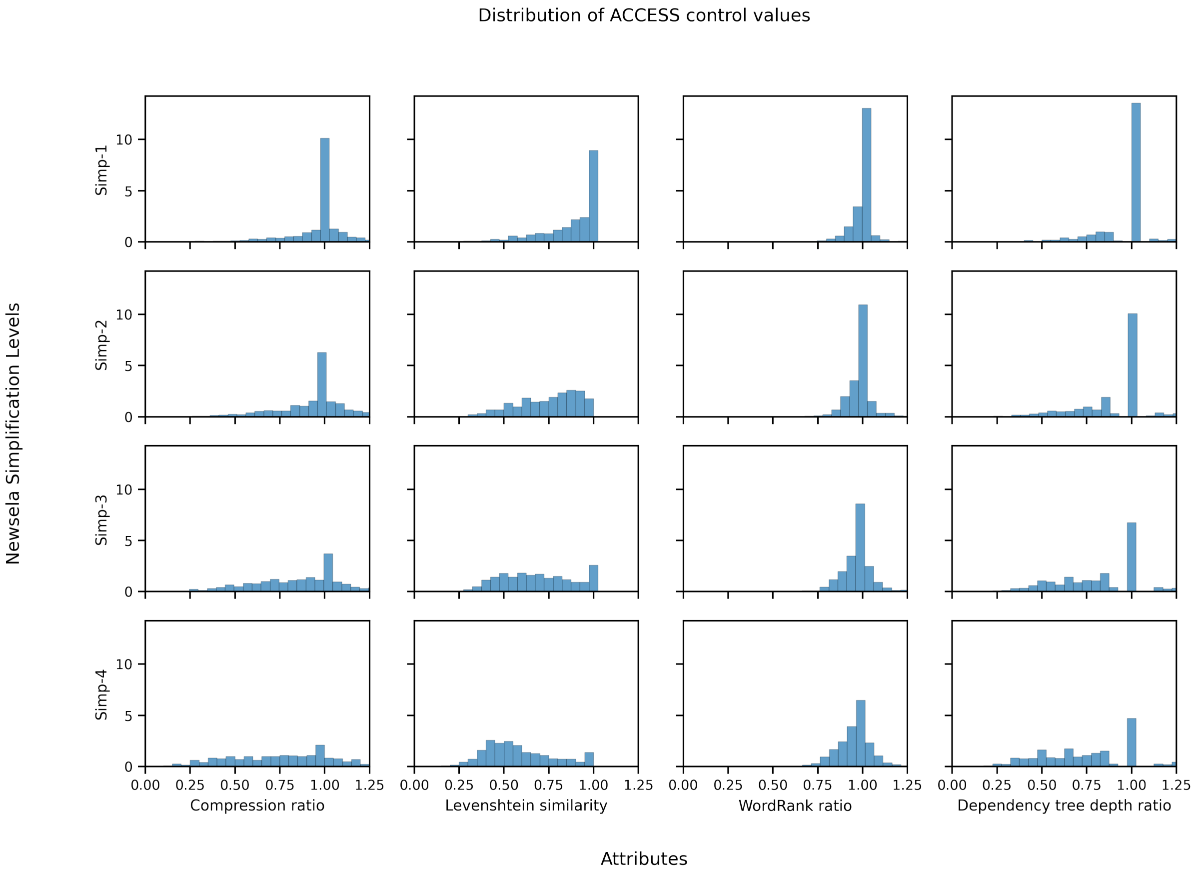

Figure 2: Density of attribute values for the four control tokens used in the ACCESS simplification method (Martin et al., 2020) and employed by MUSS (Martin et al., 2021).

Figure 3: Automatic metrics for simplification (implemented in EASSE (Alva-Manchego et al., 2019)) as a function of the number of contrastive complex-simple paragraphs from the Newsela English corpus used to train a FUDGE classifier $\mathcal{B}_{Simp-4}$

.

Target Level: Simp-1

Simp-0 (source)	They found that there was indeed a pattern in their motion — though it wasn't quite what they expected.
Simp-1 (reference)	They found that there was indeed a pattern in their motion — though it was not quite what they expected.
MUSS	They found that there was a pattern in their motion - but it wasn't what they expected.
SUPER	They found that there was indeed a pattern in their motion — though it wasn't quite what they expected.
$\mathcal{B}_{simp-1}$	They found that there was a pattern in their movement, though it wasn't what they expected.
Simp-0 (source)	The tribes helped members of the expedition through the winter, bringing them food and assisting with navigation.
Simp-1 (reference)	The tribes helped members of the expedition through the winter, bringing them food and assisting with navigation.
MUSS	The tribes helped members of the expedition by bringing them food and helping with navigation.
SUPER	The tribes helped members of the expedition through the winter, bringing them food and assisting with navigation.
$\mathcal{B}_{simp-1}$	The tribes helped members of the expedition through the winter, bringing food and helping with navigation.

Target Level: Simp-2

Simp-0 (source)	The feat should be possible based on Einstein's theory of general relativity, which describes gravity as the warping of spacetime by energy and matter.
Simp-2 (reference)	It should be possible based on Einstein's theory of general relativity. His famous theory describes gravity as the bending of space and time, which are one thing called "spacetime."
MUSS	Einstein's theory, called general relativity, describes gravity as the warping of spacetime by energy and matter.
SUPER	The feat should be possible based on Einstein's theory of general relativity, which describes gravity as the warping of spacetime by energy and matter.
$\mathcal{B}_{simp-2}$	The feat is based on Einstein's theory of general relativity. It describes gravity as the warping of spacetime by matter and energy.
Simp-0 (source)	Changes include moving the earliest date from which tribes must provide documentation of their existence into the 1930s and allowing tribes to reapply for federal recognition under certain circumstances, something previously prohibited.
Simp-2 (reference)	The revisions include establishing the 1930s as the earliest date from which tribes must provide documentation of their existence. Also, tribes may reapply for federal recognition under certain situations, something previously prohibited.
MUSS	Changes include moving the earliest date for tribes to provide documentation of existence into the 1930s, and allowing under certain circumstances tribes to reapply for federal recognition.
SUPER	Changes include moving the earliest date from which tribes must provide documentation of their existence into the 1930s and allowing tribes to reapply for federal recognition under certain circumstances, something previously prohibited.
$\mathcal{B}_{simp-2}$	Changes include moving the earliest date from which tribes must provide documentation of their existence. Tribes will also be allowed to reapply for federal recognition.

Target Level: Simp-3

Simp-0 (source)	Cafes, restaurants and bakeries around the nation are feeling the pain from recent outbreaks of bird flu, or avian influenza, expanding through the Midwest.
Simp-3 (reference)	The rising prices is caused by recent outbreaks of bird flu.
MUSS	Restaurants and bakeries around the nation are feeling the effects of the recent bird flu outbreak in the Midwest.
SUPER	Cafes, restaurants and bakeries around the nation are feeling the pain from recent outbreaks of bird flu. The disease is spreading through the Midwest.
$\mathcal{B}_{simp-3}$	Cafes, restaurants and bakeries around the nation are feeling the pain of bird flu outbreaks that have spread in the Midwest.
Simp-0 (source)	"That is, a time traveler who emerges from a Deutschian CTC enters a universe that has nothing to do with the one she exited in the future.
Simp-3 (reference)	Lloyd says a time traveler coming from Deutsch's CTC "enters a universe that has nothing to do with the one she exited."
MUSS	A Deutschian CTC traveler enters a universe that has nothing to do with the one she left.
SUPER	"That is, a time traveler who emerges from a Deutschian CTC enters a universe that has nothing to do with the one she exited in the future.
$\mathcal{B}_{simp-3}$	"That is, a time traveler who emerges from a Deutschian CTC enters a universe. This universe has nothing to do with the one she exited in the future. *The universe she enters into has nothing to do with the one she exited in the future.*

Target Level: Simp-4

Simp-0 (source)	If the particle flips the switch, the machine emits a particle—the particle—back into the CTC; if the switch isn't flipped, the machine emits nothing.
Simp-4 (reference)	If the particle flips the switch, the machine shoots a particle — the particle — back into the CTC. If the switch isn't flipped, the machine shoots out nothing.
MUSS	If the particle turns, the machine sends a particle back to the CTC. If it doesn't, it sends nothing.
SUPER	If the particle flips the switch, the machine emits a particle—the particle—back into the CTC. If the switch isn't flipped, the machine emits nothing.
$\mathcal{B}_{simp-4}$	If the switch is turned, the machine emits a particle-like stateBack into the CTC. If the switch is not turned, the machine emits nothing.
Simp-0 (source)	Deutsch's insight was to postulate self-consistency in the quantum realm, to insist that any particle entering one end of a CTC must emerge at the other end with identical properties.
Simp-4 (reference)	Deutsch assumes that tiny quantum particles are stable and fixed.
MUSS	In quantum theory, Deutsch insisted that any particle entering one end of a CTC must emerge at the other end with equal properties.
SUPER	Deutsch's idea was to show that any particle entering one end of a CTC must emerge at the other end *of a CTC must emerge at the other end* with identical properties.
$\mathcal{B}_{simp-4}$	Deutsch's idea was to postulate a very nature. He was claiming that any particle entering one end of a CTC must emerge at the other end with identical properties.