

Continuous Difficulty Control for Text Simplification

Yuki Fujiwara¹ Rina Miyata¹ Tomoyuki Kajiwara^{1,2} Yuki Arase³

¹Ehime University ²The University of Osaka ³Institute of Science Tokyo
{fujiwara@ai., miyata@ai., kajiwara@}cs.ehime-u.ac.jp, arase@c.titech.ac.jp

Abstract

Controlling text difficulty is crucial for adapting texts to individual language learners. Existing controllable text simplification methods condition generation on discrete proficiency-level tokens, treating these levels as independent categories. As a result, they fail to capture the ordinal structure of language proficiency and cannot provide fine-grained control beyond predefined levels. To address this limitation, we propose a *continuous difficulty control* method that adjusts simplification difficulty by editing the norm of a conditioning embedding derived from the input sentence semantics. The embedding is incorporated into the transformer decoder through attention-based conditioning, enabling continuous control over output difficulty while modeling ordinal relationships across levels. Unlike token-based approaches, our method supports interpolation between proficiency levels and flexible control over edit magnitude. Experiments on controllable text simplification benchmarks demonstrate improved difficulty controllability, simplification quality, and sentence difficulty estimation performance.

1 Introduction

Text simplification (Alva-Manchego et al., 2020b) rewrites complex text into simpler forms while preserving its original meaning, with the goal of improving readability for target readers. This technique is useful for assisting people with aphasia (Carroll et al., 1998) and developing children (De Belder and Moens, 2010) to comprehend texts. We focus on text simplification for language learning and education, aiming to support learners (Petersen and Ostendorf, 2007) and teachers, e.g., by adjusting text difficulty based on students’ language proficiency. In this context, controlling difficulty levels is crucial, which has been actively studied in recent years (Scarton and Specia, 2018; Nishihara et al., 2019; Yanamoto

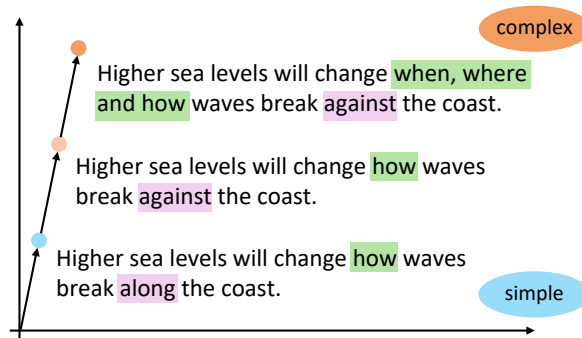


Figure 1: Our method controls the difficulty in simplification by scaling the norm of the level representation.

et al., 2022). These previous studies achieved controllability by prepending a discrete special token representing the target difficulty level to an input sentence. However, these token-based approaches do not explicitly model the ordinal relationships among proficiency levels, making it difficult to achieve smooth transitions in simplification difficulty or fine-grained control beyond predefined levels. Moreover, the existing design inherently limits difficulty control to discrete levels, precluding continuous or fine-grained control.

To enable continuous and ordinal difficulty control, we seek a conditioning mechanism whose representation can be smoothly adjusted during inference. We propose controlling difficulty through the norm of a shared embedding conditioned on the input sentence semantics. Our method prepends a shared token, [Lv], to the input sentence, allowing the encoder to produce a representation reflecting both sentence semantics and difficulty. The norm of this conditioning representation corresponds to the target difficulty level, as illustrated in Figure 1: simpler sentences yield smaller norms, while more difficult sentences yield larger ones. By editing the norm during inference, the model can continuously adjust output difficulty levels. The decoder attends to this representation while generating simpli-

fied text, naturally modeling ordinal relationships among proficiency levels and enabling fine-grained control over edit magnitude. We adopt the Common European Framework of Reference for Languages (CEFR),¹ an international standard for describing language proficiency levels. Because difficulty is controlled through continuous norm manipulation, the proposed method enables smoother transitions in output difficulty across CEFR levels.

Our motivation is based on an architectural intuition that the norm provides a continuous scalar that can be varied while preserving the embedding direction, thereby enabling interpolation. Our experiments provide empirical evidence supporting this design. We evaluate the proposed method under two tasks: difficulty estimation (Arase et al., 2022) and difficulty control (Scarton and Specia, 2018). Experimental results revealed that the proposed method achieves superior control of difficulty levels to existing methods. In addition, detailed analysis confirms that our method enables smoother and more fine-grained difficulty control than existing methods. Furthermore, the proposed method outperforms recent LLM-based approaches (Llama Team, 2024) with a substantially smaller BART model (Lewis et al., 2020). Resources related to this work are available online.²

The contributions of this study are threefold:

- We propose a norm-based conditioning method for controllable text simplification that explicitly models the ordinal structure of CEFR proficiency levels and enables continuous difficulty control.
- We demonstrate that editing the conditioning norm enables smooth interpolation of simplification difficulty beyond discrete levels.
- Experimental results show that the proposed method improves difficulty controllability and simplification quality while using substantially fewer parameters than recent LLM-based approaches.

2 Related Work

Text simplification (Nisioi et al., 2017; Zhang and Lapata, 2017; Zhao et al., 2018; Maddela et al.,

2021) is commonly formulated as a sequence-to-sequence transformation, analogous to monolingual machine translation, where edit operations on lexical and grammatical structures are acquired by training on a parallel corpus of complex-simple sentence pairs (Coster and Kauchak, 2011; Xu et al., 2015; Kajiwar and Komachi, 2016; Zhang and Lapata, 2017; Jiang et al., 2020). More recently, LLMs have demonstrated strong performance in simplification (Kew et al., 2023), outperforming the previous state-of-the-art in general text simplification (Martin et al., 2022). However, they suffer from precise control of simplicity and tend to generate overly complex words (Wu and Arase, 2025).

There has been a series of studies that aim to achieve mechanistic control of simplification process. These studies control edit operations, i.e., addition, deletion, and replacement of tokens (Kajiwar, 2019; Dong et al., 2019; Mallinson et al., 2020; Omelianchuk et al., 2021; Agrawal et al., 2021; Zetsu et al., 2022, 2024) typically using lexically constrained decoding techniques (Hokamp and Liu, 2017; Lu et al., 2022). There have been also studies that control stylistic features of outputs, such as sentence length or syntactic structures (Sheang and Saggion, 2021; Martin et al., 2020, 2022; Agrawal and Carpuat, 2023).

For aiding language learning and education, control of sentence difficulty is crucial. Scarton and Specia (2018) first addressed the control of output sentences’ difficulty by prepending a special token that represents the target difficulty to an input. Nishihara et al. (2019) improved this approach by encouraging generating words that frequently appear in a target difficulty level. Yanamoto et al. (2022) employed reinforcement learning using rewards on target and estimated difficulty levels of output sentences. Li et al. (2025) also conducted reinforcement learning on LLMs, which achieved difficulty control without a parallel corpus by harnessing the strong language generation ability of LLMs. In these previous studies, difficulty levels are treated as independent and discrete variables without interactions within.

Nonetheless, none of these methods incorporates a mechanism to model ordinal relationships among difficulty levels. This limitation hinders continuous and smooth difficulty control in text simplification, which may lead to suboptimal performance.

¹<https://www.coe.int/en/web/common-european-framework-reference-languages>

²<https://github.com/EhimeNLP/continuous-difficulty-control>

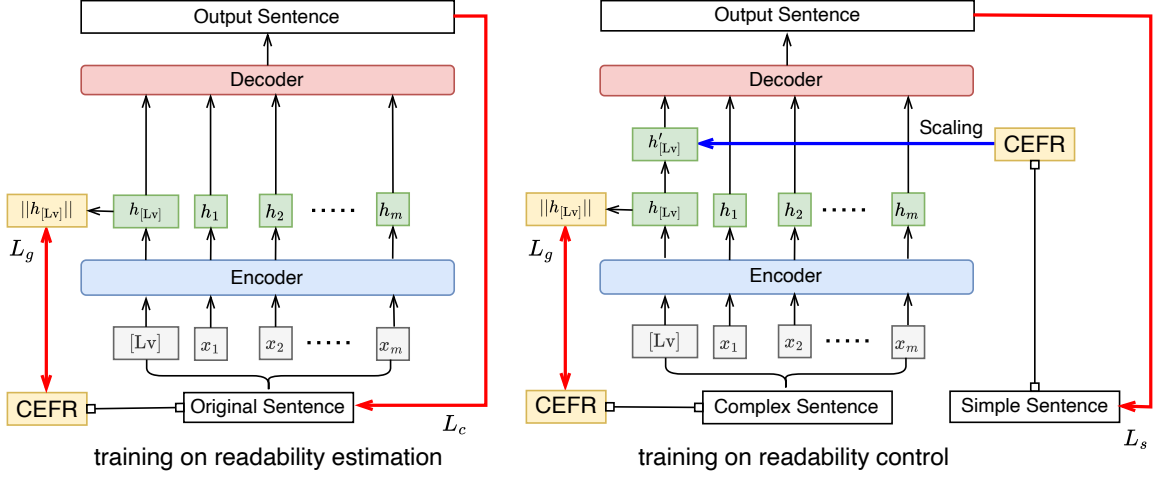


Figure 2: Model architecture (Left: training on difficulty estimation; Right: training on difficulty control).

3 Proposed Method

Our method is built on a transformer-based encoder–decoder architecture, specifically the pre-trained BART model (Lewis et al., 2020), chosen as a well-established encoder–decoder model to isolate the effect of the proposed method from differences in model architecture, scale, or pre-training. The encoder–decoder architecture provides a natural mechanism for explicitly conditioning generation through editable encoder representations, which aligns well with our norm-based difficulty control framework. Each input is prepended a special token, $[Lv]$, which serves as a shared level representation in the model. We then perform two-stage fine-tuning (Figure 2): (1) learning to represent sentence difficulty as the norm of the level representation, and (2) learning to generate simplified sentences conditioned on the level representation whose norm indicates the desired difficulty.

3.1 Training of CEFR Level Representation

The first fine-tuning aims to train the model so that its encoder can represent the difficulty level of a sentence as the norm of level representation as illustrated in Figure 1. We use N labeled samples: $\{(X_1, g_1), (X_2, g_2), \dots, (X_N, g_N)\}$, where X_i corresponds to an input sentence and $g_i \in G = \{1, 2, \dots, k\}$ represents ordinal difficulty levels. CEFR consists of six levels of A1 (corresponds to the most basic level), A2, B1, B2, C1, and C2 (the most proficient level). We thus assign $g_i = 1$ to A1 and $6 (= k)$ to C2, according to their ordinal relationships. We assume the linear spacing of CEFR levels for simplicity, which has been empirically worked out (Arase et al., 2022). Further investiga-

tion of relations among CEFR levels constitutes our future work. Note that this first fine-tuning does not require any parallel corpus for text simplification. It only uses CEFR level-annotated sentences.

The first-stage training is formulated as a multi-task auto-encoding framework, as illustrated on the left side of Figure 2, with two objectives. The first objective is learning to represent difficulty levels as norms. For each input sentence and difficulty level pair (X_i, g_i) , the encoder is trained so that the norm of the level representation $\hat{g}_i := \|h_{[Lv]}\|$, generated by the encoder from X_i , matches g_i . The loss function is the mean-squared error (MSE) between g_i and $\hat{g}_i$:

$$L_g = \frac{1}{N} \sum_{i=1}^N (g_i - \hat{g}_i)^2. \quad (1)$$

The second objective is the standard auto-encoding. The decoder is trained to reconstruct the input sentence, from the encoder’s hidden outputs $\mathbf{H}_i = \{h_{[Lv]}, h_1, \dots, h_{|X_i|}\}$ generated from the input $X_i = \{[Lv], x_1, \dots, x_{|X_i|}\}$. The loss function is cross-entropy:

$$L_c = -\frac{1}{|X_i|} \sum_{m=1}^{|X_i|} \log p(x_m | \mathbf{x}_{<m}, \mathbf{H}_i). \quad (2)$$

The overall loss function is weighted sum of L_g (Equation (1)) and L_c (Equation (2)) with a hyperparameter α tuning the training priority.

$$L = L_g + \alpha L_c \quad (3)$$

3.2 Training on Difficulty Control

As shown in Figure 2 right, the second fine-tuning learns to simplify the input sentence, attending

to the *scaled* level representation that presents the desired difficulty level. This training uses a parallel corpus annotated with difficulty levels: $\{(X_1, g_1, Y_1, t_1), \dots, (X_M, g_M, Y_M, t_M)\}$ where X_i is the input of level $g_i \in G$ and Y_i is the reference sentence of target level $t_i \in G$ ($t_i < g_i$).

The encoder generates the hidden outputs $\mathbf{H}_i$ from an input (complex) sentence X_i . Then we scale the level representation $\mathbf{h}_{[Lv]}$ to match the target level t_i :

$$\mathbf{h}'_{[Lv]} = \mathbf{h}_{[Lv]} \cdot \frac{t_i}{\|\mathbf{h}_{[Lv]}\|}. \quad (4)$$

Then, the hidden outputs with scaled level representation $\mathbf{H}'_i = [\mathbf{h}'_{[Lv]}, \mathbf{h}_1, \dots, \mathbf{h}_{|X_i|}]$ are input to the decoder to generate a simplified sentence. The loss function is the standard cross-entropy:

$$L_s = -\frac{1}{|Y_i|} \sum_{m=1}^{|Y_i|} \log p(y_m | \mathbf{y}_{<m}, \mathbf{H}'_i). \quad (5)$$

We also conduct multi-task learning as in Section 3.1 using Equation (1) to preserve the encoder’s ability to represent difficulty as the norm. The overall loss function is

$$L_f = L_g + \beta L_s, \quad (6)$$

where the hyperparameter β tunes the balance.

4 Evaluation Dataset & Implementation

4.1 Dataset

For training our model, we employ two datasets, CEFR-SP (Arase et al., 2022), which annotates CEFR levels to English sentences, and Newsela-Auto (Jiang et al., 2020), the standard parallel corpus for text simplification. We used the official training, validation, and test splits for both datasets.

CEFR-SP dataset was used for the training on difficulty level representation (Section 3.1). While the CEFR-SP dataset has two difficulty labels for each sentence by two annotators, the average value of both labels was used as the difficulty label of the corresponding sentence.

Newsela-Auto, a parallel corpus of English news articles, was used for the training on difficulty control (Section 3.2). The original Newsela-Auto assigns *grade* levels to each *document*. Therefore, in this experiment, the sentence-level CEFR was estimated and assigned to Newsela-Auto parallel

	Train	Valid	Test
CEFR-SP	13,990	1,843	1,660
Newsela-Auto	96,085	12,203	12,388

Table 1: The numbers of sentences in our datasets.

	MAE	RMSE
(Arase et al., 2022)	0.454	0.573
BERT-Regression	0.398	0.503
Ours (Step1)	0.423	0.540
Ours (Step2)	0.401	0.522
Ours (Step1→2)	0.395	0.501

Table 2: Results on the difficulty estimation task. Ours (Step1) is a variant of the proposed model that only trains the CEFR level representation (Section 3.1). Similarly, Ours (Step2) is a variant of the proposed model that only trains the difficulty control (Section 3.2). Ours (Step1→2) is the proposed method.

sentences. Specifically, we employed the sentence-level CEFR estimator³ released by Arase et al. (2022). In this process, there were cases where both complex and simple sentences were assigned the same CEFR levels. There were also cases where multiple reference sentences of the same source, which are supposed to be different *grades*, were assigned the same CEFR levels. Therefore, we removed sentence pairs if (a) both complex and simple sentences of a pair had the same CEFR level and (b) multiple references of the same source had the same CEFR levels. This filtering left us parallel sentence pairs whose complex-simple sentences have distinctive levels and multiple references of the same source all have different CEFR levels. Table 1 shows the number of samples in CEFR-SP and Newsela-Auto after filtering.

4.2 Implementation Details

We employed the pre-trained encoder-decoder model BART⁴ (Lewis et al., 2020) as the base model to benefit from its pre-training of a large corpus. The hyperparameters of α and β were both empirically set to 1.0. Throughout the training of our method, the batch size was 32, Adam (Kingma and Ba, 2015) was employed as an optimizer with a learning rate of 1e-6. Also, early stopping with three epochs on the validation loss was applied.

³<https://github.com/yukiar/CEFR-SP>

⁴<https://huggingface.co/facebook/bart-base>

	SARI	SARI submetrics			Meaning	Difficulty
		add	keep	del	BERTScore	MAE
Original	8.28	0.00	24.84	0.00	0.712	1.413
BART	38.26	5.10	31.33	78.33	0.716	0.753
BART+grade	40.27	5.66	34.32	80.82	0.723	0.578
ChatGPT	37.59	4.49	27.89	80.40	0.704	1.213
ChatGPT+grade (FSL)	38.77	4.85	28.81	82.65	0.707	1.073
Llama	38.33	5.23	29.35	80.41	0.702	1.129
Llama+grade (FSL)	40.95	5.98	29.15	87.73	0.713	0.725
Llama+grade (SFT)	38.98	6.34	31.38	79.22	0.717	0.737
Ours (Step1)	8.59	0.00	24.88	0.90	0.712	1.413
Ours (Step2)	40.82	5.65	35.09	81.72	0.723	0.521
Ours (Step1→2)	42.39	5.91	37.39	83.85	0.725	0.491

Table 3: Results on the difficulty control task: The proposed method achieved the best SARI and BERTScore, as well as the best control of the CEFR levels.

5 Evaluation: Difficulty Estimation

This section employs CEFR-SP to evaluate whether the norm of the level representation in the proposed method reliably encodes sentence difficulty. Since some sentences in CEFR-SP overlap with those in Newsela-Auto, we excluded sentences that appear in the Newsela-Auto training set. As baseline methods, we employed an existing method² (Arase et al., 2022) as well as a simple regression model on top of BERT⁵ (Devlin et al., 2019) trained on CEFR-SP. The evaluation metrics are mean absolute error (MAE) and root mean squared error (RMSE) against the human-annotated CEFR levels.

Table 2 presents the experimental results. The proposed method achieves lower error than the baseline models, demonstrating its effectiveness. Furthermore, our model trained using the two-stage strategy (Ours (Step1→2)) achieves better performance than those trained by individual stages only. This observation suggests that not only difficulty estimation but also difficulty control training contributes to improving difficulty representations.

6 Evaluation: Difficulty Control

This section evaluates the performance of the proposed method on controllable sentence simplification. We conducted experiments using the preprocessed Newsela-Auto dataset shown in Table 1.

⁵<https://huggingface.co/google-bert/bert-base-uncased>

6.1 Baselines

We compared our method with the baseline that prepends a special token of the target difficulty level to the input sentence (BART+grade) (Scarton and Specia, 2018). As the very basic baseline, we also evaluate the input sentence without any modifications (Original) and BART fine-tuning without a target difficulty level (BART).

In addition, we compared to simplification employing large language models (LLMs) (ChatGPT⁶ and Llama⁷ (Llama Team, 2024)). The LLMs were used under three settings: zero-shot inference, in which only task instructions were provided without any training; few-shot setting, in which three examples randomly sampled from the Newsela-Auto training set were included in the prompt (ChatGPT+grade (FSL) or Llama+grade (FSL)); and supervised fine-tuning on Newsela-Auto (Llama+grade (SFT)). The prompts provided to the LLMs followed Prompt 2 (Alva-Manchego et al., 2020a; Kew et al., 2023). The exact prompts and supervised fine-tuning configurations are provided in Appendix A.

6.2 Evaluation Metrics

Existing evaluation metrics for simplification are primarily designed to assess semantic adequacy and paraphrasal quality in generated sentences, but assessing the appropriateness for the target level

⁶<https://platform.openai.com/docs/models/gpt-3.5-turbo>

⁷<https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

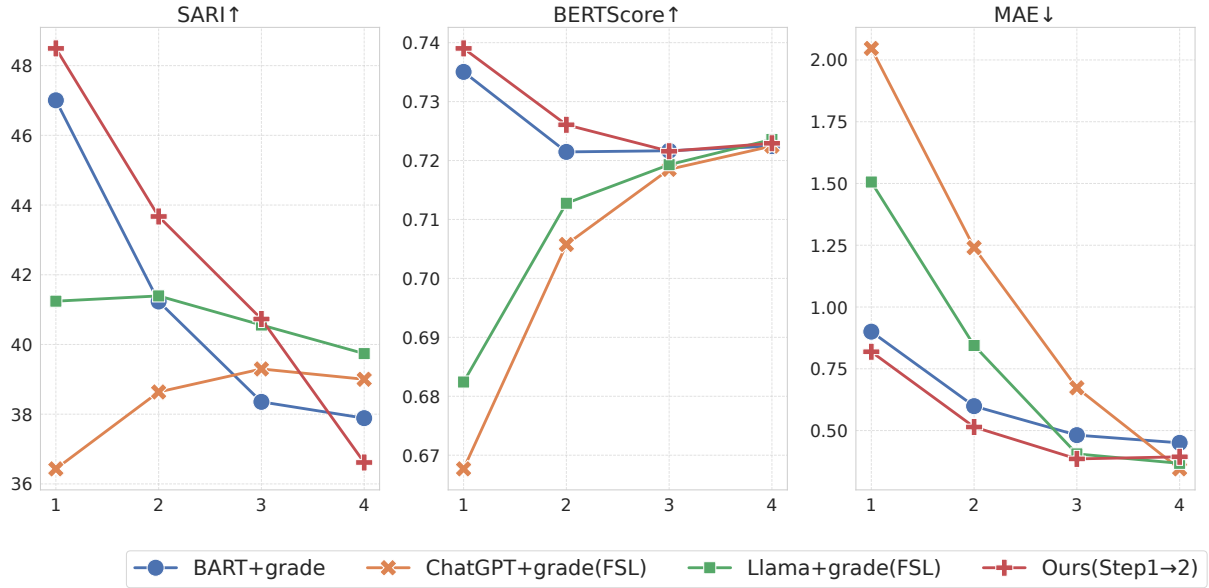


Figure 3: Performance of difficulty control on different target levels. The x -axis denotes the target difficulty level while the y -axis shows the corresponding metric scores.

has been out of scope. We conducted a meta-evaluation of existing metrics to select appropriate metrics for controlled simplification as detailed in Section 7. Based on the results, we primarily use SARI⁸ (Xu et al., 2016) to measure surface-level similarity to the reference sentences, and BERTScore⁹ (Zhang et al., 2020) to assess semantic similarity to the references. In addition, we use the error (MAE) between the target difficulty level and the estimated difficulty² of the output sentence.

6.3 Results

Overall Results Table 3 presents the experimental results. The proposed method achieves the best performance across all evaluation metrics, surpassing the LLM-based models. Focusing on the SARI submetrics (add, keep, and del), Llama+grade (FSL) exhibits high del scores but underperforms even the BART baseline in keep, indicating a tendency to excessively delete already simple expressions in the input sentence. In contrast, the proposed method achieves the best results on keep, which evaluates the preservation of simple expressions in the input, while also shows good performance on add and del. Although the proposed method achieved strong performance even when trained solely on the second-stage training, combining it with first-stage training for difficulty representation led to improvements across all evaluation

metrics. These results confirm the effectiveness of the proposed method for sentence difficulty control.

Performance per Target Level To gain deeper insights into how each model behaves under different target difficulty levels, we analyze BART+grade and ChatGPT+grade (FSL), Llama (FSL), which achieved relatively high scores in Table 3. Figure 3 plots the metric scores per target difficulty levels of {1 (A1), 2, 3, 4 (B2)}. Remarkably, the proposed method consistently outperforms the baseline methods on more challenging lower target levels (i.e., 1 to 3). The LLM-based models, in contrast, achieved higher scores on the level 4.

Distribution of Difficulty Levels Figure 4 shows the distributions of the estimated difficulties of the generated sentences per target level, which reveals whether generated sentences are overly simple or complex against the target. It confirms that our method can generate more sentences closer to the target level in lower levels, particularly on levels of 1 and 2. In contrast, the LLM-based models tend to generate excessively complex sentences than the target level. As the target difficulty increases, the performance gap narrows. It is notable that the distributions of levels by the LLM-based models barely change, indicating their insensitivity to the target difficulty. This finding is consistent with the previous study that revealed LLMs tend to generate overly complex words (Wu and Arase, 2025).

⁸<https://github.com/feralvam/easse>

⁹https://github.com/Tiiiger/bert_score

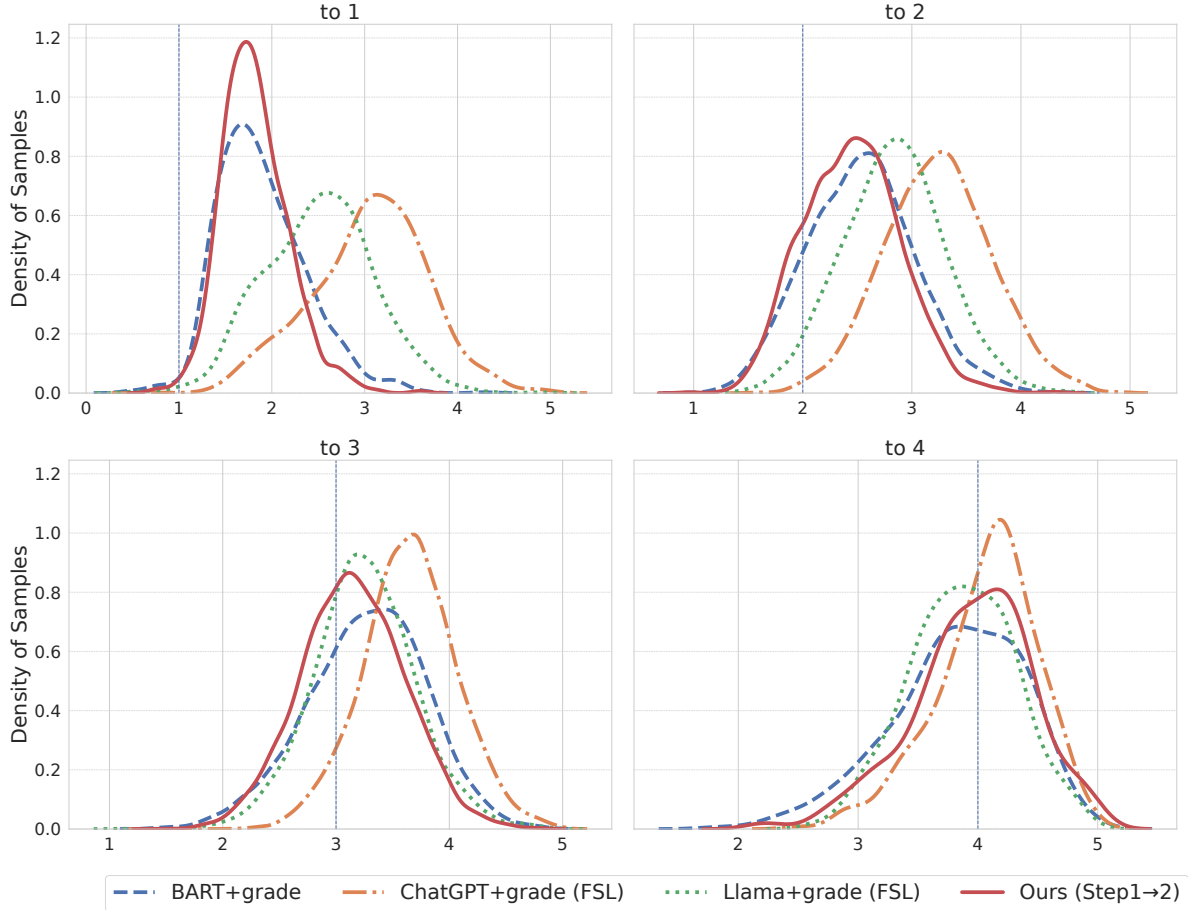


Figure 4: Distributions of the estimated difficulty of generated sentences per target difficulty level. The dashed vertical line indicates the target level. The x -axis shows the estimated difficulty and the y -axis indicates density.

6.4 Human Evaluation on Controllability

We further conduct a human evaluation to assess the controllability in simplification levels from readers’ perspectives.

Settings We randomly sampled 50 sentences with difficulty levels of $g \in \{5, 6\}$ from our Newsela-Auto test set. Each sentence was simplified to target difficulty levels of coarse-grained $\{4, 3, 2, 1\}$ or fine-grained $\{4.0, 3.5, 3.0, 2.5\}$. For the proposed method, simplifications of intermediate target levels (e.g., 3.5 and 2.5) are generated by scaling the norm of the conditioning representation. In contrast, the baseline methods cannot directly handle such intermediate levels. Therefore, for BART+grade, we approximated intermediate targets by linearly interpolating the corresponding level embeddings. For the LLM-based baselines, we specified the intermediate CEFR levels in the prompts.

Annotation We recruited three annotators on Amazon Mechanical Turk¹⁰ who held the “Master” qualification. The task was to order four simplified sentences from the same source by difficulty.¹¹ The inter-annotator agreement was assessed using pairwise comparisons based on Spearman’s rank correlation coefficient. As a result, we confirmed a fairly high agreement rate: on average, 0.918 on coarse-grained and 0.911 on fine-grained control tasks, respectively.

Evaluation metrics We compute accuracy (Acc) and Spearman’s rank correlation coefficient (ρ) between the human-assessed orders and the intended difficulties in simplification. The accuracy scores 1 only when the order of assessed levels of generated sentences are perfectly matched with that of the target levels, otherwise 0.

¹⁰<https://www.mturk.com/>

¹¹We paid our cloudworkers \$12 per hour. This is the recommended rate on Prolific, one of the widely used crowdsourcing platforms. <https://www.prolific.com/pricing>

	Discrete Difficulty Control $\rightarrow \{4, 3, 2, 1\}$		Fine-grained Difficulty Control $\rightarrow \{4.0, 3.5, 3.0, 2.5\}$	
	Acc	ρ	Acc	ρ
BART+grade	0.380	0.906	0.007	0.802
Llama+grade (FSL)	0.333	0.812	0.053	0.596
Proposed (Step2)	0.467	0.917	0.133	0.852
Proposed (Step1 $\rightarrow$ 2)	0.687	0.962	0.387	0.897

Table 4: Human evaluation results (average of scores computed on each annotator)

	Sentence	Target CEFR
Original	Some marine biologists and ocean advocates warn that such a compassion-laden response is little more than a futile attempt to wrap a Band-Aid on an oceanwide problem.	5.0
System	Some marine biologists and ocean advocates say that such a compassionate response is little more than a Band-Aid on an oceanwide problem.	4.0
	Some marine biologists and ocean advocates say that such a response is little more than a way to help the ocean.	3.5
	Some scientists and ocean advocates say that is not the answer.	3.0
	Some scientists and ocean advocates say that is not the case.	2.5

Table 5: Example of fine-grained difficulty control by the proposed method. Lower target difficulty levels lead to simpler vocabulary and sentence structures in the generated text.

Results Table 4 presents the average accuracies and correlation coefficients against the three annotators. The proposed method achieves the best scores on both coarse (left) and fine-grained (right) simplification. We would remark that our two-stage training resulted in higher performance than the training Step 2 alone, providing empirical evidence that Step 1 contributes to the controllability improvement. In contrast, the comparison methods resulted in low accuracy on the fine-grained simplification. This result suggests that discrete-level conditioning makes it difficult to model fine-grained changes in target difficulty. The proposed method enables fine-grained difficulty control through norm editing, leading to substantially improved performance. These results provide independent evidence that the gains of our method are not merely artifacts of the automatic estimator.

Output Examples Table 5 presents an example: a level 5 sentence was simplified to the target difficulties of 4.0, 3.5, 3.0, and 2.5. As the target difficulty decreases, our method edits the source sentence more extensively to generate sentences with simpler vocabulary and structures. These ex-

amples further demonstrate that our method can appropriately control sentence difficulty based on norm editing of the level representation.

7 Meta-Evaluation of Evaluation Metrics

Evaluation metrics for controlled text simplification should assess not only semantic adequacy with the input or reference sentence, but also whether the output sentence is appropriate for the target difficulty level. Therefore, this section conducts a meta-evaluation of existing evaluation metrics to see how they can reflect the appropriateness of simplification outputs in terms of difficulty control.

7.1 Settings

First, we created five sentence sets with equivalent meaning but different difficulty levels from the Newsela-Auto (Jiang et al., 2020). These five sentences are sorted in descending order of their difficulty level. Among these sorted sentences $s = \{s_1, s_2, s_3, s_4, s_5\}$, we considered the most complex s_1 as the input sentence and simplest s_5 as the reference sentence, and automatically evaluated the other three sentences (which are regarded as outputs from the text simplification model). The

requirement for an evaluation metric is to assign lower scores to more difficult sentences (s_2) and higher scores to simpler sentences (s_4).

We employed accuracy (Acc), Spearman’s rank correlation coefficient (ρ), and nDCG (Normalized Discounted Cumulative Gain) to measure the effectiveness of the existing evaluation metrics: SARI (Xu et al., 2016), BERTScore¹² (Zhang et al., 2020), LENS¹³ (Maddela et al., 2023), and LENS-SALSA¹⁴ (Heineman et al., 2023).

7.2 Results

Table 6 shows the results: SARI scored significantly higher than the other metrics in all of accuracy, Spearman’s rank correlation coefficient, and nDCG, followed by BERTScore. In contrast, recently proposed metrics, LENS and LENS-SALSA performed poorly. We conjecture this trend is relevant to the fine-tuning process involved in them. LENS and LENS-SALSA employ a fine-tuned model to improve correlation with human judgment of text simplification; however, difficulty levels have been out of scope. This leads to lower alignment of level appropriateness. Based on these results, we adopted SARI and BERTScore as the evaluation metric in our primary experiments.

8 Conclusion

This study proposed a text simplification method for continuous difficulty control based on CEFR levels. By controlling difficulty through the norm of a shared conditioning embedding, the proposed method models ordinal relationships among proficiency levels and enables fine-grained control over output difficulty. Experimental results demonstrated improved difficulty controllability and simplification quality over existing controllable simplification and recent LLM-based approaches.

We employed an encoder–decoder architecture because it naturally supports explicit conditioning through editable encoder representations, making it well-suited for norm-based difficulty control. Extending the proposed framework to more recent encoder–decoder models and decoder-only LLMs is an important direction for future work, potentially combining continuous difficulty control with stronger generation capabilities.

¹²https://github.com/Tiiiger/bert_score

¹³<https://huggingface.co/davidheineman/lens>

¹⁴<https://huggingface.co/davidheineman/lens-salsa>

	Acc	ρ	nDCG
SARI	0.733	0.953	0.994
BERTScore	0.408	0.841	0.970
LENS	0.226	0.731	0.955
LENS-SALSA	0.142	0.559	0.942

Table 6: Meta-evaluation results of evaluation metrics

Limitations

This study focused on English text simplification using CEFR-based difficulty control. Therefore, the effectiveness of the proposed method for other languages or alternative difficulty frameworks remains unclear.

In addition, our method assumes a linear relationship among CEFR levels through norm scaling. Although this assumption was effective in our experiments, more sophisticated formulations of proficiency-level relationships may further improve controllability.

Finally, our method was trained using automatically estimated CEFR labels for the Newsela-Auto corpus. While both automatic and human evaluations demonstrated the effectiveness of the proposed method, errors in difficulty estimation may affect training quality and controllability. Developing high-quality parallel corpora with manually annotated sentence-level difficulty remains an important direction for future work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP25K03233.

References

- Sweta Agrawal and Marine Carpuat. 2023. [Controlling Pre-trained Language Models for Grade-Specific Text Simplification](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12807–12819.
- Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. [A Non-Autoregressive Edit-Based Approach to Controllable Text Simplification](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769.
- Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020a. [ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations](#). In *Proceedings of the*

- 58th Annual Meeting of the Association for Computational Linguistics, pages 4668–4679.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020b. [Data-Driven Sentence Simplification: Survey and Benchmark](#). *Computational Linguistics*, 46(1):135–187.
- Yuki Arase, Satoru Uchida, and Tomoyuki Kajiwar. 2022. [CEFR-Based Sentence Difficulty Annotation and Assessment](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6206–6219.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. [Practical Simplification of English Newspaper Text to Assist Aphasic Readers](#). In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.
- William Coster and David Kauchak. 2011. [Simple English Wikipedia: A New Text Simplification Task](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.
- Jan De Belder and Marie-Francine Moens. 2010. [Text Simplification for Children](#). In *Proceedings of the SIGIR 2010 Workshop on Accessible Search Systems*, pages 19–26.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.
- 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.
- David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. 2023. [Dancing between success and failure: Edit-level simplification evaluation using SALSA](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3495.
- Chris Hokamp and Qun Liu. 2017. [Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1535–1546.
- 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.
- Tomoyuki Kajiwar. 2019. [Negative Lexically Constrained Decoding for Paraphrase Generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052.
- Tomoyuki Kajiwar and Mamoru Komachi. 2016. [Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158.
- Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023. [BLESS: Benchmarking Large Language Models on Sentence Simplification](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13291–13309.
- Diederik P. Kingma and Jimmy Lei Ba. 2015. [Adam: A Method for Stochastic Optimization](#). In *Proceedings of the 3rd International Conference for Learning Representations*.
- 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.
- Guanlin Li, Yuki Arase, and Noel Crespi. 2025. [Aligning Sentence Simplification with ESL Learner’s Proficiency for Language Acquisition](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 492–507.
- Llama Team. 2024. [The Llama 3 Herd of Models](#). *arXiv:2407.21783*.
- Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. [NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799.
- 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.
- Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. [LENS: A learnable evaluation metric](#)

- for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 16383–16408.
- 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.
- Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. **Controllable Sentence Simplification**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698.
- Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. **MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases**. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664.
- Daiki Nishihara, Tomoyuki Kajiwar, 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*, pages 85–91.
- 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.
- Sarah E Petersen and Mari Ostendorf. 2007. **Text Simplification for Language Learners: A Corpus Analysis**. In *Proceedings of the Workshop on Speech and Language Technology in Education*, pages 69–72.
- 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*, pages 712–718.
- Kim Cheng Sheang and Horacio Saggion. 2021. **Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer**. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352. Association for Computational Linguistics.
- Xuanxin Wu and Yuki Arase. 2025. **An in-depth evaluation of large language models in sentence simplification with error-based human assessment**. *ACM Trans. Intell. Syst. Technol.* Just Accepted.
- 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.
- Daiki Yanamoto, Tomoki Ikawa, Tomoyuki Kajiwar, Takashi Ninomiya, Satoru Uchida, and Yuki Arase. 2022. **Controllable Text Simplification with Deep Reinforcement Learning**. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 398–404.
- Tatsuya Zetsu, Yuki Arase, and Tomoyuki Kajiwar. 2024. **Edit-Constrained Decoding for Sentence Simplification**. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7161–7173.
- Tatsuya Zetsu, Tomoyuki Kajiwar, and Yuki Arase. 2022. **Lexically Constrained Decoding with Edit Operation Prediction for Controllable Text Simplification**. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability*, pages 147–153.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. **BERTScore: Evaluating Text Generation with BERT**. In *Proceedings of the 8th International Conference on Learning Representations*.
- 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.
- Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. **Integrating Transformer and Paraphrase Rules for Sentence Simplification**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173.

A Details of LLM Baselines

A.1 Prompts

We used two types of prompts for the LLM-based baselines. For the models with difficulty control (+grade), we used the following system prompt:

Please rewrite the following complex sentence in order to make it easier to understand by non-native speakers of

English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentence needs to be grammatical, fluent, and retain the main ideas of its original counterpart without altering its meaning.

CEFR (Common European Framework of Reference for Languages) levels range from A1 to C2, which correspond to levels 1 to 6. The input sentence begins with a token indicating the target CEFR level (for example, <4>). Please generate a simplified version of the sentence that is appropriate for the given target CEFR level.

For the few-shot setting, each demonstration consisted of the following user and assistant messages:

User Complex: <[TARGET]>[SOURCE]

Assistant Simple: [SIMPLIFIED]

For the models without difficulty control, we used the following system prompt:

Please rewrite the following complex sentence in order to make it easier to understand by non-native speakers of English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentence needs to be grammatical, fluent, and retain the main ideas of its original counterpart without altering its meaning.

A.2 Supervised Fine-tuning

For Llama+grade (SFT), we performed supervised fine-tuning using SFTTrainer with LoRA. The batch size was 4, the learning rate was 1×10^{-5} , and the number of gradient accumulation steps was 8. The model was evaluated every 500 training steps. Early stopping with a patience of three evaluations was applied, and the checkpoint with the lowest validation loss was selected. For LoRA, the rank r was 4, lora_alpha was 8, and the dropout rate was 0.05.