Enhancing Readability-Controlled Text Modification with Readability Assessment and Target Span Prediction

Fengkai Liu, John S. Y. Lee
Department of Linguistics and Translation
City University of Hong Kong
Hong Kong SAR, China
fengkaliu3-c@my.cityu.edu.hk, jsylee@cityu.edu.hk

Abstract

Readability-controlled text modification aims to rewrite an input text so that it reaches a target level of difficulty. This task is closely related to automatic readability assessment (ARA) since, depending on the difficulty level of the input text, it may need to be simplified or complexified. Most previous research in LLM-based text modification has focused on zero-shot prompting, without further input from ARA or guidance on text spans that most likely require revision. This paper shows that ARA models for texts and sentences, as well as predictions of text spans that should be edited, can enhance performance in readability-controlled text modification.

1 Introduction

Readability-controlled text modification aims to rewrite the input text so that it reaches a target level of difficulty (Chi et al., 2023; Farajidizaji et al., 2024). The nature of the modification depends on whether the difficulty level of the input text is higher, lower, or the same as the target level. If higher, the system should perform *text simplification*, which has been extensively studied (Agrawal and Carpuat, 2023; Alva-Manchego et al., 2020b; Mo and Hu, 2024; Štajner et al., 2017). If lower, the system needs to "increase the lexical and syntactic complexity of a text" (Berov and Standvoss, 2018), a task that has been studied under various terms including "textual embellishment" (Berov and Standvoss, 2018), "text elaboration" (Farajidizaji et al., 2024) and "complexification" (Chi et al., 2023). In Table 1, for example, the input text is simplified to the target level "elementary" and complexified to the "advanced" level, but is left unchanged when the target level is "intermediate", since it is already at the intermediate level.

In the most recent study on readability-controlled text modification (Farajidizaji et al., 2024), zero-shot prompting of Large Language Models (LLMs) has been shown capable of nudging the difficulty level of a text towards the target level, in terms of the Flesch Reading-Ease Score (FRES) (Kincaid et al., 1975). There was no attempt, however, to further improve the proposed method using training data from text modification or automatic readability assessment (ARA) (Martinc et al., 2021), which is directly relevant to readability-controlled text modification. Since an ARA model estimates the difficulty level of a text, it can help determine whether and how much modification is needed, and in which parts of the input text.

This paper investigates whether ARA and text

Input (intermediate level)	Argentina is unhappy that the US outdoor clothing retailer, Patagonia, is claiming a domain name that has been known far longer as a region of spectacular beauty that also has its own parliament ...
Output for target level "elementary"	Argentina is unhappy that the US outdoor clothing retailer, Patagonia *wants* a domain name that ...
Output for target level "intermediate"	Argentina is unhappy that the US outdoor clothing retailer, Patagonia, is claiming a domain name that has been known far longer as a region of spectacular beauty that ...
Output for target level "advanced"	Argentina has *lodged an expression of its unhappiness* that the US outdoor clothing retailer, Patagonia, is claiming a domain name that ...

Table 1: Example input and outputs for readability-controlled text modification at the target levels "elementary", "intermediate", and "advanced"

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 293–303

modification training data can improve the performance of readability-controlled text modification. Specifically, we address the following research questions:

ARA data Can text modification performance be improved with automatic assessment of the difficulty of a text and/or of individual sentences?

Text modification data Can text modification performance be improved with automatic identification of text spans that require editing?

The rest of the paper is organized as follows. After a review of previous research (Section 2), we describe the proposed auxiliary models for guiding the text modification process (Section 3), and the prompt that incorporates information from these auxiliary models (Section 4). We then present our dataset (Section 5) and report automatic (Section 6) and human (Section 7) evaluation results.

2 Previous work

Readability-controlled text modification may be viewed as a paraphrasing task (Barzilay and Lee, 2003) with an additional constraint, namely, that the output must be at the target level of difficulty. To our knowledge, there have been only three published studies on text modification. Trott and Rivière (2024) assessed the ability of GPT-4 Turbo to make a text easier or harder, but did not require the output to reach a specific difficulty level. Chi et al. (2023) focused on text modification at the sentence level. T5 models were fine-tuned on sentence pairs from text simplification datasets; for sentence complexification, T5 models were fine-tuned with the inputs and outputs reversed. Most related to our work, Farajidizaji et al. (2024) generated versions of the input text at different target FRES, using zero-shot prompting of ChatGPT and Llama-2. In terms of the readability of the modified texts, their best model achieved 24.2% accuracy in reaching the target FRES. In terms of the content of the modified texts, however, no evaluation was reported because of the lack of gold outputs.

Other studies have focused exclusively on either text complexification or text simplification. For the former task, Berov and Standvoss (2018) trained an LSTM model on the inverse of WikiLarge, a simplification corpus based on simple and standard English Wikipedia. The model is then applied to complexify the texts in a story corpus. Naskar et al. (2019) adopted a similar LSTM encoder-decoder, and reported both BLEU and human evaluation.

For text simplification, early studies tended to take the relative approach, i.e., to make the input text easier but without specifying a target difficulty level or audience (Belder and Moens, 2010; Kajiwara et al., 2013; Paetzold and Specia, 2017; Siddharthan, 2002). Recent work has increasingly recognized the need for text simplification to an absolute target level (Štajner et al., 2017). Nishihara et al. (2019) used lexical and syntactic complexity features, while Yanamoto et al. (2022) applied deep reinforcement learning using a reward calculated based on the difference between the difficulty of the output sentence and the target difficulty. Agrawal et al. (2021) used a non-autoregressive model to iteratively edit the source sentence. Agrawal and Carpuat (2023) predicts low-level control tokens for text simplification. Similar to this work, more recent studies have exploited LLMs. For example, the *SimplifyMyText* system rewrites the input text in plain language (Färber et al., 2025). In the *ExpertEase* system, LLM-based agents collaborate in text simplification playing the roles of the expert, the teacher, and the student (Mo and Hu, 2024).

3 Approach

Readability-controlled text modification requires judgment on text difficulty, and on the kinds of content that are most suitable for revision. Therefore, it may potentially benefit from auxiliary models that can assess the difficulty of sentences (Section 3.1) and texts (Section 3.2), and predict the text spans that require revision (Section 3.3). For LLM-based text modification, the information produced by these auxiliary models can be incorporated into the prompt (Section 4).

3.1 Sentence ARA (Sent ARA) Model

Generally, it is not necessary to rewrite every sentence in a text, even when transforming the text to a distant target difficulty level. To make judicious changes, it could be useful to highlight the sentences that most likely require revision, i.e., the easiest or the most complex sentences.

A sentence-level ARA model predicts the complexity of an individual sentence (Brunato et al., 2018; Garbacea et al., 2021; Liu et al., 2025; Lu et al., 2020; Schicchi et al., 2020; Štajner et al., 2017). This information can help guide the LLM in identifying sentences that deviate most from the

With the owners out of the cats' line of vision, researchers played recordings of three strangers calling the cats' names followed by a call from the cat's owner and then by the call of another stranger. `advanced` Researchers charted the cats' reactions by measuring a number of responses, including head movements, tail and ear movements, eye dilation and vocalization or whether they moved their paws. `advanced` When strangers called their names, the cats had no reaction to the voices whatsoever. `intermediate`

Table 2: Example output of the Sent ARA Model (Section 3.1), which labels each sentence in the input text with its difficulty level.

*With *the *owners *out *of *the *cats *' *line *of *vision *, *researchers *played *recordings *of *three *strangers *calling *the *cats *' *names *followed *by *a *call *from *the *cat *'s *owner *and *then *by *the *call *of *another *stranger *. Researchers charted the cats ' reactions by measuring a number of responses *, *including head movements , tail and ear movements , eye dilation and *vocalization *or whether they moved their paws . When strangers called their names , the cats had no reaction to the voices whatsoever .

Table 3: Example output of the Target Span Prediction Model (Section 3.3): words predicted by the model to require revision are marked with asterisks.

target difficulty level. Table 2 shows an example assessment, which labels the level of each sentence in a text as "elementary", "intermediate", or "advanced".

3.2 Text ARA Model

As opposed to sentence-level ARA, a text-level ARA model (Martinc et al., 2021) assesses the overall difficulty level of a text. This model can identify input texts that are already at the target difficulty level, and therefore do not require any modification. Further, since the LLM may not be able to modify the input text to its target level in one round (Farajidizaji et al., 2024), this model can also determine the necessity of an additional round of modification.

3.3 Target Span Prediction Model

As a preliminary step before text modification, it could be useful to identify the words that require revision (Chen and Meurers, 2019; Collins-Thompson, 2014; Liu et al., 2024). For example, complex word identification can serve as the first step in a text simplification pipeline to identify the *target words*, i.e., the difficult vocabulary items that should be replaced (Gooding and Kochmar, 2019; Paetzold and Specia, 2016; Shardlow, 2014).

In the context of text modification, we will use the term *target span* to refer to the parts of the input text that should be edited. Example input-output pairs of text simplification and complexification can be used for training a tagger that predicts these spans. Compared to the Sent ARA Model (Section 3.1), this model can provide more fine-grained guidance. In the example shown in Table 3, both sentences and individual words (e.g., "including") have been predicted to require editing.

4 Prompt implementation

The system prompt (Table 9 in Appendix A.1) describes the text modification task. The average FRES (Kincaid et al., 1975) is provided for each difficulty level, since the LLM may not be familiar with the difficulty scale.

Table 4 shows the user prompt and the auxiliary models (Section 3) from which the content is derived. The prompt states both the target difficulty level, as well as the difficulty level of the source text ("elementary", "intermediate", or "advanced"), as estimated by the Text ARA Model (Section 3.2).

In the input text, asterisks are placed on each word that is predicted to require editing, according to the Target Span Prediction Model (Section 3.3). Finally, each sentence in the input text is shown with its difficulty level, as estimated by the Sent ARA Model (Section 3.1).

As demonstrations, two sample input texts are shown with their gold output text and gold predictions from the Sent ARA, Text ARA and Target Span Prediction Models[1].

4.1 Proposed prompts

The following variations of the proposed prompt were implemented:

SentARA+Span The full prompt (Table 4).

[1]Using the actual rather than gold predictions from these models led to slightly worse performance.

Prompt template	Model
Rewrite the following <text_ara_output> passage at the <target_level> level. Tokens starting with a '*' symbol in the source text indicate the words that were changed from the source passage to the rewritten passage. <demonstrations> Source passage: <target_span_output> Difficulty of individual sentences in source passage: <sent_ara_output> Rewritten passage:	Text ARA Target Span Prediction Sent ARA

Table 4: Prompt for text modification (left), which makes use of information provided by the auxiliary models (right): <target_level> is the target difficulty level; <text_ara_output> is the predicted level of the input text; <target_span_output> is the input text with target span predictions (Table 3); <sent_ara_output> is the predicted level of each sentence in the input text (Table 2); <demonstrations> are the two sample input/output pairs

SentARA Information from the Target Span Prediction Model is omitted.

SentARA+Span+Ling The full prompt, with linguistic features (Section 4.3) added.

SentARA+Span$^{\times n}$ The full prompt, iteratively issued until the Text ARA Model predicts the input text has reached the target level, up to a maximum of n iterations.

4.2 Chain-of-thought prompts

Chain-of-thought (CoT) guides LLMs in generating their own intermediate steps for completing a task. CoT reasoning has led to robust performance in multiple NLP tasks (Brown et al., 2020; Chen et al., 2019; Ling et al., 2017; Wei et al., 2022). We implemented the following CoT prompts:

CoT The prompt uses the instruction "Let's think step by step" (Kojima et al., 2022) to obtain the reasoning for text modification. The LLM-generated reasoning is then included after the input text in the prompt, which does not use any information from the auxiliary models.

CoT (zero-shot) Same as above, except that no sample input/outputs are provided.

4.3 Baseline prompts

Linguistic features have been shown to be effective for text simplification (Agrawal et al., 2021; Maddela et al., 2021; Mo and Hu, 2024; Nishihara et al., 2019; Yanamoto et al., 2022). To identify the most salient features, we extracted all available features in the Lexical Complexity Analyzer (Lu, 2012) and calculate their correlation with the difficulty levels of the text in our training data. According to the Pearson Correlation Coefficient (Table 10 in Appendix A.2), the top 5 features are Root Text-to-Token Ratio (TTR), which measures lexical diversity; Corrected TTR, a refined version of TTR accounting for text length; Number of Different Words, which counts the number of unique words; the Uber Index, a composite, holistic measure for lexical complexity and diversity; and L2, the proportion of content words. The following baseline prompts were implemented:

Ling This prompt includes only the statement: "The measurements of the five linguistic features of the source passages are <ling_feats>", where <ling_feats> refers to the five features mentioned above.

Vanilla No linguistic feature or auxiliary model is used.

Vanilla (Zero-shot) Same as above, and no sample input/outputs are given.

5 Data

Newsela (Xu et al., 2015) is a graded parallel corpus derived from 1,911 news articles. For each article, simplified versions have been composed by professional editors for students between Grade 2 and Grade 12.

To facilitate learning of the revision patterns across the spectrum of difficulty levels, for each of these 1,911 articles, we retrieved three versions that span the grades: one version between Grades 2 and 5, which we will refer to as the "elementary" level

(ele); one version between Grades 6 and 8, the "intermediate" level (int); and one version between Grades 9 and 12, the "advanced" level (adv).

In our experiments, each of these 5,733 texts is to be revised to all three target levels {ele,int,adv}. Since the level of the source text is not disclosed to the system, it does not know whether the text should be simplified (int->ele, adv->ele, adv->int), complexified (ele->int, ele->adv, int->adv), or left unchanged (ele->ele, int->int, adv->adv).

6 Automatic Evaluation

6.1 Implementation details

We used Meta-Llama-3.1-8B-Instruct[2] for all prompts (Section 4). The auxiliary models were implemented as follows:

Sent ARA Model Following the approach proposed by Liu and Lee (2023), we trained a BART-large (Lewis et al., 2019) model to classify a sentence [3] at the ele, int, or adv level.

Text ARA Model Following the approach proposed by Lee et al. (2021), we trained a neural ARA model by fine-tuning BART (Lewis et al., 2020) on the Newsela dataset.[4]

Target Span Prediction Model We trained a tagger to label each word in the source text as REVISE or KEEP. As shown in Table 3, the words tagged as REVISE will be asterisked in the source text in the prompt. The gold labels were derived from the sentence-aligned text pairs from Newsela (Section 5), with sentence alignments automatically produced by SentAlign [5] (Steingrímsson et al., 2023). All words in a source sentence that are not in the aligned target sentence are considered REVISE; all words in a source sentence that is not aligned to any target sentence are also considered REVISE, since they are deleted. We trained six separate RoBERTa-based (Liu et al., 2019) sequence taggers[6] to cover all combinations of source and target levels ({ele,int,adv}).

We also attempted training these three models using several other transformers, but did not produce any significant improvement in performance.

6.2 Evaluation metrics

The output text should have the target complexity and appropriate content. Complexity is evaluated with two metrics. The first, Mean Absolute Error (**MAE**) in FRES, is the difference between the FRES of the output text and the FRES of the gold text. We also report **Accuracy**, i.e. whether the output text is at the target level of difficulty. The difficulty level of the output text is estimated with the Text ARA Model (Section 3.2).

The content quality of the output text is evaluated with four metrics. To determine the degree of meaning preservation, the semantic similarity between the source text and output text is evaluated using **BERTScore** (Zhang et al., 2019). Further, the output text is compared against the gold text using three widely adopted metrics in text simplification evaluation: **BLEU** (Papineni et al., 2002)[7], **SARI** (Xu et al., 2016)[8], and **D-SARI** (Sun et al., 2021) which aims at document-level simplification[9].

6.3 Results

6.3.1 Auxiliary models

To construct the prompt for each input text (Table 4), we obtained outputs from the auxiliary models (Section 3) using 5-fold cross validation. The Text ARA Model achieved an accuracy of 98.87% on the three-way classification of difficulty level (ele, int, or adv). The high accuracy validates its reliability as an evaluation metric for text complexity (Section 6.2). In contrast, the use of FRES, even with score thresholds optimized on the Newsela dataset, would yield only 66.08% accuracy on the three-way classification of difficulty level.

The Sent ARA Model performed at 0.680 accuracy and 0.674 F1-score on the three-way classification of difficulty level for sentences. The Target Span Prediction Model attained 0.454 precision and

[2] meta-llama/Meta-Llama-3.1-8B-Instruct

[3] We used the BartForSequenceClassification model from the transformers library of HuggingFace (Wolf et al., 2020)

[4] We used the pre-trained base version of BART from Huggingface (Wolf et al., 2020).

[5] https://github.com/steinst/SentAlign

[6] We used the RobertaForTokenClassification model from HuggingFace (Wolf et al., 2020).

[7] The NLTK (Bird, 2006) implementation was used.

[8] The EASSE simplification evaluation suite (Alva-Manchego et al., 2019) was used.

[9] The implementation by Sun et al. (2021) (https://github.com/RLSNLP/Document-level-text-simplification) was used. include D-SARI incorporates several penalty factors, in addition to the add, keep and delete scores in SARI. The same weights for these scores were used for both text simplification and development.

Aux. Model	Prompt method	D-SARI	SARI	BLEU	BertScore	Accuracy	MAE ↓
Nil	Vanilla (zero-shot)	16.18	43.35	39.73	0.878	0.512	16.83
	CoT (zero-shot)	17.90	43.38	45.18	0.875	0.588	16.36
	Vanilla	15.08	43.22	41.74	0.890	0.584	11.80
	CoT	18.32	42.96	50.70	0.890	0.614	10.76
	Ling	16.05	42.51	47.08	0.902	0.487	8.52
Sent ARA	SentARA	17.48	41.20	45.33	0.901	**0.720**	9.19
Sent ARA and Target Span Prediction	SentARA+Span	21.38	**49.39**	52.28	0.904	0.609	**7.74**
	SentARA+Span+Ling	20.34	46.48	51.30	**0.908**	0.559	7.83
	SentARA+Span$^{\times 2}$	**22.69**	49.22	**52.80**	0.902	0.688	7.74
	SentARA+Span$^{\times 3}$	22.66	49.16	52.72	0.902	0.698	7.80

Table 5: Text modification performance using different auxiliary models (↓ means smaller is better)

Task	Prompt	P	R
Overall	Vanilla (zero-shot)	0.640	0.479
	SentARA	0.642	0.549
	SentARA+Span	**0.694**	**0.603**
Simp.	Vanilla (zero-shot)	0.644	0.482
	SentARA	0.639	0.547
	SentARA+Span	**0.665**	**0.568**
Comp.	Vanilla (zero-shot)	0.635	0.476
	SentARA	0.645	0.551
	SentARA+Span	**0.719**	**0.634**

Table 6: Performance in identifying text spans to edit (without regard to the quality of the final output), with breakdown into simplification (simp.) and complexification (comp.)

0.650 recall when simplifying texts, and 0.448 precision and 0.657 recall when complexifying texts.

6.3.2 Effect of Sentence ARA

Table 5 presents experimental results on text modification. When none of the auxiliary models is used ("Nil" row in Table 5), CoT prompting gave the best Accuracy (0.614), D-SARI (18.32) and BLEU (50.70) scores, though it was outperformed by the CoT (zero-shot) prompt in terms of SARI. Both of these CoT prompts improved performance over their vanilla version. Consistent with Wei et al. (2022) and Kojima et al. (2022), the self-generated reasoning steps were helpful in guiding the LLM in performing text modification. The linguistic features (Section 4.3) led to the best result in terms of MAE (8.52) and BertScore (0.902).

The use of the Sent ARA Model led to the highest Accuracy (0.720). To better understand the effect of this auxiliary model, we measured its precision and recall in identifying text spans for revision, without considering the quality of the actual revision. As shown in Table 6, the gains of the Sent ARA prompt over the Vanilla baseline were mostly due to the recall (0.549 vs. 0.479). This suggests that the ARA predictions helped the LLM in selecting sentences for revision that were missed by the baseline.

However, the use of the Sent ARA Model did not generally improve the quality of the modified text. It was outperformed by the CoT and Ling prompts in most metrics other than Accuracy.

6.3.3 Effect of Target Span Prediction

Incorporating predictions of the target spans (SentARA+Span) resulted in the best overall performance (Table 5). These predictions helped produce output texts that resembled the gold texts to a greater extent. In terms of the content, SentARA+Span attained higher D-SARI, SARI and BLEU scores compared to all baselines. As shown in Table 6, it yielded higher precision (0.694) and recall (0.603) than SentARA, likely because it was able to make judicious choices in selecting individual words for revision, whereas SentARA provided guidance only at the sentence level. In terms of complexity, it also produced outputs that are closest to the gold texts in FRES (7.74 MAE). However, it was outperformed by the SentARA prompt on Accuracy. This suggests that, if the overriding objective is to achieve the target difficulty level, then the use of sentence ARA alone could be worth considering.

Although the addition of linguistic features (SentARA+Span+Ling) further increased BERTScore, it did not help improve the quality of text modification on the other metrics. This suggests that the LLM may have difficulty interpreting the linguistic features and their implications.

6.3.4 Effect of Text ARA

Initial modification. As discussed in Section 6.3.1,

Prompt method	Task	D-SARI	SARI	BLEU	BertScore	Accuracy	MAE ↓
SentARA+Span	Simp.	**22.84**	46.30	45.21	0.892	0.440	8.42
	Comp.	19.91	**52.47**	**59.34**	**0.916**	**0.777**	**7.06**
SentARA+Span	Simp.	**25.42**	55.81	**59.74**	**0.928**	**0.685**	7.14
(intermediate only)	Comp.	20.07	**60.97**	58.32	0.913	0.675	**6.14**

Table 7: Breakdown of text modification performance into simplification (Simp.) and complexification (Comp.), based on (top) all input texts; (bottom) intermediate input texts only (↓ means smaller is better)

the Text ARA Model has a much higher accuracy in estimating the difficulty level of a text than the use of FRES. For text modification, it was thus effective in determining whether the input text is already at the target difficulty level, or requires modification. Only 1.13% of the input texts underwent unnecessary modification; conversely, only 0.50% of the input texts failed to undergo modification, due to incorrect estimation from this model. The corresponding percentages would have been 33.92% and 24.95%, respectively, if FRES were used for this purpose.

Iterative modifications. In terms of the content of the modified texts, an additional round of modification (SentARA+Span$^{\times 2}$) further improved the BLEU and D-SARI scores. A third iteration (SentARA+Span$^{\times 3}$), however, led to a slight decline in performance, likely because repeated modifications may exacerbate the biases inherent in LLMs (Gallegos et al., 2024; Yu et al., 2024). In terms of text complexity, while the Accuracy improved as expected, a second iteration had no effect on MAE (7.74) and a third one led to negative impact (7.80).

6.3.5 Simplification vs. complexification

To analyze the differences between text simplification and complexification, we compare the performance of the best prompt (SentARA+Span) on these two tasks. As shown in the top of Table 7, complexification appears to be an easier task than simplification, offering better performance on all but one metric (D-SARI). Complexification often requires inserting new content at appropriate places in a text, which could be more challenging and subjective than removing existing content in simplification. This may explain its lower score for D-SARI, which puts more emphasis on the quality of document-level organization.

A potential confounding factor is the length of the input text. The input texts that required simplification were on average longer, since they were originally at the intermediate and advanced levels; those that required complexification were shorter, since they were taken from the intermediate and elementary levels. To avoid this bias, the bottom of Table 7 considers only the input texts at the intermediate level. Simplification now offers better performance in terms of BLEU and BERTScore. This may be due to the wider array of choices when selecting more complex words or sentence structures, in comparison to selecting simpler ones. In terms of complexity, the gap between the two tasks is narrower for both Accuracy and MAE, but more research is needed to explain the difference.

7 Human evaluation

7.1 Evaluation metrics

The quality of text modification was evaluated by two human judges, a master and a PhD student in Linguistics. Similar to previous schemes (Alva-Manchego et al., 2020a; Yang et al., 2023), Fluency and Meaning were annotated on a 5-point Likert scale (1=Strongly disagree, 5=Strongly agree):

- **Fluency**: The output text is fluent and free of grammatical errors.
- **Meaning**: The output text adequately preserves the meaning of the source text.

To accommodate both simplification and complexification, Complexity was scored from -5 to +5:

- **Complexity**: A score of +5 means the output text is much more complex and harder to understand than the source input; 0 means they are comparable in complexity; and -5 means the output text is much easier.

7.2 Evaluation set-up

Two source texts were randomly selected at each difficulty level (`ele`, `int` and `adv`). Each of these six texts was paired with four modified versions, namely, its modified version at the two other levels as produced by the Vanilla (zero-shot) and

Modification type	Source->target level	Fluency		Meaning		Complexity	
		Zero-shot	Proposed	Zero-shot	Proposed	Zero-shot	Proposed
Text Simp.	Overall	4.00	**4.17**	3.17	**3.50**	-3.50	-3.50
	adv->int	4.00	4.00	3.50	**4.00**	**-3.00**	-2.50
	adv->ele	4.00	4.00	3.00	3.00	-3.50	**-4.00**
	int->ele	4.00	**4.50**	3.00	**3.50**	-4.00	-4.00
Text Comp.	Overall	**3.50**	3.17	4.00	**4.17**	+3.00	**+3.50**
	int->adv	**3.50**	3.00	4.50	**5.00**	+3.50	**+4.00**
	ele->int	**4.00**	3.00	3.00	**3.50**	**+3.00**	+2.50
	ele->adv	3.00	**3.50**	**4.50**	4.00	+2.50	**+4.00**

Table 8: Human evaluation scores on Fluency, Meaning and Complexity (Section 7)

SentARA+Span$^{\times 2}$ models. The two human judges independently scored these 24 text pairs on Complexity, Fluency and Meaning (Section 7.1).

7.3 Results

The human evaluation results are shown in Table 8. The two judges achieved a Cohen's kappa (Cohen, 1968) of 0.67 for Fluency, 0.88 for Meaning, and 0.85 for Complexity, all at/above a substantial level of agreement.

Meaning. The proposed model (SentARA+Span$^{\times 2}$) achieved a higher Meaning score than the zero-shot baseline in four of the six settings. It was slightly outperformed in `adv->ele` and tied at `ele->adv`. As both of these settings required greater modification (distance of two levels rather than one), the proposed model was more likely to make changes that altered the original meaning.

Fluency. The proposed model performed better in simplification, but it slightly underperformed in complexification, particularly with two-level modifications. While the auxiliary models help the proposed model in modifying the content and difficulty, they do not necessarily improve the fluency of the output text, which sometimes contains awkward phrasing.

Complexity. Both models were capable of revising the input text towards the required complexity level, obtaining positive Complexity scores when the level of the input text was lower than the target (complexification needed), and negative scores when its level was higher than the target (simplification needed). When simplifying `adv` texts, the proposed model succeeded in differentiating between the target levels `ele` and `int`, reducing text complexity to a much greater degree for the former (-4.00) than the latter (-2.50). A smaller difference was observed for the zero-shot baseline (-3.50 vs. -3.00). When complexifying `ele` texts, the proposed model was again able to make a sharper distinction, producing a more sophisticated output for the `adv` target grade (+4.00) than for `int` (+2.50). The zero-shot model failed to do so and produced an `adv` output (+2.50) that is easier than the `int` output (+3.00).

8 Conclusion

Human editors often need to tailor a text for readers at different proficiency levels. Readability-controlled text modification aims to rewrite an input text so that it reaches a target level of difficulty. Depending on the difficulty of the input text, it may need to be simplified or complexified. This paper has presented the first study on LLM-based readability-controlled text modification that leverages ARA and prediction of target spans, i.e. the parts of the input text that require editing.

We trained ARA models that can predict the difficulty level of a sentence or a text, and taggers that predict whether each word should be revised. The information from these auxiliary models are then incorporated into the prompt for the LLM. Experimental results on the Newsela corpus showed that both the ARA models and the target span prediction model improved the quality of the modified text. In future work, we plan to evaluate this approach on other text genres, and investigate whether fine-tuning an LLM on text modification data can lead to further performance gains.

Acknowledgments

We gratefully acknowledge support from the eLearning Ancillary Facilities Programme of the Quality Education Fund (Project "Knowledge Overlord - A Self-sustaining AI Game-based Online Platform to Enhance Student's Literacy Ability and

21st Century Skills"); and from the Language Fund of the Standing Committee on Language Education and Research (project EDB(LE)/P&R/EL/203/14).

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*, page 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, Online. Association for Computational Linguistics.

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.

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

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*.

J. De Belder and M. F. Moens. 2010. Text Simplification for Children. In *Proc. SIGIR Workshop on Accessible Search Systems*.

Leonid Berov and Kai Standvoss. 2018. Discourse embellishment using a deep encoder-decoder network. In *Proc. 3rd Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2018)*.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi. 2018. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2690–2699.

Xiaobin Chen and Detmar Meurers. 2019. Linking text readability and learner proficiency using linguistic complexity feature vector distance. *Computer Assisted Language Learning*, 32(4):418–447.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V Le. 2019. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Alison Chi, Li-Kuang Chen, Yi-Chen Chang, Shu-Hui Lee, and Jason S. Chang. 2023. Learning to Paraphrase Sentences to Different Complexity Levels. *Transactions of the Association for Computational Linguistics*, 11:1332–1354.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339.

M. Färber, P. Aghdam, K. Im, M. Tawfelis, and H. Ghoshal. 2025. SimplifyMyText: An LLM-Based System for Inclusive Plain Language Text Simplification. *Advances in Information Retrieval. ECIR 2025. Lecture Notes in Computer Science*, 15575.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable prediction of text complexity: The missing preliminaries for text simplification. 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 1086–1097.

Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153.

Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting Proper Lexical Paraphrase for Children. In *Proc. 25th Conference on Computational Linguistics and Speech Processing (ROCLING)*, pages 59–73.

Peter J. Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas for Navy enlisted personnel. In *Research Branch Report 8–75*. Chief of Naval Technical Training: Naval Air Station Memphis.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Bruce W. Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on Text Readability Assessment: A Transformer Meets Handcrafted Linguistic Features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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 *Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, page 7871–7880.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Fengkai Liu, Yishi Jiang, Chun Lai, and Tan Jin. 2024. Teacher engagement with automated text simplification for differentiated instruction.

Fengkai Liu, Tan Jin, and John S. Y. Lee. 2025. Automatic readability assessment for sentences: neural, hybrid and large language models. *Language Resources and Evaluation*, 59:2265–2296.

Fengkai Liu and John S. Y. Lee. 2023. Hybrid Models for Sentence Readability Assessment. In *Proc. 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, page 448–454.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Dawei Lu, Xinying Qiu, and Yi Cai. 2020. Sentence-level readability assessment for l2 chinese learning. In *Chinese Lexical Semantics: 20th Workshop, CLSW 2019, Beijing, China, June 28–30, 2019, Revised Selected Papers 20*, pages 381–392. Springer.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners' oral narratives. *The Modern Language Journal*, 96(2):190–208.

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.

Matej Martinc, Senja Pollak, Marko, and Robnik-Šikonja. 2021. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179.

Kaijie Mo and Renfen Hu. 2024. Expertease: A multi-agent framework for grade-specific document simplification with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9080–9099.

Subhajit Naskar, Soumya Saha, and Sreeparna Mukherjee. 2019. Text embellishment using attention based encoder-decoder model. In *Proc. 4th Workshop on Computational Creativity in Language Generation*, page 28–38.

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.

Gustavo H Paetzold and Lucia Specia. 2016. Plumberr: An automatic error identification framework for lexical simplification. In *Proceedings of the first international workshop on Quality Assessment for Text Simplification (QATS)*, pages 1–9.

Gustavo H. Paetzold and Lucia Specia. 2017. Lexical Simplification with Neural Ranking. In *Proc. EACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Daniele Schicchi, Giovanni Pilato, and Giosué Lo Bosco. 2020. Deep neural attention-based model for the evaluation of italian sentences complexity. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 253–256. IEEE.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *LREC*, pages 1583–1590.

Advaith Siddharthan. 2002. An Architecture for a Text Simplification System. In *Proc. Language Engineering Conference (LEC)*.

Sanja Štajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic assessment of absolute sentence complexity. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI*, volume 17, pages 4096–4102.

Steinþór Steingrímsson, Hrafn Loftsson, and Andy Way. 2023. SentAlign: Accurate and Scalable Sentence Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Singapore, Singapore. Association for Computational Linguistics.

Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. Document-level text simplification: Dataset, criteria and baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7997–8013.

Sean Trott and Pamela Rivière. 2024. Measuring and Modifying the Readability of English Texts with GPT-4. In *Proc. 3rd Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, page 126–134.

Sanja Štajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic Assessment of Absolute Sentence Complexity. In *Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. In *Proc. Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, page 38–45.

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 Kajiwara, 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 (Volume 2: Short Papers)*, pages 398–404.

Shiping Yang, Renliang Sun, and Xiaojun Wan. 2023. A new dataset and empirical study for sentence simplification in Chinese. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8306–8321, Toronto, Canada. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

A.1 System prompt

You are a professional expert in modifying texts into a target difficulty level. The Flesch Reading-Ease Score (FRES) measures the readability of the text: The averaged FRES for elementary texts is `<aver_ele_FRES>`. The averaged FRES for intermediate texts is `<aver_int_FRES>`. The averaged FRES for advanced texts is `<aver_adv_FRES>`.

Table 9: System prompt

A.2 Linguistic Features

Rank	Feature	Correl.
1	Root TTR	0.684
2	Corrected TTR	0.684
3	Number of Different Words	0.628
4	Uber Index	0.610
5	LS2	0.546

Table 10: The five lexical features that are most correlated with readability levels of the texts in our dataset.