Teaching the Pre-trained Model to Generate Simple Texts for Text Simplification

[1]Renliang Sun, [2]Wei Xu, [1]Xiaojun Wan
[1]Wangxuan Institute of Computer Technology, Peking University
[1]Center for Data Science, Peking University
[1]The MOE Key Laboratory of Computational Linguistics, Peking University
[2]School of Interactive Computing, Georgia Institute of Technology
sunrenliang@stu.pku.edu.cn wei.xu@cc.gatech.edu
wanxiaojun@pku.edu.cn

Abstract

Randomly masking text spans in ordinary texts in the pre-training stage hardly allows models to acquire the ability to generate simple texts. It can hurt the performance of pre-trained models on text simplification tasks. In this paper, we propose a new continued pre-training strategy to teach the pre-trained model to generate simple texts. We continue pre-training BART, a representative model, to obtain SimpleBART. It consistently and significantly improves the results on lexical simplification, sentence simplification, and document-level simplification tasks over BART. At the end, we compare SimpleBART with several representative large language models (LLMs).

1 Introduction

Text simplification (TS) is a task in the field of natural language generation. It aims at rewriting a complex text into simple text while keeping the primary meaning intact (Laban et al., 2021).

Recently, several works have leveraged pre-trained models for TS (Omelianchuk et al., 2021; Devaraj et al., 2022). However, problems arise when pre-trained models are applied to TS directly. In the pre-training stage, the model hardly acquires the ability to generate simple texts. The improvement of results on simplification tasks relies almost on the fine-tuning stage. It can hurt the performance of pre-trained models, especially for low-resource sub-tasks like lexical simplification. One reason for this shortcoming is the pre-training strategy. It randomly masks text spans in ordinary texts, teaching the model to generate ordinary texts rather than simple texts.

We are committed to adapting the pre-trained model to TS in this paper. The pre-trained model has gained the ability to generate ordinary texts, and it is costly to start pre-training from scratch. Therefore, we focus on the continued pre-training strategy (Gururangan et al., 2020). We first aim to continue pre-training on simple texts because it contains plenty of simple words. In TS, simple texts are derived almost from SimpleWiki (Zhang and Lapata, 2017) and Newsela (Xu et al., 2015). We identify simple text spans in simple texts and dynamically replace them with <mask> tokens. Then, the pre-trained model will learn by reconstructing simple words. Meanwhile, we expect the pre-trained model to learn from ordinary texts. We use a dictionary to replace complex words in ordinary texts with simple words. We also ensure the quality of the replaced sentences.

Based on BART (Lewis et al., 2020), we continue pre-training to teach it to generate simple texts and obtain SimpleBART. We then conduct experiments on three main tasks of TS: sentence simplification, lexical simplification, and document-level simplification. SimpleBART achieves consistent and noticeable improvements across several datasets on all three tasks over BART and several other baselines. The results illustrate that our proposed strategy helps the pre-trained model to gain the ability to generate simple texts.

To summarize, our contributions include: (1) We propose a new continued pre-training strategy to teach the pre-trained model to generate simple texts. (2) We continue pre-training BART, a representative seq2seq model, to obtain SimpleBART. It can be used for several simplification tasks and achieve consistent performance improvement. Code and SimpleBART will be released at https://github.com/RLSNLP/SimpleBART.

2 Methodology

As illustrated in Figure 1, our strategy is divided into two parts: learning dynamically to reconstruct simple words from simple texts and from ordinary texts where complex words are replaced with simple ones.

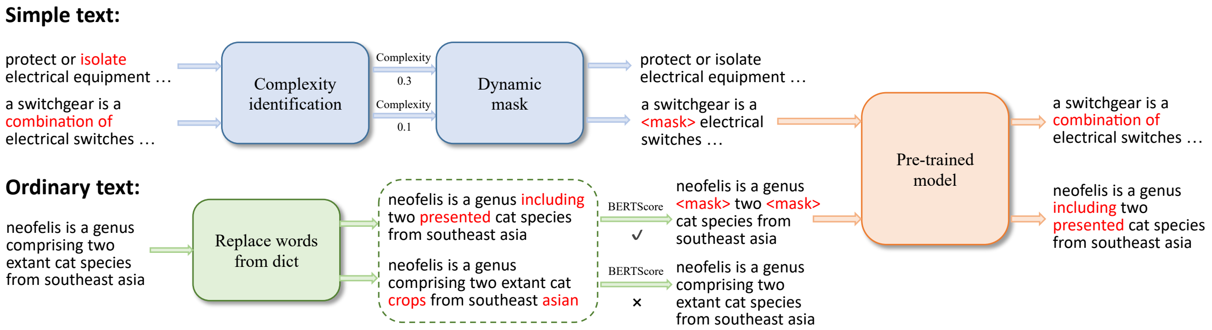
Simple text:
protect or isolate electrical equipment …
a switchgear is a combination of electrical switches …
Complexity identification
Complexity 0.3
Complexity 0.1
Dynamic mask
protect or isolate electrical equipment …
a switchgear is a <mask> electrical switches …
Pre-trained model
a switchgear is a combination of electrical switches …
Ordinary text:
neofelis is a genus comprising two extant cat species from southeast asia
Replace words from dict
neofelis is a genus including two presented cat species from southeast asia
neofelis is a genus comprising two extant cat crops from southeast asian
BERTScore
✓
BERTScore
×
neofelis is a genus <mask> two <mask> cat species from southeast asia
neofelis is a genus comprising two extant cat species from southeast asia
neofelis is a genus including two presented cat species from southeast asia

Figure 1: Overview of our continued pre-training strategy to teach the pre-trained model to generate simple texts.

2.1 Masking Simple Words in Simple Texts

We need to identify the simple words in simple texts at first. We take advantage of the DeepBlueAI model (Pan et al., 2021) that achieves state-of-the-art results on the lexical complexity prediction task (Shardlow et al., 2021). A text span of length n consists of n words. The input to the DeepBlueAI model is a text span and the output is a complex value between 0 and 1. The closer this value is to 0, the simpler the text span.

Unlike the previous constant mask probability, in our strategy, the simpler a text span is, the higher its probability of being masked. This means that the mask probability is dynamic. We also set a complexity threshold of T. If the complexity c of a text span exceeds T, we will not mask this span. In our experiments, we set T to 0.25 as an empirical value. Following Lewis et al. (2020), we set the max mask probability to 0.15, and the length of a text span obeys a Poisson distribution (λ = 3). Finally, the mask probability m is calculated as:

$$m = \begin{cases} 0.15 \times (1 - \frac{1}{T} \cdot c), & c \leq T \\ 0, & c > T \end{cases} \tag{1}$$

The function to mask the text span is denoted as $g(\cdot)$. Given a sentence x, the pre-trained model will learn to reconstruct x from the masked sentence:

$$l(x) = -logP(x|g(x)) \tag{2}$$

2.2 Replacing Complex Words in Ordinary Texts

We also expect the pre-trained model to learn helpful information from ordinary texts. However, ordinary texts contain more complex words than simple ones, making the pre-trained model learn to reconstruct simple words much less frequently. We introduce the dictionary SimplePPDB++ (Maddela and Xu, 2018) to address this issue. It contains millions of paraphrase rules with readability scores. Therefore, we can replace the complex words in ordinary texts with simple words. Then, the pre-trained model will learn to reconstruct these simple words as in Eq.(2).

Nevertheless, a word may have different meanings in different sentences. Using a dictionary to replace complex words may change the meaning of the original sentence. Therefore, we use BERTScore (Zhang et al., 2019) to calculate the similarity between the original and replaced sentences to avoid this problem. We will discard the replaced sentences if the calculated BERTScore is lower than a similarity threshold. In our experiments, the similarity threshold is set to 0.95 as an empirical value.

3 Experimental Settings

3.1 Continued Pre-training

We select the BART-Large model to continue pre-training. It is a representative seq2seq model suitable for three main simplification tasks. We follow the task-adaptive pre-training method (Gururangan et al., 2020) and continue pre-training on the training set of the corresponding simplification task, ensuring that the continued pre-training texts have no intersection with the test set. We refer to the pre-trained models obtained by our strategy collectively as SimpleBART.

3.2 Simplification Tasks

We select three representative tasks for experiments: sentence simplification, document-level simplification, and lexical simplification. For sentence simplification, we conduct experiments on Wikiauto (Jiang et al., 2020) and Newsela (Xu et al., 2015). Wikiauto is only a training set, so we use

Turkcorpus (Xu et al., 2016) as its validation and test set. Following Sun et al. (2023), we use SARI (Xu et al., 2016) and BERTScore (Zhang et al., 2019) as the evaluation metrics. BLEU and FKGL have been proven to be unsuitable for evaluating simplification (Sulem et al., 2018; Tanprasert and Kauchak, 2021). For document-level simplification, we conduct experiments on the D-Wikipedia dataset (Sun et al., 2021). We use D-SARI (Sun et al., 2021) as the evaluation metric. For lexical simplification, we conduct experiments on LexMTurk (Horn et al., 2014) and BenchLS (Paetzold and Specia, 2016). We use precision, recall, and F1 score as the evaluation metrics. For more hyper-parameter setting details, please refer to Appendix B.

4 Results

4.1 Sentence Simplification

To demonstrate the advantages of our strategy, we develop BART-CP for a fair comparison. It continues pre-training with the same number of steps on the same data using the previous pre-training strategy from Lewis et al. (2020). In the continued pre-training stage, text spans are masked randomly.

Turkcorpus	SARI↑	Keep	Del	Add	BS↑
EditNTS	37.9	67.3	43.1	3.4	0.950
T5	37.8	**73.5**	35.6	4.2	**0.982**
ControlTS	**40.4**	70.4	44.5	6.2	0.959
BART	38.3	65.4	44.0	5.6	0.973
BART-CP	38.6	64.6	45.9	5.4	0.967
SimpleBART	39.5	64.6	**47.2**	**6.6**	0.972

Newsela	SARI↑	Keep	Del	Add	BS↑
EditNTS	37.1	34.9	74.8	1.6	0.897
T5	36.0	**41.8**	61.9	4.4	0.905
ControlTS	39.7	37.6	77.3	4.1	0.894
BART	40.1	40.5	73.8	6.2	0.904
BART-CP	40.3	41.7	72.6	**6.9**	**0.908**
SimpleBART	**41.6**	40.5	**77.4**	**6.9**	0.902

Table 1: Results on the Turkcorpus test set and the Newsela test set. We use **bold** to indicate the best result.

We choose EditNTS (Dong et al., 2019), T5-base (Raffel et al., 2020), and ControlTS (Maddela et al., 2021) as baselines. T5-base is close to SimpleBART in size. ControlTS achieves the state-of-the-art result on the Newsela dataset. Following Alva-Manchego et al. (2021), BERTScore$_{precision}$ (BS) is also reported. From Table 1, the BS scores of all outputs are high enough, which means that the outputs are of high quality. According to SARI, the most important automatic evaluation metric for sentence simplification, SimpleBART improves SARI values over BART by 1.2 points and 1.5 points, respectively. Overall, it achieves comparable results to the advanced model for the sentence simplification task. We also notice that SimpleBART outperforms BART-CP, demonstrating the effectiveness of our proposed strategy. The example outputs are given in Appendix D.

4.2 Lexical Simplification

We focus on generating suitable words using the pre-trained model, which is a critical step in lexical simplification. We follow Qiang et al. (2020a) and let the pre-trained models generate several candidate words. BenchLS and LexMTurk are just two test sets, so we continue pre-training on the Wiki-auto training set. We choose Paetzold-NE (Paetzold and Specia, 2017a) and LSBert (Qiang et al., 2020b) as two baselines. LSBert achieves the state-of-the-art result in this task.

BenchLS	F1↑	Precision	Recall
Paetzold-NE	23.6	27.0	20.9
LSBert	**28.1**	24.4	**33.1**
BART	19.2	19.6	18.9
BART-CP	25.8	26.0	25.7
SimpleBART	27.8	**28.0**	27.6

LexMTurk	F1↑	Precision	Recall
Paetzold-NE	19.5	**31.0**	14.2
LSBert	26.8	30.6	23.8
BART	18.8	19.2	18.3
BART-CP	26.9	27.2	26.6
SimpleBART	**28.5**	28.7	**28.2**

Table 2: Results on the BenchLS test set and the LexMTurk test set.

As shown in Table 2, SimpleBART improves the F1 scores over BART by 8.6 points and 9.7 points, respectively. It achieves comparable results to LSBert. The results also demonstrate that BART needs to gain the ability to generate simple words and the importance of introducing continued pre-training when training data is scarce.

4.3 Document-level Simplification

SimpleBART also performs well on the document-level simplification task. We choose BertSumextabs (Liu and Lapata, 2019), which achieves the state-of-the-art result on this task as a baseline. Compared with BART, SimpleBART improves the

D-Wikipedia	D-SARI↑	D_{keep}	D_{del}	D_{add}
BertSumextabs	39.88	35.71	**72.06**	11.87
BART	39.84	35.87	70.26	13.40
BART-CP	40.13	36.21	71.54	12.64
SimpleBART	**41.64**	**37.91**	71.96	**15.04**

Table 3: Results on the D-Wikipedia test set

D-SARI value by 1.8 points, making it the new state-of-the-art result.

5 Analysis

5.1 Human Evaluation

We hire three workers to conduct a human evaluation of the 100 randomly selected outputs of the sentence simplification task. Following Dong et al. (2019), workers rate on simplicity (Simp), fluency (Flu), and adequacy (Ade) on a 5-point Likert scale. Following Xu et al. (2016), we use simplicity gain (S+) to demonstrate how many word-level simplifications occur in sentence simplification.

	Simp↑	Flu↑	Ade↑	S+↑
EditNTS	3.30*	4.65*	3.56*	0.14*
T5	3.16*	**4.91***	**4.47***	0.25*
ControlTS	3.39*	4.67*	4.26*	**0.60**
BART	3.22*	4.80	4.31*	0.34*
BART-CP	3.45*	4.68*	3.95	0.37*
SimpleBART	**3.62**	4.82	4.01	0.55
Reference	3.74	4.85	4.03	0.93*

Table 4: Results of the human evaluation. The results significantly different from those of SimpleBART are marked as * according to the student t-test with p<0.05.

Table 4 shows that SimpleBART achieves the highest Simp score among all the simplification models, close to that of the reference. SimpleBART also significantly makes more word-level simplifications compared to BART and BART-CP.

5.2 Domain Adaptation

Continued pre-training using our strategy on task-related data can improve the results. However, we still want to know if continued pre-training on more data from the same domain and different domains will improve the results. We design the following experiments. 1) Exp1: We continue pre-training on more sentences from Wikipedia and SimpleWiki, except those contained in the Wikiauto dataset. 2) Exp2: We continue pre-training on more sentences in the Newsela corpus, except those contained in the Newsela dataset. The sizes of the above texts used for continued pre-training are roughly five times larger than the simplification training set. 3) Exp3: We continue pre-training on the Newsela training set. 4) Exp4: We continue pre-training on the Wikiauto training set.

	SARI↑	Keep	Del	Add	BS↑
Exp1	38.9	64.9	45.7	6.0	0.968
Exp2	41.1	39.5	77.4	6.5	0.900
Exp3	38.0	39.2	69.7	5.0	0.975
Exp4	39.6	42.1	71.1	5.7	0.907

Table 5: Results of domain adaptation experiments. For Exp1 and Exp3, we fine-tune on Wikiauto and test on Turkcorpus. For Exp2 and Exp4, we fine-tune and test on the Newsela dataset.

From the results of Exp1 and Exp2 in Table 5, continued pre-training on more texts from the same domain can still enhance the simplification results. Compared to BART in Table 1, the SARI values improve by 0.6 and 1 point, respectively. From the results of Exp3 and Exp4, continued pre-training on more texts in a different domain can instead harm the results. Compared to BART, the SARI values decrease by 0.3 and 0.5 points, respectively. Thus, we suggest that future researchers use texts within the same domain (e.g., Wikiauto and Wikipedia) for continued pre-training in text simplification.

5.3 Generating Complex Texts

There are numerous studies dedicated to simplifying complex texts. Nevertheless, none has attempted to rewrite simple texts into complex ones. We make such an interesting attempt. We have changed our strategy to mask complex words and name the obtained model ComplexBART. When fine-tuning and testing on the Newsela dataset, we use simple texts as input and complex texts as reference.

	SARI↑	Keep	Del	Add	BS↑
BART	35.7	53.2	50.5	3.3	0.901
ComplexBART	37.2	52.9	55.4	3.4	0.900

Table 6: Results of generating complex texts.

From Table 6, ComplexBART improves the SARI value by 1.5 points over the BART model, indicating that the modified strategy can help the pre-trained model learn to generate complex texts. Thus, ComplexBART can serve as a better baseline for generating complex texts in the future.

6 Comparing SimpleBART with Large Language Models

Large language models (LLMs) have received widespread attention from researchers recently and have achieved state-of-the-art results on many natural language generation tasks. In this section, we select several representative large models to conduct experiments on text simplification and compare them with SimpleBART. We hope these results can serve as baselines for future research.

We choose those LLMs that provide API or model files to ensure reproducibility. We choose GPT-3.5-Turbo-0301[1], FLAN-T5-XL (Chung et al., 2022), and LLaMA-7B (Touvron et al., 2023) as LLM baselines and use zero-shot generation. Then, we follow the implementation[2] and fine-tune FLAN-T5-base as another baseline. We collect the training sets of Wikiauto, Newsela, and D-Wikipedia and conduct instruction fine-tuning.

6.1 Comparison and Analysis

The comparison of SimpleBART results with those of the LLMs is shown in Tables 7, 8, and 9.

For the sentence-level simplification task, LLaMA and FLAN-T5-XL seem unable to understand the prompt for simplifying sentences, and they are inclined to repeat the original text. However, FLAN-T5-base, only 10% of the parameters of the above two models, performs better. It illustrates fine-tuning phase can improve performance when the model is not super large. It may be a little strange that GPT-3.5 performs worse than SimpleBART. We find that with the zero-shot setting, GPT-3.5 may not know the "degree of simplification" we want. It makes many reasonable changes to the original text, but it also keeps some of the complex parts of the original text.

For the document-level simplification task, LLaMA over-repeats sentences from the original article, and the generated text is difficult to read. The shortcomings of GPT-3.5 are similar to those of the sentence-level simplification task. Besides, limited by the number of API accesses per minute of OpenAI, we only select 1000 original documents for simplification, which takes nearly five hours.

For the lexical simplification task, neither the LLaMA nor the FLAN-T5 model could understand the instruction to replace complex words with simple words. However, GPT-3.5 outperforms the other models substantially. We also find that GPT-3.5 makes many sensible substitutions not included in the reference, such as replacing "acquired"with "earned". Such results illustrate that LLMs are dominant for this task.

Turkcorpus	SARI↑	Keep	Del	Add	BS↑
GPT-3.5	32.4	43.4	43.4	10.4	0.896
FLAN-T5	31.5	64.1	29.6	1.0	0.892
LLaMA	29.3	69.3	16.3	2.3	0.873
FLAN-T5 (Fine-tuned)	36.5	74.4	31.3	3.8	0.901
SimpleBART	39.5	64.6	47.2	6.6	0.972

Newsela	SARI↑	Keep	Del	Add	BS↑
GPT-3.5	38.7	32.5	78.1	5.3	0.897
FLAN-T5	32.2	29.7	65.7	1.3	0.891
LLaMA	19.9	35.8	23.2	0.8	0.822
FLAN-T5 (Fine-tuned)	29.9	40.3	46.7	2.7	0.902
SimpleBART	41.6	40.5	77.4	6.9	0.902

Table 7: Comparison on the Turkcorpus test set and the Newsela test set.

D-Wikipedia	D-SARI↑	Keep	Del	Add
GPT-3.5	26.68	18.45	59.36	2.25
FLAN-T5	26.77	15.07	64.83	0.40
LLaMA	/	/	/	/
FLAN-T5 (Fine-tuned)	33.22	25.08	67.50	7.09
SimpleBART	41.64	37.91	71.96	15.04

Table 8: Comparison on the D-Wikipedia test set.

BenchLS	F1↑	Precision	Recall
GPT-3.5	36.6	36.6	36.6
SimpleBART	27.8	28.0	27.6

LexMTurk	F1↑	Precision	Recall
GPT-3.5	31.4	31.5	31.4
SimpleBART	28.5	28.7	28.2

Table 9: Comparison on the BenchLS test set and the LexMTurk test set.

7 Conclusion

In this paper, we are committed to adapting the pre-trained model to text simplification. We propose a new pre-training strategy to allow the pre-trained model to learn to generate simple texts. The adapted pre-trained model improves the results on various simplification tasks.

[1] https://openai.com/blog/chatgpt

[2] https://github.com/philschmid/deep-learning-pytorch-huggingface/blob/main/training/deepseed-flan-t5-summarization.ipynb

Limitations

The limitation of our method comes from the requirement to identify simple words in simple texts in Section 2.1. The DeepBlueAI we have used is a deep model, meaning it takes much time when inference. In our experiment, it takes 362.78 seconds to identify simple words from 10,000 sentences with an average length of 8.12. We expect that there will be methods with higher identification accuracy and higher inference speed in the future.

Due to page limitations, we have placed the related work in Appendix A and the ablation experiments in Appendix C.

Due to time constraints, we do not perform a human evaluation of the output of LLMs. We hope to conduct a more comprehensive evaluation of the performance of LLMs in the future.

Ethics Statement

The texts we have used for continued pre-training come from Wikipedia dumps and the Newsela Corpus. Using Wikipedia dumps requires following the CC-BY-SA license and GFDL. Using Newsela Corpus requires authorization, and we have received it.

This paper contains a human evaluation. We hire three experienced workers to perform it. In the recruiting process, we follow a first-come, first-served order. We pay much more than the local minimum hourly rate.

Acknowledgements

We thank Mounica Maddela for her help on the ControlTS baseline. This work is supported by National Key R&D Program of China (2021YFF0901502), National Science Foundation of China (No. 62161160339), State Key Laboratory of Media Convergence Production Technology and Systems and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). Wei Xu and Xiaojun Wan are the corresponding authors.

References

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

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Ashwin Devaraj, William Sheffield, Byron C Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345.

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.

Sian Gooding and Ekaterina Kochmar. 2019. Recursive context-aware lexical simplification. 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 4853–4863.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463.

Junjie Hu, Hiroaki Hayashi, Kyunghyun Cho, and Graham Neubig. 2022. Deep: Denoising entity pretraining for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1753–1766.

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.

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.

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.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740.

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 and Wei Xu. 2018. A word-complexity lexicon and a neural readability ranking model for lexical simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Gustavo Paetzold. 2021. Utfpr at semeval-2021 task 1: Complexity prediction by combining bert vectors and classic features. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 617–622.

Gustavo Paetzold and Lucia Specia. 2016. Benchmarking lexical simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3074–3080.

Gustavo Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40.

Gustavo H Paetzold and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020a. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8649–8656.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020b. Lsbert: A simple framework for lexical simplification. *arXiv preprint arXiv:2006.14939*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex—a new corpus for lexical complexity prediction from likert scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62.

Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021. Semeval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Bleu is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744.

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.

Renliang Sun, Zhixian Yang, and Xiaojun Wan. 2023. Exploiting summarization data to help text simplification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 39–51.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

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.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022. Dict-bert: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918.

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

A Related Work

A.1 Text Simplification

Text simplification contains sentence simplification, document-level simplification, and lexical simplification. Sentence simplification is rewriting a complex sentence into a more straightforward and semantically identical sentence (Alva-Manchego et al., 2020). Document-level simplification is rewriting an original complex article into a simple article (Sun et al., 2021). Information not relevant to the central meaning can be removed to improve readability. Lexical simplification is to replace complex words in a sentence with more straightforward but identical meaning words (Paetzold and Specia, 2017b). It is usually framed as a pipeline consisting of generating multiple candidate words and developing rules to select the most appropriate word from candidate words.

A.2 Lexical Complexity Prediction

The lexical complexity prediction (LCP) task is to assign a value from a continuous scale to represent the complexity of a word (Shardlow et al., 2020). Given a text and a text span in this text, the model will predict the complexity of this text span. Many studies have been devoted to improving the accuracy of model predictions (Gooding and Kochmar, 2019; Paetzold, 2021). On the latest LCP 2021 task (Shardlow et al., 2021), the DeepBlueAI model (Pan et al., 2021) achieves state-of-the-art results.

A.3 Adapting Pre-trained models

Pre-trained models have been widely used in natural language processing in recent years. However, Gururangan et al. (2020) observe the gap between the language model pre-training domain and the data distribution of the downstream task. Since then, researchers have focused on how to adapt pre-trained models to downstream tasks. They have designed new methods for different tasks. Downstream tasks like machine translation (Hu et al., 2022), sentiment analysis (Gu et al., 2020), and many understanding tasks (Yu et al., 2022) can benefit from the adapted pre-trained models.

B Training Parameters

We use the Huggingface transformers (Wolf et al., 2020) to conduct sentence and lexical simplification experiments. For document-level simplification, we follow Sun et al. (2021) and use Fairseq (Ott et al., 2019) to conduct the experiments. We choose the model that performs best on the validation set for testing. The specific parameter settings for each task are shown in Tables 10, 11, and 12. A detailed description of the dataset sizes is given in Table 13.

Here are the sources of the automatic evaluation methods we use: SARI (`https://github.com/mounicam/BiSECT/tree/main/metrics`), BERTScore (`https://github.com/Tiiiger/bert_score`), and D-SARI (`https://github.com/RLSNLP/Document-level-text-simplification`).

Parameter	Value	Parameter	Value
epochs	10	max source length	128
batchsize	64	max target length	128
optimizer	Adam	dropout	0.1
learning rate	5e-5	weight decay	0
warm up steps	5000	seed	42

Table 10: Training parameters for sentence simplification.

Parameter	Value	Parameter	Value
epochs	10	max source length	128
batchsize	64	max target length	128
optimizer	Adam	dropout	0.1
learning rate	5e-5	weight decay	0
warm up steps	5000	seed	42

Table 11: Training parameters for lexical simplification.

Parameter	Value	Parameter	Value
max update steps	1e5	max source length	512
max tokens	2048	max target length	512
optimizer	Adam	dropout	0.1
learning rate	1e-4	weight decay	1e-4
warm up steps	2000	seed	42

Table 12: Training parameters for document-level simplification.

Dataset	train	dev	test
Sentence simplification			
Wikiauto	488K	\	\
Turkcorpus	\	2000	359
Newsela	94K	1129	1077
Lexical simplification			
BenchLS	\	\	929
LexMTurk	\	\	500
Document-level simplification			
D-Wikipedia	133K	3000	8000

Table 13: Sizes of the datasets used in experiments.

	SARI↑	Keep	Del	Add	BS↑
BART	40.1	40.5	73.8	6.2	0.904
BART-S	40.9	41.6	74.2	6.9	0.906
BART-T	40.9	40.6	74.9	7.2	0.905
SimpleBART	41.6	40.5	77.4	6.9	0.902

Table 14: Results of ablation experiments on the Newsela dataset of the sentence simplification task.

C Ablation Study

We conduct ablation experiments to explore the different contributions of replacing complex words in ordinary texts (BART-S) and masking simple words in simple texts (BART-T). We continue pre-training and fine-tuning on the Newsela dataset.

From Table 14, both methods in our proposed strategy allow the model to acquire the ability to generate simple words. Their contributions are roughly the same, but the improvement to the SARI value is less than combining them.

D Example Outputs

Original sentence
gary goddard is the founder of gary goddard entertainment .
Reference sentence
gary goddard started gary goddard entertainment .
BART
gary is the founder of gary goddard entertainment .
BART-CP
gary goddard is the founder of gary goddard entertainment .
SimpleBART
gary goddard started a company called gary goddard entertainment .

Table 15: In this sentence simplification example, SimpleBART replaces the phrase “is the founder of" with a simpler phrase “started a company", which is similar to the reference sentence. Both BART and BART-CP do not simplify the original sentence.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section.

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract section.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

2

☑ B1. Did you cite the creators of artifacts you used?
2

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Ethics Statement section.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix B

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix B

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
5

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Ethics Statement section.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Ethics Statement section.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.