International Journal of Asian Language Processing
Vol. 30, No. 1 (2020) 2050001 (15 pages)

DOI: 10.1142/S2717554520500010

Extremely Low-Resource Text Simplification with Pre-trained Transformer Language Model

Takumi Maruyama* and Kazuhide Yamamoto†

Department of Electrical, Electronics and Information Engineering
Nagaoka University of Technology
1603-1, Kamitomioka Nagaoka, Niigata 940-2188, Japan
**maruyama@jnlp.org*
†yamamoto@jnlp.org

Published 30 May 2020

Inspired by machine translation task, recent text simplification approaches regard a task as a monolingual text-to-text generation, and neural machine translation models have significantly improved the performance of simplification tasks. Although such models require a large-scale parallel corpus, such corpora for text simplification are very few in number and smaller in size compared to machine translation task. Therefore, we have attempted to facilitate the training of simplification rewritings using pre-training from a large-scale monolingual corpus such as *Wikipedia* articles. In addition, we propose a translation language model to seamlessly conduct a fine-tuning of text simplification from the pre-training of the language model. The experimental results show that the translation language model substantially outperforms a state-of-the-art model under a low-resource setting. In addition, a pre-trained translation language model with only 3000 supervised examples can achieve a performance comparable to that of the state-of-the-art model using 30,000 supervised examples.

Keywords: Low resource; text simplification; language modeling; transfer learning.

1. Introduction

Automatic text simplification is a task that reduces the complexity of vocabulary and expressions while preserving the meaning of the text. This technique can be used to ensure that numerous text resources are available for a wide range of readers including children, nonnative speakers, and the disabled. Over the years, the number of tourists in Japan has increased. According to a survey conducted by the National Institute for Japanese Language and Linguistics, the number of people who can understand Japanese is more than the number of people who can understand English.[1] Hence, text simplification is one of the important ways to provide information to foreigners. In addition, text simplification can improve the performance of natural language processing tasks including parsing,[2]

*Corresponding author.

summarization,[3,4] semantic role labeling,[5] information extraction,[6] and machine translation.[7,8]

Recent approaches have regarded the simplification process as a monolingual text-to-text generation task.[9–15] Simplification rewritings are trained automatically from examples of original simplified sentence pairs. Neural machine translation has greatly improved the simplification performance compared to previous methods hence requiring a large-scale parallel corpus. However, parallel corpora for text simplification are extremely few in number and smaller in size compared to machine translation tasks. In Japanese, there are no large-scale simplified corpora corresponding to Simple English Wikipedia.[16–18,a] Therefore, we focus on language model pre-training to address a low-resource condition.[19,20] This has led to impressive results on various tasks such as text classification, question answering, and sequence labeling.[21–23] In particular, Shleifer[22] achieved a striking performance despite the use of small supervised examples.

In this study, we attempted to develop a simple approach at fine-tuning a pre-trained language model for text simplification using only a small parallel corpus. Specifically, we experimented with the following two models: (1) a transformer-based encoder–decoder model and (2) a language model that receives a joint input of the original and simplified sentences, which is called the translation language model.

2. Related Works

Research for automatic text simplification is generally divided into three systems: rule-based, lexical simplification and machine translation. Rule-based systems use rules manually created for syntactic simplification. Through analyzing a syntactic structure, the structure transforms into a simple structure.[24,25] Lexical simplification substitutes complex words with simpler alternatives.[26,27] The process includes the following four steps: complex word identification, substitution generation, substitution selection, and substitution ranking. Kajiwara and Yamamoto[28] usesd several Japanese paraphrasing datasets for substitution generation. Hading *et al.*[29] also used Japanese thesaurus and dependency-based word embeddings. In machine translation approaches, original sentences and simplified sentences are regarded as two different languages. Text simplification is the process to translate the original language into simplified language. These approaches need parallel corpora.[9,16,30,31] However, our simplification data has only 30,000 sentence pairs. Hence, we focus on data augmentation methods.

Back-translation is a method used for data augmentation. It constructs a synthetic parallel corpus by translating target monolingual data into a source language.[32,33] This augmentation method is effective not only for machine translation but also for monolingual translation tasks with few resources such as grammatical

[a] `https://dumps.wikimedia.org/simplewiki/`.

error correction.[34] Qiang[35] used a synthetic parallel corpus generated by a back-translating Simple English Wikipedia, as inspired by Sennrich *et al.*'s[32] method. By adding such synthetic data to the training data, even a simple machine translation model can outperform more complex models such as a model using reinforcement learning. However, back-translation cannot be applied to text simplification if no monolingual simplified corpus is available.

Kauchak[30] combined a language model trained with a small simplified corpus and another trained with a large original corpus. The combined model performed as effectively as the one trained using a large simplified corpus on language modeling and lexical simplification tasks. Motivated by this result, we attempted to improve the text simplification model using a large original corpus instead of a large simplified corpus. Specifically, through this approach, we aim to train a language model using a large original corpus, and subsequently fine-tune it using a small parallel corpus for text simplification.

3. Methods

We build two text simplification models by fine-tuning a pre-trained language model. In this section, we describe the pre-training method of a language model (Sec. 3.1). We then describe two simplification models: (1) an encoder–decoder model (Sec. 3.2) and (2) a translation language model (Sec. 3.3).

3.1. *Language model pre-training*

We use a language model based on a transformer.[36] Instead of bidirectional models such as ELMo[19] and BERT,[20] we use unidirectional models including GPT[37] for pre-training. For a sentence with N tokens $(x_1, x_2, \ldots, x_N)$, our language model trains the parameter θ to maximize the likelihood $p(x_1, x_2, \ldots, x_N; \theta)$:

$$p(x_1, x_2, \ldots, x_N; \theta) = \prod_{k=1}^{N} p(x_k \mid x_0, x_1, \ldots, x_{k-1}; \theta). \tag{1}$$

For pre-training, we use an article extracted from Japanese Wikipedia[b] by *WikiExtractor*[c] and the *Balanced Corpus of Contemporary Written Japanese* (BCCWJ).[d]

3.2. *Text generation from pre-trained encoder–decoder*

We incorporate the weights of the pre-training language model into a standard encoder–decoder model. The encoder–decoder model (Fig. 1) comprises an encoder

[b]https://dumps.wikimedia.org/jawiki/latest/jawiki-latest-pages-articles.xml.bz.
[c]https://github.com/attardi/wikiextractor.
[d]https://pj.ninjal.ac.jp/corpus_center/bccwj/.

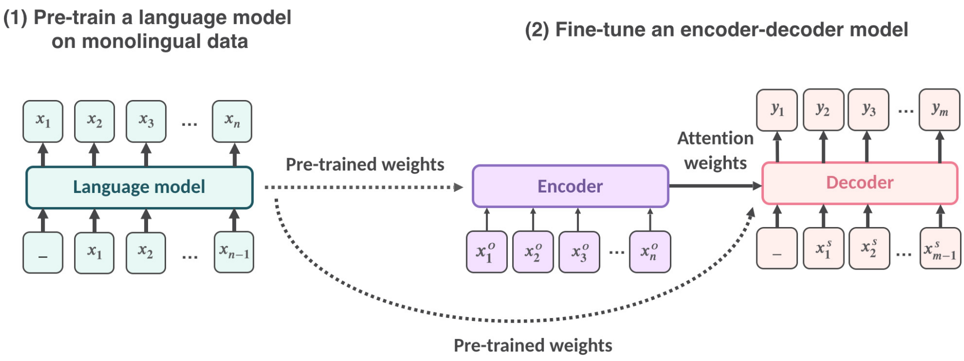

Fig. 1. Fine-tuning of the encoder–decoder model.

that reads the original sentences, a decoder that generates the simplified sentences, and an attention mechanism[38] that allows the decoder to access the encoder states during generation. Both the encoder and decoder use the same structure. We compared three different ways to incorporate the weights from a pre-trained language model according to Ramachandran *et al.*[39]: (1) pre-training the encoder only, (2) pre-training the decoder only, and (3) pre-training both the encoder and decoder. The parameters of the encoder–decoder attention mechanism are randomly initialized.

To show the effectiveness of the monolingual corpus, we conduct an experimental pre-training using only a parallel corpus instead of a large-scale monolingual corpus. During pre-training using a parallel corpus, the encoder and decoder are initialized through pre-trained weights on the original and simplified sides, respectively.

3.3. *Text generation from pre-trained language model*

We translate an original sentence into a simplified sentence using only a transformer decoder (Fig. 2) similar to that used by Khandelwal *et al.*[40] and Hoang *et al.*[41] Given the N tokens in the original sentence $X^o = (x_1^o, x_2^o, \ldots, x_N^o)$ and the M tokens in the simplified sentences $X^s = (x_1^s, x_2^s, \ldots, x_M^s)$, a transformer decoder receives the following input sequence, where $\langle\text{delim}\rangle$ is a special token, which is a delimiter between an original sentence and a simplified sentence:

$$X = [X^o, \langle\text{delim}\rangle, X^s]. \tag{2}$$

We use the same word-embedding layer when the original sentence and the simplified sentence are vectorized. The positional embedding obtained from the following equations are added to the word embeddings:

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin(\mathrm{pos}/10000^{2i/d_{\mathrm{model}}}), \tag{3}$$

$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos(\mathrm{pos}/10000^{2i/d_{\mathrm{model}}}), \tag{4}$$

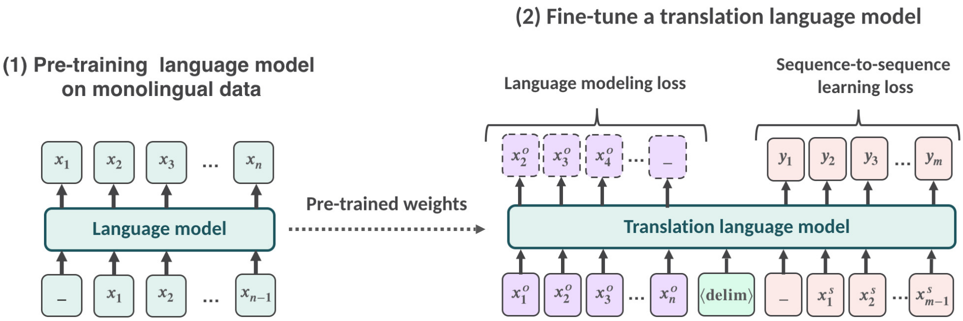

Fig. 2. Fine-tuning of the translation language model.

where pos indicates a position, i indicates the dimension, and d_{model} indicates the embedding dimension. Note that when the delimitation token ⟨delim⟩ is reached, the position counter is reset.

Unlike the encoder–decoder model, the translation language model can be fine-tuned without changing the structure of the pre-trained model. However, this fine-tuning procedure often leads to a catastrophic forgetting,[42] particularly when trained on small supervised datasets. To avoid this problem, we add a language modeling loss to the translation loss during the fine-tuning step. The translation and language modeling losses are weighted equally.

4. Experimental Setup

4.1. *Datasets*

We use the *simplification corpus of local government announcement*[43] as the supervised data. This corpus contains 1100 official documents distributed in public facilities, such as a city office, hospital, and school. The documents were simplified by 40 Japanese language teachers. The parallel corpus has three simplified versions: *literal translation*, *free translation*, and *summary*. Each simplified level is defined as follows:

- **Literal translation.** It is the simplified version that rewrites difficult words or phrases into simple expressions.
- **Free translation.** It is the simplified version that rewrites a difficult sentence into a simplified sentence while preserving the meaning in the best possible manner.
- **Summary.** It is the simplified version that contains document-level rewritings such as sentence extraction, in addition to sentence-level rewritings.

These comprise grammar and vocabulary defined in the Japanese-Language Proficiency Test Level 2 (N2). Each simplified sentence is manually aligned. In this study, we attempted to translate an original sentence into a *literal-translation* sentence or a *free-translation* sentence, which is a word-level or sentence-level simplification. A *summary*, which is a document-level simplification, will be addressed in the future.

The official document has numerous noisy sentences such as phone numbers, addresses, postal codes, and meaningless sentences depending on the document format. For preprocessing, we excluded those sentences and sentence pairs that had over 100 tokens on the original side or simplified side. The literal-translation corpus contains 32,949 sentence pairs for training and 1781 sentence pairs for testing. In addition, the free-translation corpus contains 30,259 sentence pairs for training and 1637 sentence pairs for testing. Some statistics of these datasets are detailed in Table 1.

Table 1. Comparison of text simplification datasets.

Datasets	N-gram overlap (%)				Mean # words	
	$N = 1$	$N = 2$	$N = 3$	$N = 4$	Original	Simplified
Literal	64.48	42.00	31.76	25.28	15.06	17.14
Free	61.97	38.37	28.05	21.85	15.32	15.84

4.2. *Model specifications and training details*

We use a unidirectional transformer language model with six layers and 16 masked self-attention heads for pre-training and fine-tuning. We set the number of dimensions of the word embedding layer to 512, and the number of dimensions of the feed-forward networks to 2048. The encoder–decoder and translation language models use the same parameters. We use scholastic gradient descent (SGD) for optimizing all models. We set the initial learning rate to 0.25, and multiply it with 0.1 when a validation loss has stopped improving during 10 epochs. The training ends if the learning rate becomes less than $1.0 * 10^{-5}$.

4.3. *Evaluation*

We evaluated the model's output based on two metrics, BLEU[44] and SARI.[10] BLEU is a traditional evaluation metric for machine translation tasks. It has a positive correlation with fluency and meaning preservation during a text simplification task that does not include sentence splitting.[45] The System output Against References and against the Input (SARI) is a recently proposed simplification metric that compares the system output against the references and input sentence, and is an arithmetic average of N-gram precision and the recall of three rewrite operations: addition, retention, and deletion. It rewards the addition operations in which the system output was not in the input but in the references. It also rewards the words

Table 2. Results on non pre-training setting

Model	BLEU	SARI	N-gram overlap [%] N=1	N=2	N=3	N=4	Avg. # word
Literal-translation							
Identical	34.65	17.87	-	-	-	-	17.23
Encoder-decoder	19.70	38.35	49.40	23.02	14.22	9.97	11.69
Translation LM	42.86	51.91	65.79	44.03	33.08	26.64	19.35
Reference	-	-	67.95	47.55	37.94	32.64	19.77
Free-translaiton							
Identical	29.31	15.86	-	-	-	-	17.89
Encoder-decoder	20.11	40.40	54.49	31.94	22.95	17.89	12.12
Translation LM	35.96	49.78	67.32	47.83	38.11	31.66	17.91
Reference	-	-	63.12	41.66	32.74	27.18	18.62

Identical denotes a system that outputs an input sentence. Furthermore, *Encoder-decoder* is the model described in Sec. 3.2 and *Translation LM* is the model described in Sec. 3.3.

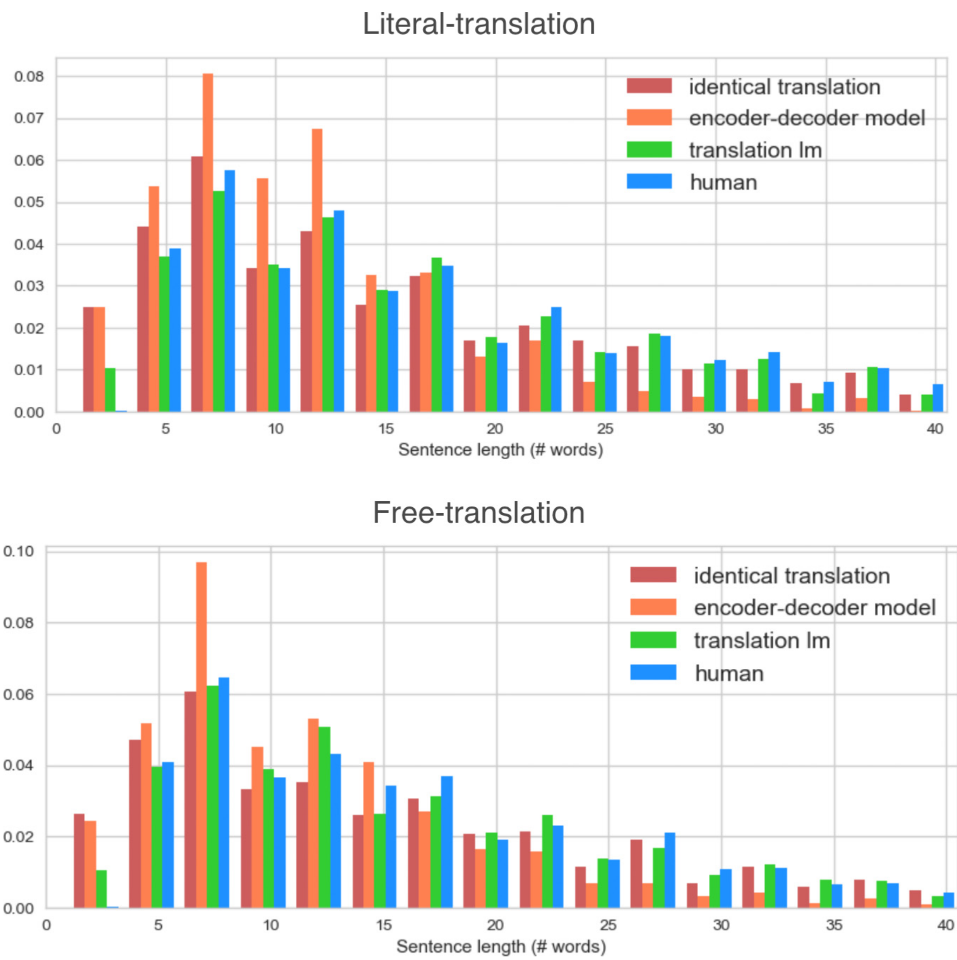

Fig. 3. Distributions of sentence length.

retained/deleted in both the system output and the references. SARI has a positive correlation with simplicity.[45,46]

5. Results

From Table 2 we can see that the translation language model greatly outperforms the encoder–decoder model in both BLEU and SARI. We believe that correct copying improves the performance. An N-gram overlap of the translation language model is close to that of the reference compared to the encoder–decoder model. In addition, Fig. 3 shows that the distribution of the translation language model is similar to that of the reference. There is a large difference in the copying performance between the translation language model and the encoder–decoder model. We believe that a self-attention mechanism[36] in the translation language model operates like a copy mechanism[47,48] because the encoder and decoder are the same components. In a monolingual translation task such as text simplification, the copying of words in an input sentence occupies an extremely large proportion during the translation operation. Therefore, the translation language model achieves a high performance.

The data presented in Table 3 shows that the pre-training of the simplification corpus does not improve the performance of either the encoder–decoder model or the

Table 3. Results on pre-training setting.

Model	Pre-trained corpus	Literal translation		Free translation	
		BLEU	SARI	BLEU	SARI
Identical translation	—	34.65	17.87	29.31	15.86
Encoder–decoder	—	19.70	38.35	20.11	40.40
Translation LM		42.86	51.91	35.96	49.78
Pre-train encoder only	Simplification corpus of local government announcement	18.44	38.17	17.09	39.25
Pre-train decoder only		10.86	31.10	8.92	31.19
Pre-train encoder and decoder		14.38	33.92	15.04	36.18
Translation LM		34.45	46.36	25.54	42.74
+ language modeling loss		30.03	43.52	24.67	41.99
Pre-train encoder only	Wikipedia	25.21	41.63	24.16	42.86
Pre-train decoder only		7.44	30.88	10.38	33.70
Pre-train encoder and decoder		13.32	34.41	13.67	36.16
Translation LM		44.15	52.46	**37.37**	**50.39**
+ language modeling loss		40.69	50.37	34.22	48.55
Pre-train encoder only	BCCWJ	17.89	38.22	16.88	39.23
Pre-train decoder only		11.38	33.20	13.31	35.84
Pre-train encoder and decoder		15.58	36.16	14.72	37.38
Translation LM		**45.07**	**53.25**	37.22	50.37
+ language modeling loss		41.29	50.99	34.47	48.80

translation language model. By contrast, pre-training with a large-scale monolingual corpus does improve the performance. These results indicate the effectiveness of pre-training with the original corpus. The pre-training of the encoder improves the result, whereas pre-training the decoder deteriorates it, which is in agreement with

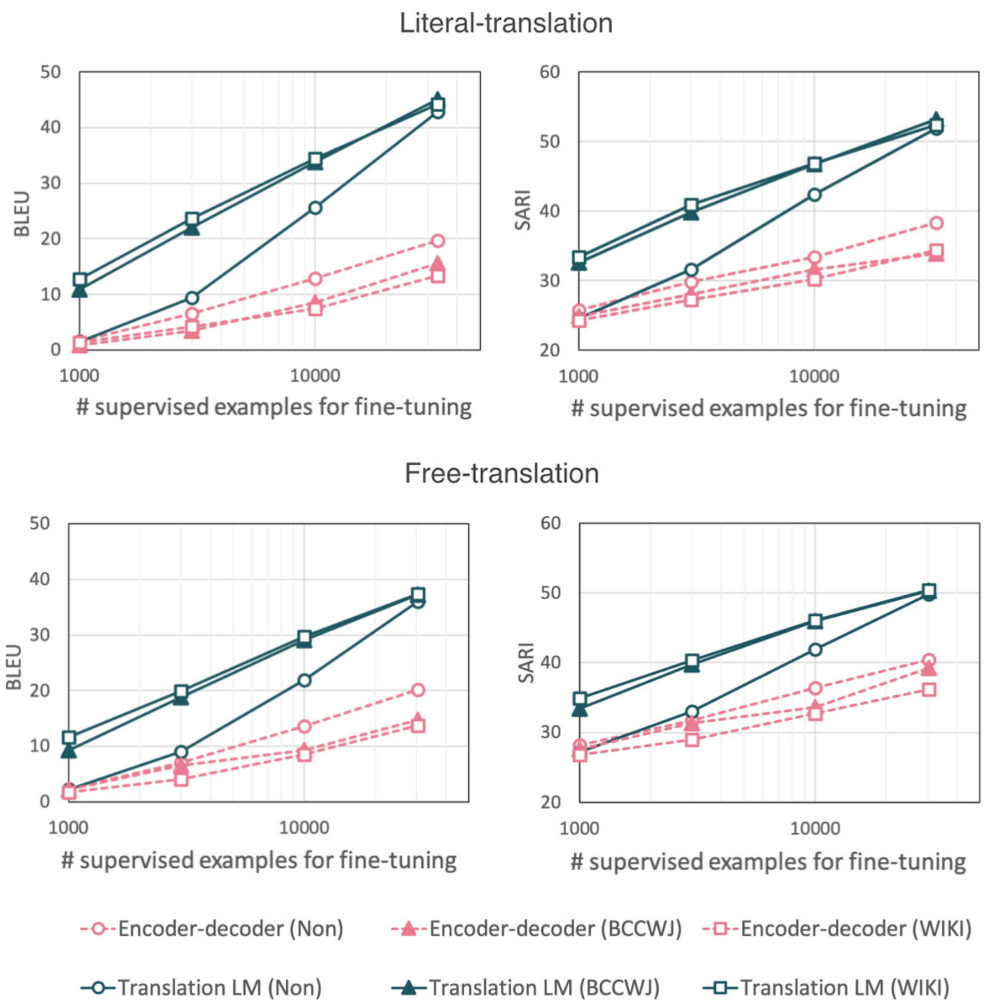

Fig. 4. Results at various data sizes. The round points denote the results under the non-pre-training setting. The triangle and square points denote the results for pre-training using BCCWJ and Japanese Wikipedia, and the dotted and solid lines denote the encoder–decoder model and the translation language model, respectively. We use the encoder–decoder model for which both the encoder and decoder are pre-trained, as well as the translation language model without language modeling loss.

the results of previous research.[40] In addition, the results presented in the table show that adding the language modeling loss into the fine-tuning of the translation language model results in poorer performance.

The results of SARI and BLEU at various supervised data sizes are presented in Fig. 4. We use an encoder–decoder model for which both the encoder and decoder are pre-trained, in addition to a translation language model without a language modeling loss. Figure 4 shows that pre-training with a large-scale monolingual corpus is more effective on the translation language model than on the encoder–decoder model. In particular, the translation language model fine-tuned with only 3000 examples achieves a performance comparable to the encoder–decoder model trained using 30,000 supervised data.

6. Conclusion

Neural text simplification models have significantly improved the simplification performance. However, parallel corpora for text simplification are very few in number and smaller in size. Therefore, we attempted to facilitate the training of simplification rewritings by pre-training from large-scale monolingual corpora such as Wikipedia articles and BCCWJ. To fine-tune a language model in a seamless manner, we proposed the use of a translation language model. Experimental results show that the translation language model substantially outperforms a state-of-the-art model under a low-resource setting. In particular, the proposed model is able to copy accurately the words and phrases in the original sentence. In addition, the pre-trained translation language model with only 3000 supervised examples can achieve a performance comparable to a state-of-the-art model with more than 10 times the number of supervised examples.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481.

Appendix A

Table 4 shows that *Translation LM* can copy source words more correctly than the *Encoder–decoder*.

Table 4. Examples of output.

	Examples
Identical	健康 診査 票 が ない と 健診 を 受ける こと が でき ませ ん (今回 ご`案内 さ せ て いただい た 郵便 物 に 同封 さ れ て い ます)。 If you do not have a medical checkup form, you will not be able to receive a medical checkup (it is enclosed in this mail).
Encoder–decoder	健康 診断 の 結果 が でき ませ ん 。 です。 You cannot get the result of your health check.
Translation LM	健康 診断 の 紙 が ない と 健康 診断 を 受ける こと が でき ませ ん (今回 案内 し た 手紙 に 入っ て い ます) 。 If you do not have a form for medical checkup, you will not be able to receive a medical checkup. (It is in this mail).
Reference	健康 診断 票 が なかっ たら 健康 診断 を 受ける こと が でき ませ ん (今回 案内 し た 手紙 に 入っ て い ます)。 If you do not have a medical checkup form, you will not be able to receive a medical checkup (it is in this mail).
Identical	警報 ・ 避難 の 指示 等 の 内容 の 伝達 訓練 及び 被災 情報 ・ 安否 情報 に 係る 情報 収集 訓練。 Training to transmit information about warning and evacuation instructions and training to gather information regarding disaster and safety.
Encoder–decoder	逃げる 住民 を 案内 の 情報 を 集め て， 整理 し ます。 Gather and organize guides for the people who will run away.
Translation LM	警報 ・ 逃げる 指示 など の 内容 の 連絡 練習 と 災害 に つい て の 情報 を 集め て の 練習。 Training to transmit information about warning and instructions to escape and training to gather information about disasters.
Reference	警報 や 逃げる 指示 など の 内容 を 伝える 練習 と 災害 に あっ た 情報 ・ 無事 か どう か の 情報 に つい て の 情報 を 集める 練習。 Training to transmit information about warning and instructions to escape, and training to gather information about disaster and safety.
Identical	請求の際には、本人又は法定代理人自身であることを証 明 する 書類 (運転 免許 証，旅券， 健康 保険 の 被 保険 者 証 等) の 提出 が 必要 です。 When you make a claim, you need to show an identity card (such as a driver's license, passport, health insurance card, etc.) in order to prove that you are the principal or legal representative.
Encoder–decoder	健康保険の証明書を出すときは，本人だということが必 要 です。 When you issue a health insurance certificate, you need to be the principal.
Translation LM	請求のときには，その人か法定代理人であることを証明 する 書類 (運転 免許 証， 健康 保険 の 被 保険 者 証 など) の 出 す こと が 必要 です。 When you make a claim, you need to show an identity card (such as a driver's license, health insurance card, etc.) in order to prove that you are the principal or legal representative.
Reference	請求するときには，自分が本人か法定代理人であることを 証明 する 書類 (運転 免許 証， パスポート， 健康 保険 の 被 保険 者 証 等) を 出す こと が 必要 です。 When you make a claim, you need to show an identity card (such as a driver's license, passport, health insurance card, etc.) in order to prove that you are the principal or legal representative.

References

1. K. Iwata, The preference for English in linguistic services: 'Japanese for Living: Countrywide Survey' and Hiroshima, *Jpn. J. Lang. Soc.* **13** (2010) 81–94.
2. R. Chandrasekar, C. Doran and B. Srinivas, Motivations and methods for text simplification, in *Proc. 16th Int. Conf. Computational Linguistics (COLING 2004)*, Vol. 2 (1996).

3. A. Siddharthan, A. Nenkova and K. McKeown, Syntactic simplification for improving content selection in multi-document summarization, in *Proc. 20th Int. Conf. Computational Linguistics* (The COLING 2004 Organizing Committee, 2004), pp. 896–902.
4. W. Xu and R. Grishman, A parse-and-trim approach with information significance for Chinese sentence compression, in *Proc. 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)* (Association for Computational Linguistics, 2009), pp. 48–55.
5. D. Vickrey and D. Koller, Sentence simplification for semantic role labeling, in *Proc. ACL-08: HLT* (Association for Computational Linguistics, 2008), pp. 344–352.
6. M. Miwa, R. Sætre, Y. Miyao and J. Tsujii, Entity-focused sentence simplification for relation extraction, in *Proc. 23rd Int. Conf. Computational Linguistics (COLING 2010)* (The COLING 2010 Organizing Committee, 2010), pp. 788–796.
7. H.-B. Chen, H.-H. Huang, H.-H. Chen and C.-T. Tan, A simplification-translation-restoration framework for cross-domain SMT applications, in *Proc. COLING 2012: Technical Papers* (The COLING 2012 Organizing Committee, 2012), pp. 545–560.
8. S. Štajner and M. Popovic, Can text simplification help machine translation?, in *Proc. 19th Annu. Conf. European Association for Machine Translation* (2016), pp. 230–242.
9. S. Wubben, A. van den Bosch and E. Krahmer, Sentence simplification by monolingual machine translation, in *Proc. 50th Annu. Meeting of the Association for Computational Linguistics: Long Papers*, Vol. 1 (Association for Computational Linguistics, 2012), pp. 1015–1024.
10. W. Xu, C. Napoles, E. Pavlick, Q. Chen and C. Callison-Burch, Optimizing statistical machine translation for text simplification, *Trans. Assoc. Comput. Linguist.* **4** (2016) 401–415, doi:10.1162/tacl_a_00107.
11. S. Nisioi, S. Štajner, S. P. Ponzetto and L. P. Dinu, Exploring neural text simplification models, in *Proc. 55th Annu. Meeting of the Association for Computational Linguistics: Short Papers*, Vol. 2 (Association for Computational Linguistics, 2017), pp. 85–91.
12. X. Zhang and M. Lapata, Sentence simplification with deep reinforcement learning, in *Proc. 2017 Conf. Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2017), pp. 584–594.
13. S. Zhao, R. Meng, D. He, A. Saptono and B. Parmanto, Integrating transformer and paraphrase rules for sentence simplification, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2018), pp. 3164–3173.
14. H. Guo, R. Pasunuru and M. Bansal, Dynamic multi-level multi-task learning for sentence simplification, in *Proc. 27th Int. Conf. Computational Linguistics* (Association for Computational Linguistics, 2018), pp. 462–476.
15. R. Kriz, J. Sedoc, M. Apidianaki, C. Zheng, G. Kumar, E. Miltsakaki and C. Callison-Burch, Complexity-weighted loss and diverse reranking for sentence simplification, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Long and Short Papers*, Vol. 1 (Association for Computational Linguistics, 2019), pp. 3137–3147.
16. Z. Zhu, D. Bernhard and I. Gurevych, A monolingual tree-based translation model for sentence simplification, in *Proc. 23rd Int. Conf. Computational Linguistics (COLING 2010)* (The COLING 2010 Organizing Committee, 2010), pp. 1353–1361.
17. K. Woodsend and M. Lapata, Learning to simplify sentences with quasi-synchronous grammar and integer programming, in *Proc. 2011 Conf. Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2011), pp. 409–420.

18. W. Coster and D. Kauchak, Simple English Wikipedia: A new text simplification task, in *Proc. 49th Annu. Meeting of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, 2011), pp. 665–669.
19. M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, Deep contextualized word representations, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies — Long Papers*, Vol. 1 (Association for Computational Linguistics, 2018), pp. 2227–2237.
20. J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2018), pp. 4171–4186.
21. J. Howard and S. Ruder, Universal language model fine-tuning for text classification, in *Proc. 56th Annu. Meeting of the Association for Computational Linguistics: Long Papers*, Vol. 1 (Association for Computational Linguistics, 2018), pp. 328–339.
22. S. Shleifer, Low resource text classification with ULMFit and backtranslation, arXiv:1903.09244 [CS.CL].
23. A. Chronopoulou, C. Baziotis and A. Potamianos, An embarrassingly simple approach for transfer learning from pretrained language models, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies — Long and Short Papers*, Vol. 1 (Association for Computational Linguistics, 2019), pp. 2089–2095.
24. A. Siddharthan and A. Mandya, Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules, in *Proc. 14th Conf. European Chapter of the Association for Computational Linguistics* (Association for Computational Linguistics, 2014), pp. 722–731.
25. J. Lee and J. B. K. P. Don, Splitting complex English sentences, in *Proc. 15th Int. Conf. Parsing Technologies* (Association for Computational Linguistics, 2017), pp. 50–55.
26. G. H. Paetzold and L. Specia, Unsupervised lexical simplification for non-native speakers, in *Proc. Thirtieth AAAI Conf. Artificial Intelligence* (2016).
27. G. Paetzold and L. Specia, Lexical simplification with neural ranking, in *Proc. 15th Conf. European Chapter of the Association for Computational Linguistics: Short Papers*, Vol. 2 (Association for Computational Linguistics, 2017), pp. 34–40.
28. T. Kajiwara and K. Yamamoto, Evaluation dataset and system for Japanese lexical simplification, in *Proc. ACL-IJCNLP 2015 Student Research Workshop* (Association for Computational Linguistics, 2015), pp. 35–40.
29. M. Hading, Y. Matsumoto and M. Sakamoto, Japanese lexical simplification for non-native speakers, in *Proc. 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)* (The COLING 2016 Organizing Committee, 2016), pp. 92–96.
30. D. Kauchak, Improving text simplification language modeling using unsimplified text data, in *Proc. 51st Annu. Meeting of the Association for Computational Linguistics: Long Papers*, Vol. 1 (Association for Computational Linguistics, 2013), pp. 1537–1546.
31. W. Xu, C. Callison-Burch and C. Napoles, Problems in current text simplification research: New data can help, *Trans. Assoc. Comput. Linguist.* **3** (2015) 283–297, doi:10.1162/tacl_a_00139.

32. R. Sennrich, B. Haddow and A. Birch, Improving neural machine translation models with monolingual data, in *Proc. 54th Annu. Meeting of the Association for Computational Linguistics: Long Papers*, Vol. 1 (Association for Computational Linguistics, 2016), pp. 86–96.
33. S. Edunov, M. Ott, M. Auli and D. Grangier, Understanding back-translation at scale, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2018), pp. 489–500.
34. Z. Xie, G. Genthial, S. Xie, A. Ng and D. Jurafsky, Noising and denoising natural language: Diverse backtranslation for grammar correction, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies — Long Papers*, Vol. 1 (Association for Computational Linguistics, 2018), pp. 619–628.
35. J. Qiang, Improving neural text simplification model with simplified corpora, arXiv:1810.04428 [CS.CL].
36. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in *Advances in Neural Information Processing Systems* **30** (Curran Associates, Inc., 2017), pp. 5998–6008.
37. A. Radford *et al.*, Improving language understanding by generative pre-training (transformer in real world) (2018), *OpenAI*, http://S3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.
38. D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, in *Proc. 3rd Int. Conf. Learning Representations* (2015), pp. 1–15.
39. P. Ramachandran, P. Liu and Q. Le, Unsupervised pretraining for sequence to sequence learning, in *Proc. 2017 Conf. Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2017), pp. 383–391.
40. U. Khandelwal, K. Clark, D. Jurafsky and L. Kaiser, Sample efficient text summarization using a single pre-trained transformer, arXiv:1905.08836 [CS.CL].
41. A. Hoang, A. Bosselut, A. Celikyilmaz and Y. Choi, Efficient adaptation of pretrained transformers for abstractive summarization, arXiv:1906.00138 [CS.CL].
42. I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville and Y. Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, arXiv:1312.6211 [stat.ML].
43. M. Moku and H. Yamamoto, Automatic easy Japanese translation for information accessibility of foreigners, in *Proc. Workshop on Speech and Language Processing Tools in Education* (The COLING 2012 Organizing Committee, 2012), pp. 85–90.
44. K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, BLEU: A method for automatic evaluation of machine translation, in *Proc. 40th Annu. Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, 2002), pp. 311–318.
45. E. Sulem, O. Abend and A. Rappoport, BLEU is not suitable for the evaluation of text simplification, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2018), pp. 738–744.
46. T. Vu, B. Hu, T. Munkhdalai and H. Yu, Sentence simplification with memory-augmented neural networks, in *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies — Short Papers*, Vol. 2 (Association for Computational Linguistics, 2018), pp. 79–85.
47. J. Gu, Z. Lu, H. Li and V. O. Li, Incorporating copying mechanism in sequence-to-sequence learning, in *Proc. 54th Annu. Meeting of the Association for Computational*

Linguistics: Long Papers, Vol. 1 (Association for Computational Linguistics, 2016), pp. 1631–1640.
48. A. See, P. J. Liu and C. D. Manning, Get to the point: Summarization with pointer-generator networks, in *Proc. 55th Annu. Meeting of the Association for Computational Linguistics: Long Papers*, Vol. 1 (Association for Computational Linguistics, 2017), pp. 1073–1083.