Parallel Corpus Filtering for Japanese Text Simplification

Koki Hatagaki **Tomoyuki Kajiwara** **Takashi Ninomiya**
Graduate School of Science and Engineering, Ehime University, Japan
hatagaki@ai.cs.ehime-u.ac.jp {kajiwara, ninomiya}@cs.ehime-u.ac.jp

Abstract

We propose a method of parallel corpus filtering for Japanese text simplification. The parallel corpus for this task contains some redundant wording. In this study, we first identify the type and size of noisy sentence pairs in the Japanese text simplification corpus. We then propose a method of parallel corpus filtering to remove each type of noisy sentence pair. Experimental results show that filtering the training parallel corpus with the proposed method improves simplification performance.

1 Introduction

The number of foreign residents in Japan is increasing yearly due to government policies and the progress of globalization. Iwata (2010) reported that many of them can partially understand Japanese, more than the number who can understand other languages such as English or Chinese. Therefore, many information such as disaster information and daily news (Tanaka et al., 2013) are provided in "easy Japanese" in Japan today.

Recent research in text simplification has focused on data-driven approaches (Alva-Manchego et al., 2020) based on parallel corpora (Coster and Kauchak, 2011; Xu et al., 2015; Zhang and Lapata, 2017; Jiang et al., 2020). For Japanese, a parallel corpus with tens of thousands of sentence pairs (Maruyama and Yamamoto, 2018; Katsuta and Yamamoto, 2018) is available for the study of text simplification. However, the Japanese text simplification corpus contains 16% of noisy sentence pairs, as shown in Table 1, which hinders the simplification performance.

In this study, we first identify the type and size of noisy sentence pairs in the Japanese text simplification corpus. As shown in Table 1, there are three main types of noise: sentence pairs with large differences in sentence length, sentence pairs with different meanings, and sentence pairs with low fluency. We then propose a method of parallel corpus filtering to remove each type of noisy sentence pair. For noisy sentence pairs with large differences in sentence length, we design methods of parallel corpus filtering based on differences in the number of tokens or Levenshtein distance. For noisy sentence pairs with different meanings, we design methods of parallel corpus filtering based on word embeddings or sentence embeddings. For noisy sentence pairs with low fluency, we design methods of parallel corpus filtering based on the perplexity of language models.

Type of noise	Ratio
large difference in sentence length	4% (20/500)
different meanings	8% (42/500)
low fluency	8% (41/500)
other noise	1% (2/500)
sentence pairs without noise	84% (419/500)

Table 1: Types of noisy sentence pairs and their ratios in the Japanese text simplification corpus.

We conducted experiments to evaluate the effectiveness of parallel corpus filtering for Transformer- and BART-based text simplification models (Vaswani et al., 2017; Lewis et al., 2020). Experimental results show that our methods are more effective for the BART-based text simplification model. Specifically, parallel corpus filtering based on differences in sentence lengths enables BART to achieve the best simplification performance for both metrics of BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016).

2 Related Work

Parallel corpus filtering (Koehn et al., 2020) is a technique that has been studied primarily in machine translation tasks where a large parallel training corpus is available, and contributes to improving the quality of the generated text by removing

Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), pages 12 - 18
December 8, 2022

Type of noise	Original sentence	Simplified sentence
large difference in sentence length	それの代金を仕払うことによって 確立する所有権 (You establish the property right by paying for it.)	買う
	このひもは強い (This string is strong.)	この物を制限するための 長いものは強い
	洪水がおさまり始めた (The flood began to subside.)	水の量が増えて川から出る 状態が静かになり始めた
different meanings	くじで誰が勝つか決めよう (Let's decide the winner by lot.)	勝ったか負けたか 決めることができない
	熱はたいていの物を膨張させる (Heat expands most things.)	あらゆる物は熱で増える
	彼女はみんなをうんざりさせます (She drives everybody up the wall.)	彼女はみんなを飽きさせます
low fluency	豆腐は良い酒の肴になる (Tofu goes well with good sake.)	植物で作った白い柔らかい物を 食べると，うまい酒が たくさん飲むことができる
	金融引き締めで金利が上昇するだろう (Interest rates will rise due to monetary tightening.)	金の流れを厳しくすることで 金を借りる際に返す時につける 金が占める率のが上がるだろう
	酸が金属を腐食した (The acid ate into the metal.)	酸っぱい特徴を持つ水が 金属を腐らせた

Table 2: Examples of noisy sentence pairs in the Japanese text simplification corpus.

noisy sentence pairs from the training data. Text simplification tasks use relatively small training data. Therefore, parallel corpus mining (Hwang et al., 2015; Kajiwara and Komachi, 2016; Jiang et al., 2020) has been actively studied, but there is no prior research on parallel corpus filtering for text simplification.

3 Methodology

We propose a method of parallel corpus filtering for noise in Japanese simplification corpus (Maruyama and Yamamoto, 2018; Katsuta and Yamamoto, 2018) to improve the performance of Japanese text simplification. Parallel corpus filtering is performed on the training data using multiple methods, and the resulting subset of training data is used to train the text simplification model. Transformer (Vaswani et al., 2017) and pre-trained BART (Lewis et al., 2020) are used for the model, and a text simplification model is constructed by fine-tuning using a subset of the Japanese simplification corpus. First, in Section 3.1, we analyze noise in the Japanese simplification corpus and define three representative types of noise that we target. In Sections 3.2 to 3.4, we then describe our proposed method for detecting each of the noises.

3.1 Definition of Noise

We manually classified the noise type in 500 randomly selected sentence pairs from the Japanese simplification corpus, and the results are shown in Table 1. The Japanese simplification corpus is a parallel corpus in which given sentences are paraphrased to make them simpler. Since these are manually paraphrased, most of the sentence pairs are expected to be noise-free.

Our analysis confirms that 84% of the sentence pairs do not contain noise. We manually classified the noisy sentence pairs and found that the three main types of noise were sentence pairs with large sentence length differences, low synonymy, and low fluency. Table 2 shows examples of these noises. Sentence pairs with large differences in sentence length often contain examples in which complex phrases were replaced with expressions similar to dictionary definitions. Sentence pairs with low synonymy often contain errors in rewriting, in which related but non-synonymous expressions are used. Sentence pairs with low fluency often contain redundant expressions due to oversimplification and simple errors such as incorrect particles.

3.2 Methods for Sentence Length Difference

This method performs parallel corpus filtering by detecting noise with large sentence length differences between complex and simple sentences. These noise types include cases in which excessive information is lost due to extreme simplification and cases in which complex phrases are replaced with dictionary definition-like expressions. To determine the sentence length difference between sentence pairs, we propose two methods: one is to use the absolute value of the difference in the number of tokens, and the other is to use the edit distance per token. Three types of tokens are used: characters, words, and subwords. In this paper, words with the best performance are used as tokens. Sentence pairs with a sentence length difference larger than a threshold are detected as noise and removed from the training data.

3.3 Methods for Sentence Meaning

This method performs parallel corpus filtering by detecting noise with the small semantic similarity between complex and simple sentences. These noises include rewritings that use related but not identical expressions, such as "most things" and "all things". To estimate the semantic similarity between sentences, we propose three methods based on word and sentence embeddings. The method based on word embeddings uses the Japanese model of fastText (Bojanowski et al., 2017). The method based on sentence embeddings uses mUSE (Chidambaram et al., 2019), which is a multilingual version of Universal Sentence Encoder (Cer et al., 2018). Sentence pairs whose semantic similarity between sentences is lower than a threshold are detected as noise and removed from the training data.

First, we use a method (Shen et al., 2018) which constructs sentence embeddings by mean pooling of word embeddings. This method is also used as a baseline method in the previous study (Kajiwara and Komachi, 2016). The cosine similarity between the sentence variates obtained in this way is used to estimate the semantic similarity between sentences.

Second, we use the word variate alignment method (Song and Roth, 2015), which is also used in the previous study (Kajiwara and Komachi, 2016). This method considers the problem of word alignment between sentences as a weighted complete bipartite graph matching problem, where the word variates are nodes and the edge weights are the cosine similarity between the word variates, and word alignment is obtained by maximum matching. Then, the cosine similarity of the word embeddings between the aligned words are averaged to estimate the semantic similarity between the sentences.

Third, we use the cosine similarity of the sentence embeddings by mUSE. Recent general-purpose sentence encoders such as BERT (Devlin et al., 2019) are difficult to properly estimate semantic similarity between sentences without fine-tuning. Since there is no labeled corpus available for estimating semantic similarity between sentences in Japanese, we use mUSE, which can estimate semantic similarity between sentences without fine-tuning.

3.4 Methods for Sentence Fluency

To estimate sentence fluency, we propose two methods based on language models. The method based on the unidirectional language model uses the Japanese model of GPT-2 (Radford et al., 2019). The method based on the bidirectional language model uses the Japanese model of BERT. Sentence pairs containing sentences with perplexity higher than a threshold are detected as noise and removed from the training data.

First, we use perplexity based on a unidirectional language model. While the N-gram language model is used in the previous study (Zhang and Lapata, 2017; Kriz et al., 2019), this study uses the GPT-2 neural language model to consider all intra-sentence contexts.

Second, we use pseudo perplexity (Salazar et al., 2020) based on a bidirectional language model. The perplexity based on the bidirectional language model is the sum of the log-likelihoods of the conditional probabilities of estimating a masked word from surrounding words.

4 Experiments

To evaluate the effectiveness of the proposed method, we conduct experiments on Japanese text simplification. First, we describe the dataset and evaluation metrics in Section 4.1, then our experimental setup including models and hyperparameters in Section 4.2, and threshold setting on the validation set in Section 4.3. Finally, Section 4.4 presents our experimental results.

Method	Threshold	deleted sentence	Transformer		BART	
			BLEU	SARI	BLEU	SARI
Baseline (w/o parallel corpus filtering)	-	0	75.29	64.17	81.56	62.88
Difference in the number of tokens	12	5,173	73.63	62.95	**83.60**	**63.69**
Levenshtein Distance	10	4,065	**76.47**	63.92	**83.38**	**63.13**
Average of Word Embeddings	0.85	1,553	**75.90**	63.41	80.68	59.80
Word Alignment	0.7	12,533	73.81	63.48	80.93	62.31
Sentence Embeddings	0.5	2,312	73.63	62.70	81.50	61.32
Unidirectional Language Model	60	894	**77.26**	**64.19**	**82.34**	**63.00**
Bidirectional Language Model	200	4,979	74.28	63.34	**82.15**	**63.05**

Table 3: Experimental results. The upper, middle, and lower rows are our parallel corpus filtering methods based on differences in sentence length, synonymy, and fluency, respectively. Bolded letters indicate scores that outperform the baseline model without parallel corpus filtering, and underlined letters indicate the best performance.

4.1 Dataset and Evaluation Metrics

In our experiments, we used the Japanese simplification corpus[1,2] (Maruyama and Yamamoto, 2018; Katsuta and Yamamoto, 2018). The Japanese simplification corpus consists of 85,000 manually paraphrased sentence pairs of complex and simple sentences. Among them, 50,000 sentence pairs were annotated by university students who are native speakers of Japanese. The other 35,000 sentence pairs were annotated by native Japanese speakers hired via a crowdsourcing service. We used multi-reference 100-sentence pairs annotated with seven types of reference sentences for testing. For validation, 2,000 sentence pairs were randomly selected from other sentence pairs annotated with a single reference sentence. The other 82,300 sentence pairs were for training and were targeted for parallel corpus filtering.

The performance of the text simplification models is automatically evaluated by BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016), which are commonly used in this task. These metrics are implemented in EASSE[3] (Alva-Manchego et al., 2019). As a pre-processing step for automatic evaluation, we performed word segmentation with MeCab[4] (Kudo et al., 2004). The effectiveness of parallel corpus filtering is evaluated by comparing the performance of text simplification models trained on the entire training set and a subset of the training corpus extracted by the proposed method, using BLEU and SARI, respectively.

[1] https://www.jnlp.org/GengoHouse/snow/t15
[2] https://www.jnlp.org/GengoHouse/snow/t23
[3] https://github.com/feralvam/easse
[4] http://taku910.github.io/mecab/

4.2 Settings

For the text simplification model, we used Transformer (Vaswani et al., 2017) and BART[5] (Lewis et al., 2020), which was pre-trained Transformer on Japanese Wikipedia. We implemented these models using fairseq[6] (Ott et al., 2019) and used Adam (Kingma and Ba, 2015) as the optimization method for fine-tuning, setting $\beta_1 = 0.9$, $\beta_2 = 0.99$, and the learning rate as 5e-4. The batch size was set to 4,096 tokens, and label smoothing and dropout were used for regularization. The dropout probability was set to 0.2. Training was terminated when the cross-entropy loss in the validation set did not improve for five checkpoints.

As pre-processing for Transformer, we performed word segmentation with MeCab. As preprocessing for BART, we performed word segmentation with Juman++[7] (Morita et al., 2015) and subword segmentation with SentencePiece[8] (Kudo and Richardson, 2018). The vocabulary size for subword segmentation was set to 8,000.

Following models were used for parallel corpus filtering of the proposed method. Japanese fastText[9] (Bojanowski et al., 2017) was used for word embeddings. MeCab was used for word segmentation. For sentence embeddings, we used mUSE, a multilingual version of Universal Sentence Encoder[10] (Cer et al., 2018). For the unidirectional

[5] https://github.com/utanaka2000/fairseq/tree/japanese_bart_pretrained_model
[6] https://github.com/pytorch/fairseq
[7] https://github.com/ku-nlp/jumanpp
[8] https://github.com/google/sentencepiece
[9] https://fasttext.cc/
[10] https://tfhub.dev/google/universal-sentence-encoder-multilingual/3

language model, we used Japanese GPT-2.[11] For the bidirectional language model, we used Japanese BERT.[12] The GPT-2 and BERT language models were implemented using HuggingFace Transformers (Wolf et al., 2020).

4.3 Thresholds

We set thresholds for each method for parallel corpus filtering through the evaluation of simplification performance on the validation set. In this experiment, we set our thresholds with respect to SARI, the primary automatic evaluation metric for text simplification.

4.4 Results

Table 3 shows the experimental results. Transformer improved BLEU by parallel corpus filtering on edit distance, and improved both BLEU and SARI by parallel corpus filtering on unidirectional language models. In BART, parallel corpus filtering for sentence length difference and fluency improved both BLEU and SARI. On the other hand, parallel corpus filtering for synonymy worsened both BLEU and SARI in both models.

5 Conclusion

To improve the performance of Japanese text simplification, we proposed methods of parallel corpus filtering to remove noisy sentence pairs from the training dataset in terms of differences in sentence length, synonymy, and fluency. Experiments on text simplification models based on Transformer and BART showed that parallel corpus filtering based on differences in sentence length and perplexity of language models improved both metrics of BLEU and SARI over the baseline model without parallel corpus filtering.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP22H03651. This research was also obtained from the commissioned research (No. 225) by National Institute of Information and Communications Technology (NICT), Japan.

[11] https://huggingface.co/rinna/japanese-gpt2-medium

[12] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

References

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier Automatic Sentence Simplification Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing : System Demonstrations*, pages 49–54.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-Driven Sentence Simplification: Survey and Benchmark. *Computational Linguistics*, 46(1):135–187.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Muthu Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, pages 250–259.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning Sentences from Standard Wikipedia to Simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217.

Kazunari Iwata. 2010. The Preference for English in Linguistic Services: 'Japanese for Living: Country wide Survey' and Hiroshima. *The Japanese Journal of Language in Society*, 13(1):81–94.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF Model for

Sentence Alignment in Text Simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960.

Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158.

Akihiro Katsuta and Kazuhide Yamamoto. 2018. Crowdsourced Corpus of Sentence Simplification with Core Vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 461–466.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzman. 2020. Findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742.

Reno Kriz, Joao Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-Weighted Loss and Diverse Reranking for Sentence Simplification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3137–3147.

Taku Kudo and John Richardson. 2018. Sentencepiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237.

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.

Takumi Maruyama and Kazuhide Yamamoto. 2018. Simplified Corpus with Core Vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1153–1160.

Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. 2015. Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297.

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*, pages 48–53.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *Technical Report, OpenAI*, pages 1–24.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 440–450.

Yangqiu Song and Dan Roth. 2015. Unsupervised Sparse Vector Densification for Short Text Similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280.

Hideki Tanaka, Hideya Mino, Tadashi Kumano, Ochi, and Shibata. 2013. News Service in Simplified Japanese and Its Production Support Systems. In *Proceedings of the IBC2013 Conference*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara

Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

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.

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.