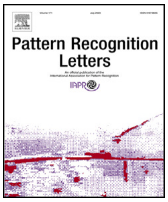
Pattern Recognition Letters 177 (2024) 40–46

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Representation transfer and data cleaning in multi-views for text simplification

Wei He [a,b,*], Katayoun Farrahi [a], Bin Chen [c], Bohua Peng [b], Aline Villavicencio [b]

[a] Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom
[b] Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, United Kingdom
[c] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, United Kingdom

ARTICLE INFO

Editor: Maria De Marsico

Keywords:
Text simplification
Sentence representation
Pre-trained language model
Data cleaning
Decoding

ABSTRACT

Representation transfer is a widely used technique in natural language processing. We propose methods of cleaning the dominant dataset of text simplification (TS) WikiLarge in multi-views to remove errors that impact model training and fine-tuning. The results show that our method can effectively refine the dataset. We propose to take the pre-trained text representations from a similar task (e.g., text summarization) to text simplification to conduct a continue-fine-tuning strategy to improve the performance of pre-trained models on TS. This approach will speed up the training and make the model convergence easier. Besides, we also propose a new decoding strategy for simple text generation. It is able to generate simpler and more comprehensible text with controllable lexical simplicity. The experimental results show that our method can achieve good performance on many evaluation metrics.

1. Introduction

Recent works have shown that models pre-trained on large corpus can learn universal language representations, which are beneficial for downstream Natural Language Processing (NLP) tasks and can avoid the need for training a new model from scratch [1]. Moreover, fine-tuning a pre-trained model (PTM) with only a relatively small dataset can outperform models trained from scratch with a large dataset [2]. Sequence-to-sequence (Seq2Seq) modeling is naturally fit for tasks with a source sequence and a target sequence, such as text simplification (TS). Training a Seq2Seq model usually heavily relies on the quality of the task-specific parallel datasets. Although the recent emerging dedicated pre-trained models provide a new paradigm to train a model (i.e., fine-tune the pre-trained model with a smaller dataset for the objective task), high-quality datasets for TS training and fine-tuning are scarce [3].

Previous fine-tuning methods normally take a PTM with task-specific fine-tuning [1]. However, the text representations of the PTM may not be suitable for the objective task, because their training objectives are different from each other. We hypothesize that borrowing representations from a PTM trained on summarization to simplification will boost performance and training speed as illustrated in Fig. 1.

Wikipedia-based datasets, such as WikiLarge [4], dominate model training in recent deep-learning-based TS studies [5]. However, those datasets have many errors like sentence pair misalignments, noise, and inaccurate and limited variations of simplifications (See Table 2). These errors negatively contribute to the model training [3,6,7].

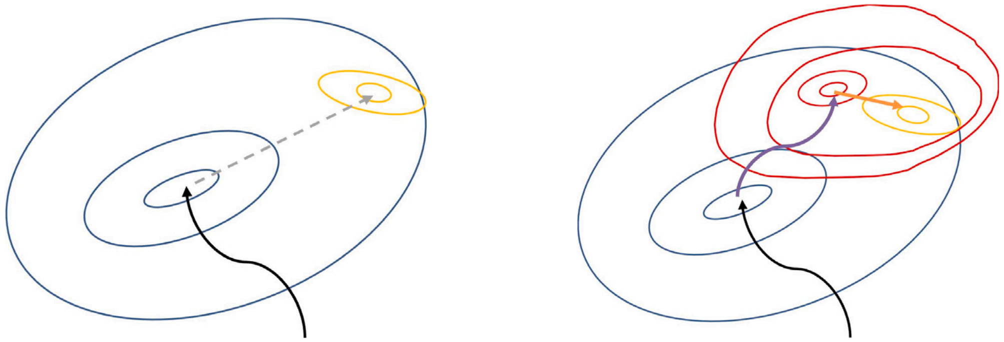
To address this, one approach is to find alternative datasets, such as Newsela [6]. However, these alternatives require permission to get access or are insufficient in data size to train a good deep learning model (See Table 1). A second option is to use text mining techniques to automatically collect paraphrases to create a large training corpus for TS, then train the model on the newly built dataset [8,9]. Another option is to adopt unsupervised methods to deal with this [10,11]. However, approaches that adopt unsupervised learning often involve complicated architectures and perform far worse than supervised methods [11,12].

In this paper, we start by exploring sentence similarity in multi-views for cleaning the widely used WikiLarge dataset to build a refined WikiLarge. The motivation is that the source and the target sentence in a pair should have consistent semantic meanings for TS. Furthermore, most errors are on the target side of the WikiLarge dataset [3]. Comparing the similarities of source and target sentences makes it easy to filter out misalignment, noise, and copies, which will remove most error pairs.

Note that our method only removes the error pairs by using different views instead of correcting them. Therefore, the refined dataset will be smaller than the original dataset. The idea can also be extended to other

[*] Corresponding author at: Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, United Kingdom.
E-mail addresses: wei.he1@sheffield.ac.uk (W. He), k.farrahi@soton.ac.uk (K. Farrahi), bin.chen@sheffield.ac.uk (B. Chen), bpeng10@sheffield.ac.uk (B. Peng), a.villavicencio@sheffield.ac.uk (A. Villavicencio).

https://doi.org/10.1016/j.patrec.2023.11.011
Received 23 July 2023; Received in revised form 29 October 2023; Accepted 8 November 2023
Available online 10 November 2023



(a) Traditional Pre-training and Fine-tuning

(b) Continue Fine-tuning

Fig. 1. Two different fine-tuning strategies. Blue is the pre-training objective, yellow is our objective, red is the auxiliary objective which is close to our objective. Taking the representations trained with the auxiliary objectives will make the model convergence easier.

Table 1
Training datasets summary in number of pairs.

	Train set	Validation set	Test set
WikiLarge	289,043	2000	359
Newsela	94,208	1129	1076

Table 2
Examples of error pairs with misalignment, noise or exact copy in the WikiLarge dataset. Example 1 demonstrates misalignments, as the source and target are unrelated. Target sentences in Examples 2 to 4 are examples of noise in the target. Example 5 is a copy where the source and target are identical.

1	Source	They take up oxygen in the lungs or gills and release it while squeezing through the body's capillaries.
	Target	Red blood cells are very large in number; in women, there are 4.8 million red blood cells per microliter of blood.
2	Source	Many Major League alumni have called Northern League teams home in an effort get back to the Majors.
	Target	Catskill Cougars-LRB-/O2000/O-RRB-
3	Source	The Greater Berlin Act was passed by the Prussian parliament on 27 April 1920 and came into effect on 1 October of the same year.
	Target	Pankow
4	Source	Because fronts are three-dimensional phenomena, frontal shear can be observed at any altitude between surface and tropopause, and therefore be seen both horizontally and vertically.
	Target	Low Level Jets.
5	Source	On July 11, 2007, the first new episode of Danny Phantom was aired on the Nicktoons Network.
	Target	On July 11, 2007, The first new episode of Danny Phantom was aired on the Nicktoons Network.

tasks for cleaning parallel datasets, such as those for paraphrasing and text style transfer tasks.

A fine-tuning strategy called continue-fine-tuning is proposed to assist our model training by taking the text representation from other similar tasks, assuming that adopting text representations from more similar tasks can make the training easier, as illustrated in Fig. 1.

Furthermore, a new decoding strategy exclusive to TS is also explored. The traditional decoding strategy chooses the best candidates with the maximum likelihood and can overfit, while our strategy is more nuanced and takes word simplicity into account. Our strategy is more likely to choose candidates that are both correct and simple, which can improve the overall accuracy of the decoding process.

Our contributions are as follows: (1) We first propose a method using sentence representations in multi-views to refine the WikiLarge dataset, which can significantly improve the dataset's quality. (2) We propose the continue-fine-tuning strategy of taking text representations from similar tasks, which speeds up model fine-tuning and achieves good results. (3) We propose a new decoding strategy for simple text generation. With the structure of the paper: To contextualize them, we start with a discussion of related (Section 2), the methodology (Section 3), and experimental conditions (Section 4). The results (Section 5) and discussions (Section 6) are followed by the conclusions (Section 7).

2. Related work

The way data is represented is crucial for the performance of the model [13]. Pre-training language models on vast corpora have been found to obtain good word representations for downstream tasks [14]. Fine-tuning pre-trained models has achieved outstanding performance in TS, just like in other NLP tasks [1]. ACCESS [5] fine-tunes the BART model [15] with pre-defined prefixes. As one of the main issues of training the Seq2Seq text simplification model is the lack of high-quality parallel data, some methods also introduce pre-generated pseudo-parallel sentences to augment the training data [8,16], which achieved excellent unsupervised results. On the other hand, a recent study found that factual accuracy needs to be considered in evaluations [17].

Although many recent studies treat text simplification as a monolingual translation and fine-tuning PTMs has achieved good performance, other Seq2Seq deep learning models have also had an impact in the TS field. The Neural Semantic Encoders by Vu et al. [18] proposes an extension of this architecture by using augmented memory. That learning from multi-view data [19,20] and information fusion [21] has been widely employed in many tasks. Guo et al. [22] introduced multi-task learning with related auxiliary tasks of entailment and paraphrase generation in this architecture. A Transformer-based model [23] developed by Zhao et al. [24] integrated external paraphrase knowledge; the authors claim it could utilize real-world simplification rules. To avoid directly copying whole sentences and to make the output more diverse when applying generic Seq2Seq simplification models, Kriz et al. [25] first incorporated content word complexities and secondly generated a re-ranking system for generated candidate simplifications, which improved the automatic evaluation results.

3. Methodology

We first refine the WikiLarge dataset to create subsets of this dataset using sentence similarity of different views. We then feed these datasets to fine-tune models with our proposed strategy.

3.1. WikiLarge dataset cleaning

As a Wikipedia-based dataset, WikiLarge is regarded as the most widely used training dataset for text simplification. Many studies [9,26,27] work on this dataset, despite there being many misaligned and noisy sentence pairs (see Table 2). In order to filter out some of these

Table 3
Examples of correct pairs in WikiLarge dataset. These examples also demonstrate that the degree of simplification varies across examples.

1	Source	There is manuscript evidence that Austen continued to work on these pieces as late as the period 1809 (when she was 36) and that her niece and nephew, Anna and James Edward Austen, made further additions as late as 1814.
	Target	There is some proof that Austen continued to work on these pieces later in life. Her nephew and niece, James Edward and Anna Austen, may have made further additions to her work in around 1814.
2	Source	When Japan earned another race on the F1 schedule ten years later, it went to Suzuka instead.
	Target	When Japan was added back to the F1 schedule ten years later, it went to Suzuka instead.
3	Source	It is by far the longest of the Pauline epistles, and is considered his "most important theological legacy".
	Target	Here, the letter is addressed to the early Church in Rome.

error pairs, we explore methods in two different views to measure the similarity between the source and target sentence.

The first is an explicit method that compares the token edit distance between the source and target. It is motivated by the observation that a simplified sentence should have a small token edit distance against the original (See examples in Table 3). In contrast, noises and misalignments show a substantial difference explicitly against their source sentences (See examples in Table 2).

Motivated by works on feature selection [28–30], the second proposed method is an implicit method (or model-based method) based on measuring sentence representation similarity (SRS) by using the Sentence-BERT (*SBERT*) [31]. Intuitively, noise or misalignment targets differ from their source sentences in terms of SRS score, while exactly copied pairs will have the inner SRS score as 1.

3.1.1. Measuring similarity by the token edit distance method

Edit distance traditionally quantifies character-level changes from one sequence to another [32]. For two sequences a and b with lengths i and j, respectively, the edit distance can be defined in Eq. (1). In this work, following the settings in [3], we compute the number of changes between the original and simplified sentences through the token edit distance at the token level. To make the results comparable across text, we divide the number of changes by the original text length and obtain values from 100% (no changes) to 0% (completely different sentence).

$$\mathrm{lev}_{a,b}(i,j)=\begin{cases}\max(i,j) & \text{if } \min(i,j)=0,\\ \min\begin{cases}\mathrm{lev}_{a,b}(i-1,j)+1\\ \mathrm{lev}_{a,b}(i,j-1)+1\\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i\neq b_j)}\end{cases} & \text{otherwise.}\end{cases} \tag{1}$$

3.1.2. Measuring similarity by sentence representations

Sentence similarity is usually obtained by calculating the similarity of sentence representations. Previous methods typically transform word representations by a mean pooling operation to create semantic representations of the input sequence [33]. The pooling operation takes the mean of all token representations and compresses them into a single vector space to create a sentence vector. They then take sentence vectors and calculate the respective similarities between different vectors using the respective measurements.

For downstream tasks, PTMs from the BERT [34] family encode the meaning of sentences into densely packed representations, where similarities among sentences can be computed [31]. We explore calculating the cosine similarity of two representations in a source-simplified sentence pair. The architecture is illustrated in Fig. 2. *SBERT* is a siamese network architecture that can derive fixed-sized vectors for input sentences. *SBERT* adds a mean pooling operation to the output of BERT to derive a fixed-sized sentence embedding. The semantic similarities of sentences can be found by calculating fixed-sized sentence embeddings with similarity measures like cosine-similarity or Manhattan/Euclidean distance. The similarity represents the correlation of a pair of sentences. We use similarity to filter out sentence pairs with less correlation or exact copy.

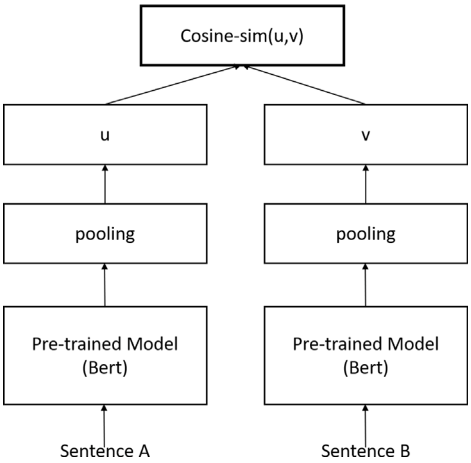


Fig. 2. The structure of calculating the sentence similarity uses Sentence BERT representations with cosine similarity. u and v are sentence embeddings. Note that the pre-trained models are fixed, and the process has no parameter update.

3.2. Model

Previous research has compared different model structures, and results indicate that encoder–decoder outperforms encoder-only and decoder-only architectures in many sequence-to-sequence tasks [35]. Our method uses the BART encoder–decoder denoising language model to conduct fine-tuning. We also reuse the representations of well-fine-tuned summarization models to conduct continue-fine-tuning.

3.3. Continue fine-tuning with text representations from similar tasks.

Usually, the pre-trained objective of the base model and the fine-tuning objective are different from each other [1]. For example, the pre-training objective of the BART model is sentence denoising, which is different from downstream tasks such as summarization and translation. It will increase the difficulty of fine-tuning convergence. We propose a continue-fine-tuning method, which can further fine-tune a model that has already been fine-tuned with similar objectives. The BART-based summarization models are chosen as base models for TS fine-tuning. It is because summarization and TS tasks are similar to each other [36]. In other words, the knowledge learned from the summarization task could be shared with the TS task to help its model convergence, as it is illustrated in Fig. 1.

Experimental results in Section 5.2.2 show that this technique decreases the training time and makes the model converge easier. We take BART-based models to continue fine-tuning with the refined datasets. The results show that our data cleaning method can efficiently remove misalignment and noise data pairs to improve the model's performance. Furthermore, reusing the representations of the model pre-trained in a TS-related task will help the TS task.

3.4. Decoding method

The decoding strategy for traditional text-to-text methods normally maximizes the likelihood with beam search. Basically, the decoding text quality relies on the features of the training corpus. This method avoids using the TS task-specific training corpus, so we propose a tailored searching space (TSS) for text simplification, which can effectively control the lexical simplicity of the output.

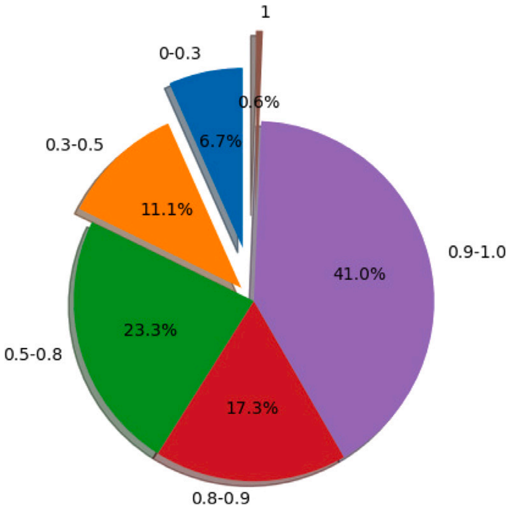


Fig. 3. The proportion of WikiLarge sentence pairs in terms of different ranges of *SBERT* similarity scores.

3.4.1. Generation with a tailored searching space

The answer space can be separated into two sub-spaces based on word frequency. The pre-trained model is trained on the whole vocabulary V, but on prediction, we only use $V^{(s)}$, the simple-word and named entities subset of V. Let $p' = \sum_{y \in V^{(s)}} P(y_i \mid y_{1:i-1}, x)$.

In the implementation, we set the candidate words from the low-frequency subset as 0 probability (excluding named entities), forcing the generation search to only occur on the high-frequency word space:

$$P'\left(y_i \mid y_{1:i-1}, x\right) = \begin{cases} P\left(y_i \mid y_{1:i-1}, x\right)/p' & \text{if } y_i \in V^{(s)} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In this way, we use a simple-word-only subset of the original vocabulary when generating text. It is based on the intuition that a complex (less frequent) expression can be replaced by its simple (more frequent) counterpart.

3.4.2. Comparing with other sampling strategies

Top-k sampling and Nucleus sampling have recently become popular sampling procedures [37]. Although TSS samples from truncated neural language model distributions, some differences exist. First, the TSS works on directed generation, in which the output is constrained by the input instead of open-ended generation. Second, TSS chooses the vocabulary subset $V^{(s)}$ based on word frequency, which is a proxy of lexical simplicity, while the other two strategies are based on candidate probability.

4. Experiments

We report the implementation details and experimental settings in this section.

4.1. Data cleaning

We follow the settings in [3] to calculate the edit distance.

To obtain sentence representations, we choose a fine-tuned model, '*all − mpnet − base − v2*', which is a model with good sentence representations in the *SBERT* pre-trained model repository.[1] We first feed each sentence into the *SBERT* model to get its sentence representation. Then, we apply cosine similarities to get the similarity score of the two sentences in each pair. We take the sentence-transformers library[2] to implement this procedure. Fig. 3 illustrates the proportion of WikiLarge sentence pairs in terms of different ranges of BERT sentence similarity. The score represents the similarity between each source and target pair. It is used to filter out sentence pairs with low similarity.

4.2. Model fine-tuning details

We implement our model training with the Transformers library [38]. For continue-fine-tuning, we choose a text summarization model *bart-large-cnn-samsum* from the HuggingFace model repository.[3] The learning rate is set to $5e{-}5$. Other hyper-parameters follow the settings in the *config* file of *bart-large-cnn-samsum* model. In the model decoding part, the word frequency list from the *GloVe* [39] embedding vocabulary[4] is used to estimate lexical complexity. Our results are evaluated by the text simplification evaluation tool *easse*[5] with a wide range of metrics. In order to make a fair comparison across different fine-tuning strategies, we train (or fine-tune) each proposed model at 27 epochs with an academic budget (A GeForce RTX 3090 GPU card in our case).

4.3. Comparison methods

Two supervised methods are reported for comparison baselines in our experiments. The first is ACCESS [5], which uses a transformer-based Seq2Seq model for training from scratch; the second is a translation-based method SBMT [40]. Both comparison methods are trained with the WikiLarge dataset without augmentation or supplementary data. Choosing these models is considered a fair comparison. Large language models showcase great performance in many NLP tasks. We also introduce a large language model (LLM) GPT-3 [41], with proper prompts as a further comparison. The prompt for the LLM is '*Please simplify the following text while preserving the core meaning*'.

5. Results and analysis

We calculate the SARI [40] score of two evaluation datasets, namely ASSET [42] and TurkCorpus [40], to evaluate the text simplification performance. Other reference-free automatic evaluation metrics used are Flesch Kincaid Grade Level (FKGL) [43], exact matching (EM) rate, and lexical complexity (LC) score to estimate the complexity of the generated outputs. All metrics above are implemented in the *easse* framework [44].

5.1. Data cleaning results

Data cleaning aims to filter out error pairs and retain the correct pairs. In our method, the error pairs are defined by sentence pair inner similarity scores, which is a hyper-parameter. Different separations are summarized in Table 4 and Fig. 3. In Fig. 3, we observe that 6.7% of total pairs have very low similarity scores (range from 0 to 0.3) in terms of *SBERT* similarity. There are also 11.1% total pairs with a relatively low score in the range $[0.3 - 0.5)$. The rest of the portions are 23.3% in range $[0.5, 0.8)$, 17.3% in range $[0.8, 0.9)$ and 41.0% in range $[0.9 - 1.0)$. It is worth noting that 0.6% of total pairs have exactly the same meanings as their target is an exact copy of the source. These pairs are also regarded as errors and filtered out to refine the training data.

On the other hand, from Table 4 we can see that the WikiLarge dataset can have various separations by different sentence similarity metrics. For example, the number of pairs at the 15% lowest similarity is 84,451 in terms of edit distance, while in terms of *SBERT* similarity, the number goes to 44,460.

[1] https://www.sbert.net/docs/pretrained_models.html.

[2] https://www.sbert.net/.

[3] https://huggingface.co/philschmid/bart-large-cnn-samsum.

[4] https://nlp.stanford.edu/projects/glove/.

[5] https://github.com/feralvam/easse.

Table 4
Statistics of WikiLarge dataset refinements. We first rank sentence pairs according to sentence similarity and then filter out the top least similar pairs by percentages. Numbers of filtered-out and retained pairs are also reported in this table.

Similarity type	Percentage	Removed number	Retained number
Edit distance	15%	84,451	211,951
	10%	29,640	266,762
	5%	14,821	281,581
	1%	2965	293,437
SBERT	40%	99,902	196,500
	30%	76,986	219,416
	20%	59,280	237,122
	15%	44,460	251,942
Non (Full Dataset)	0%	0	296,402

5.2. Model performance

The result of the model fine-tuned with different refined data is reported in the following sections.

5.2.1. Fine-tuning with refined dataset

In Table 5, it is evident that fine-tuning methods are better than the training-from-scratch methods in terms of ASSET's SARI score, FKGL score, and LC score, even though they are training with the same dataset WikiLarge.

The best fine-tuning SARI score of TurkCorpus 40.08 goes to the BART_SUM model fine-tuning with WikiLarge *BERT_sim_15%*, which is the continue-fine-tuning in our work. The best comprehensive performance goes to the above model with the new decoding strategy with a SARI score of ASSET at 41.75, FKGL at 6.84, and LC at 7.44. Our best result is not only better than the GPT-3 fix prompt methods, which is a significantly larger model than our chosen BART-based model, but also outperforms other training-based comparison methods.

Overall, models fine-tuned with the refined WikiLarge are better than the models trained with full WikiLarge considering the presented metrics in Table 5. It reveals that our data refining method can effectively clean the WikiLarge dataset and be used to better follow-up training results.

5.2.2. Results of continue-fine-tuning

The Continue-fine-tuning strategy tends to reuse the text representation of a related model to reduce the training time and enables the model to converge more easily. Results are reported in Table 6 and Fig. 4. We can see that the *BART-large-cnn-samsum* model obtains model convergence results in SARI at 40.08 by using only 12 epochs training, and with further training, the model tends to overfit. In contrast, the original *BART-large* needs to train 27 epochs to get a model convergence result with the same dataset, even if they have the same model architecture. The reason is that the text representations of the summarization task are similar to those of the text simplification task, so the summarization knowledge could be shared with the TS task to help its model convergence.

5.2.3. Results with decoding strategy

The traditional decoding strategy chooses the best candidates with the maximum likelihood. Our strategy gives priority to simple-word candidates when decoding. Table 7 illustrates that the vocabulary size of the decoding search space can significantly affect the model performance.

5.3. Decoding strategy analysis

Constraining the answer space will avoid using complex-word candidates when generating text. However, those candidates may have a higher probability of the original pre-trained language model. The smaller the search space, the more suboptimal candidates will be accumulated. When the original language model's candidates are insufficient, constraining its answer space (i.e., blocking some candidates) will harm the performance. Thus, constraining the answer space may not be suitable for small language models. Our current method sacrifices flexibility but to a limited extent. There is a trade-off between using high-frequency words and sentence simplification. A sentence with rare words can hardly be treated as simple. This method is not a simple lexical substitution. It is to replace complex text as ⟨*mask*⟩ to trigger the pre-trained language model to reconstruct the sentence. Although this might not always be the case, there is a strong prima facie case that using frequently used words is likely to improve comprehension. Using frequently used words makes this approach very straightforward to implement, and can be seen as an advantage of this approach.

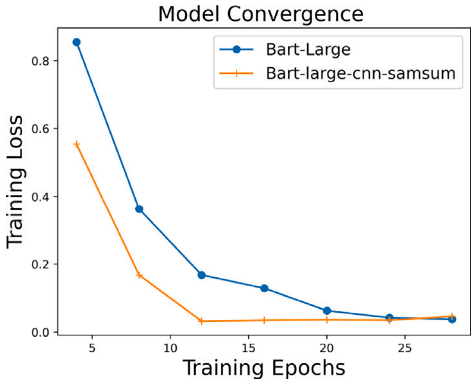


Fig. 4. The model convergence trends with the increasing of training epochs.

6. Discussion

We propose a method to refine the widely used text simplification dataset WikiLarge, and the experimental results confirm that our data refining method can improve the dataset quality. However, there are some limitations to this method. First, the method only filters out dissimilar pairs while retaining similar ones, even though a target sentence similar to the source may not necessarily be its simplification. Errors in date and time would not change the semantic meaning of the sentences significantly. In this case, our data clean method may not detect these errors.

The second is that even though the refined dataset has a higher quality, the pair numbers of the refined dataset are too small as a training set for a good transformer-based Seq2Seq model (See statistics in Table 4). In addition, the best proportion of pairs that should be filtered out for the downstream task is uncertain. These issues motivate us to explore a method without a large amount of parallel data or to find a way to collect more high-quality parallel sentences. Future work will also explore other pre-fine-tuning models with tasks related to TS.

7. Conclusions

In this paper, we propose a parallel dataset multi-view refining method by multi-view sentence representation similarity for the text simplification task. The result shows that the refining method is beneficial to improve data quality for model training. We also propose a continue-fine-tuning strategy by retaking the text representations from a text simplification-related task to help the model converge faster. Furthermore, the task-specific decoding strategy boosts the model performance on text simplification.

Table 5
Comparison of automatic evaluation metrics of different methods. Training dataset $Edit\_5\%$ means the dataset is built by filtering out 5% least similar pairs of WikiLarge by edit distance, while $BERT\_sim\_5\%$ means doing the same thing by *SBERT* similarity. Bold fonts highlight the best results.

Name	Training dataset	SARI↑		FKGL↓	EM↓	LC↓
		ASSET	TurkCorpus			
Training from scratch method						
SBMT	WikiLarge Full	37.11	39.56	7.95	0.10	8.03
ACCESS	WikiLarge Full	40.13	**41.38**	7.29	**0.04**	7.94
Fine-tuning method						
BART_large	WikiLarge Full	37.30	39.06	8.35	0.20	8.19
BART_large	Edit_5%	38.02	39.65	7.66	0.15	8.15
BART_large	Edit_10%	38.99	39.83	7.95	0.17	8.14
BART_large	Edit_15%	38.56	39.10	8.11	0.24	8.21
BART_large	BERT_sim_5%	38.12	39.76	7.86	0.15	8.15
BART_large	BERT_sim_10%	38.50	39.30	7.51	0.11	8.15
BART_large	BERT_sim_15%	38.91	39.83	7.45	0.17	8.14
BART_SUM	BERT_sim_15%	38.20	40.08	7.75	0.14	8.19
BART_SUM + Decoding	BERT_sim_15%	**41.75**	39.71	**6.84**	0.12	**7.44**
GPT-3	BERT_sim_15%	40.77	39.72	8.46	0.16	8.20

Table 6
Comparison of the fine-tuning result of different initial models in the TurkCorpus test dataset. All models are trained on a refined WikiLarge dataset $BERT\_sim\_15\%$.

Initial model	SARI↑	Epoch	FKGL↓	LC↓
BART-large	38.59	12	7.95	8.03
	39.83	27	7.45	8.14
T5_base	38.76	27	8.19	8.16
BART-sum	40.08	12	8.35	8.19
	39.76	20	7.66	8.15
	39.30	27	7.51	8.15

Table 7
Comparison of the fine-tuning result of different decoding strategies in TurkCorpus dataset. The decoding searching space is divided by the vocabulary sizes of a word frequency list.

Voc size	SARI↑		FKGL↓	LC↓	Sen-Sim↑
	ASSET	TurkCorpus			
BART-large-cnn-samsum + BERT_sim_15					
1000	42.12	37.38	**6.08**	**7.05**	85.8%
2000	**42.16**	38.74	6.49	7.30	88.9%
3000	41.75	39.71	6.84	7.44	90.6%
5000	39.76	**39.84**	7.44	7.67	93.6%
8000	38.12	39.35	7.80	7.85	95.3%
15 000	38.08	38.61	8.28	8.03	96.5%
Full	39.40	39.94	8.65	8.17	**97.8**%

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Wei He reports financial support was provided by Engineering and Physical Sciences Research Council. Aline Villavicencio reports financial support was provided by Engineering and Physical Sciences Research Council.

Data availability

Data will be made available on request.

Acknowledgments

The work was partly supported by EPSRC, United Kingdom grant EP/T02450X/1.

References

[1] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, Sci. China Technol. Sci. (2020) 1–26.

[2] K.W. Church, Z. Chen, Y. Ma, Emerging trends: A gentle introduction to fine-tuning, Natl. Lang. Eng. 27 (6) (2021) 763–778.

[3] L. Vásquez-Rodríguez, M. Shardlow, P. Przybyła, S. Ananiadou, Investigating text simplification evaluation, 2021, arXiv preprint arXiv:2107.13662.

[4] X. Zhang, M. Lapata, Sentence simplification with deep reinforcement learning, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 584–594, http://dx.doi.org/10.18653/v1/D17-1062, URL https://aclanthology.org/D17-1062.

[5] L. Martin, É.V. de la Clergerie, B. Sagot, A. Bordes, Controllable sentence simplification, in: Proceedings of the 12th Language Resources and Evaluation Conference, 2020, pp. 4689–4698.

[6] W. Xu, C. Callison-Burch, C. Napoles, Problems in current text simplification research: New data can help, Trans. Assoc. Comput. Linguist. 3 (2015) 283–297.

[7] F. Alva-Manchego, C. Scarton, L. Specia, Data-driven sentence simplification: Survey and benchmark, Comput. Linguist. 46 (1) (2020) 135–187.

[8] L. Martin, A. Fan, É. de la Clergerie, A. Bordes, B. Sagot, Multilingual unsupervised sentence simplification, 2020, arXiv preprint arXiv:2005.00352.

[9] K. Omelianchuk, V. Raheja, O. Skurzhanskyi, Text simplification by tagging, in: Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, 2021, pp. 11–25.

[10] S. Narayan, C. Gardent, Unsupervised sentence simplification using deep semantics, 2015, arXiv preprint arXiv:1507.08452.

[11] S. Surya, A. Mishra, A. Laha, P. Jain, K. Sankaranarayanan, Unsupervised neural text simplification, 2018, arXiv preprint arXiv:1810.07931.

[12] J. Qiang, X. Wu, Unsupervised statistical text simplification, IEEE Trans. Knowl. Data Eng. 33 (4) (2019) 1802–1806.

[13] S. Zhang, J. Li, W. Zhang, Y. Qin, Hyper-class representation of data, Neurocomputing 503 (2022) 200–218.

[14] H. Wang, J. Li, H. Wu, E. Hovy, Y. Sun, Pre-trained language models and their applications, Engineering (2022).

[15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019, arXiv preprint arXiv:1910.13461.

[16] X. Lu, J. Qiang, Y. Li, Y. Yuan, Y. Zhu, An unsupervised method for building sentence simplification corpora in multiple languages, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 227–237.

[17] A. Devaraj, W. Sheffield, B.C. Wallace, J.J. Li, Evaluating factuality in text simplification, in: Proceedings of the Conference. Association for Computational Linguistics. Meeting, Vol. 2022, NIH Public Access, 2022, p. 7331.

[18] T. Vu, B. Hu, T. Munkhdalai, H. Yu, Sentence simplification with memory-augmented neural networks, 2018, arXiv preprint arXiv:1804.07445.

[19] R. Hu, J. Gan, X. Zhu, T. Liu, X. Shi, Multi-task multi-modality SVM for early COVID-19 diagnosis using chest CT data, Inf. Process. Manage. 59 (1) (2022) 102782.

[20] J. Gan, R. Hu, Y. Mo, Z. Kang, L. Peng, Y. Zhu, X. Zhu, Multigraph fusion for dynamic graph convolutional network, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[21] Y. Zhu, J. Ma, C. Yuan, X. Zhu, Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis, Inf. Fusion 77 (2022) 53–61.

[22] H. Guo, R. Pasunuru, M. Bansal, Dynamic multi-level multi-task learning for sentence simplification, 2018, arXiv preprint arXiv:1806.07304.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[24] S. Zhao, R. Meng, D. He, S. Andi, P. Bambang, Integrating transformer and paraphrase rules for sentence simplification, 2018, arXiv preprint arXiv:1810.11193.

[25] R. Kriz, J. Sedoc, M. Apidianaki, C. Zheng, G. Kumar, E. Miltsakaki, C. Callison-Burch, Complexity-weighted loss and diverse reranking for sentence simplification, 2019, arXiv preprint arXiv:1904.02767.

[26] J. Mallinson, A. Severyn, E. Malmi, G. Garrido, Felix: Flexible text editing through tagging and insertion, 2020, arXiv preprint arXiv:2003.10687.

[27] L. Martin, B. Sagot, E. de la Clergerie, A. Bordes, Controllable sentence simplification, 2019, arXiv preprint arXiv:1910.02677.

[28] R. Hu, D. Cheng, W. He, G. Wen, Y. Zhu, J. Zhang, S. Zhang, Low-rank feature selection for multi-view regression, Multimedia Tools Appl. 76 (2017) 17479–17495.

[29] L. Peng, Y. Mo, J. Xu, J. Shen, X. Shi, X. Li, H.T. Shen, X. Zhu, GRLC: Graph representation learning with constraints, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[30] Y. Mo, Y. Chen, Y. Lei, L. Peng, X. Shi, C. Yuan, X. Zhu, Multiplex graph representation learning via dual correlation reduction, IEEE Trans. Knowl. Data Eng. (2023).

[31] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, 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), 2019, pp. 3982–3992.

[32] G. Navarro, A guided tour to approximate string matching, ACM Comput. Surv. (CSUR) 33 (1) (2001) 31–88.

[33] X. Zhao, E. Durmus, D.-Y. Yeung, Towards reference-free text simplification evaluation with a BERT siamese network architecture, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 13250–13264.

[34] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2019, arXiv preprint arXiv:1910.10683.

[36] F. Zaman, M. Shardlow, S.-U. Hassan, N.R. Aljohani, R. Nawaz, HTSS: A novel hybrid text summarisation and simplification architecture, Inf. Process. Manage. 57 (6) (2020) 102351.

[37] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, in: International Conference on Learning Representations, 2019.

[38] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, 2019, arXiv preprint arXiv:1910.03771.

[39] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162, URL https://aclanthology.org/D14-1162.

[40] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing statistical machine translation for text simplification, Trans. Assoc. Comput. Linguist. 4 (2016) 401–415.

[41] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[42] F. Alva-Manchego, L. Martin, A. Bordes, C. Scarton, B. Sagot, L. Specia, ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4668–4679.

[43] J.P. Kincaid, R.P. Fishburne Jr., R.L. Rogers, B.S. Chissom, Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel, Technical Report, Naval Technical Training Command Millington TN Research Branch, 1975.

[44] F. Alva-Manchego, L. Martin, C. Scarton, L. Specia, EASSE: Easier automatic sentence simplification evaluation, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 49–54, http://dx.doi.org/10.18653/v1/D19-3009, URL https://aclanthology.org/D19-3009.