Exploiting Summarization Data to Help Text Simplification

Renliang Sun, Zhixian Yang, Xiaojun Wan
Wangxuan Institute of Computer Technology, Peking University
Center for Data Science, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{sunrenliang, yangzhixian}@stu.pku.edu.cn
wanxiaojun@pku.edu.cn

Abstract

One of the major problems with text simplification is the lack of high-quality data. The sources of simplification datasets are limited to Wikipedia and Newsela, restricting further development of this field. In this paper, we analyzed the similarity between text summarization and text simplification and exploited summarization data to help simplify. First, we proposed an alignment algorithm to extract sentence pairs from summarization datasets. Then, we designed four attributes to characterize the degree of simplification and proposed a method to filter suitable pairs. We named these pairs Sum4Simp (S4S). Next, we conducted human evaluations to show that S4S is high-quality and compared it with a real simplification dataset. Finally, we conducted experiments to illustrate that the S4S can improve the performance of several mainstream simplification models, especially in low-resource scenarios.

1 Introduction

Text simplification and text summarization are two major techniques aiming at improving text readability (Margarido et al., 2008). The main objective of text simplification is to reduce the complexity of the text while keeping its meaning unchanged (Alva-Manchego et al., 2020; Al-Thanyyan and Azmi, 2021). Text summarization is to summarize the main idea of the document in less space (El-Kassas et al., 2021).

One of the major problems of text simplification is the lack of high-quality aligned data, which is essential for training most simplification models. Existing text simplification datasets are derived from Wikipedia (Zhang and Lapata, 2017) and Newsela (Xu et al., 2015). Researchers have proposed various alignment algorithms to extract complex-simple sentence pairs from articles (Jiang et al., 2020). However, aligning sentences from only two corpora hinders the acquisition of more simplification data, which motivates us to explore new ways to address this problem.

Text simplification usually involves the operations of keeping, deleting, reordering, etc.(Xu et al., 2016) Text summarization does not require a summary to be a simple text. Nevertheless, when we analyzed the datasets of text summarization meticulously, we noticed that there are many instances where several sentences in the original document are merged into one sentence, and complex parts are rewritten, as shown in Table 1. Then, a question arises naturally: to what extent is text summarization correlated with text simplification? Furthermore, is it feasible to extract data from text summarization to help low-resource text simplification?

Example	
document	What's Hollywood's **role in all of this? The same as it has always been** – to make money.
summary	What does Hollywood **want?** To make money, **of course.**

Table 1: The bolded parts indicate that the complex sentence in the document has been rewritten.

In this study, we investigated the above problems with a three-step procedure: (1) Extract aligned sentence pairs from summarization datasets. (2) Select sentence pairs in which the source sentences have been simplified. (3) Evaluate the quality of these sentence pairs for text simplification.

To extract aligned sentence pairs from the summarization datasets, we proposed an alignment algorithm based on the similarity between sentences. Then, we designed four attributes and a method to filter sentence pairs suitable for text simplification. We performed human evaluations and conducted experiments using mainstream simplification models on these pairs to show that they are of high quality and can help simplification.

To summarize, our contributions include: (1) We are the first to exploit summarization data to help

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 39–51
May 2-6, 2023

text simplification, verifying a new source of simplification data. (2) We proposed an alignment algorithm and a method for filtering complex-simple sentence pairs. We named them Sum4Simp (S4S). (3) We performed both empirical analysis and human evaluations on S4S to verify its quality, and the experimental results with several simplification models show the benefits of S4S for text simplification. The S4S dataset and codes are released at https://github.com/RLSNLP/Sum4Simp.

2 Related Work

2.1 Simplification Models

Early text simplification models are mainly based on statistic machine learning (Wubben et al., 2012; Kauchak, 2013; Narayan and Gardent, 2014). In recent years, many scholars have proposed models based on deep learning technology, such as NTS(Nisioi et al., 2017), DRESS-LS(Zhang and Lapata, 2017), EditNTS(Dong et al., 2019), ACCESS(Martin et al., 2020a), which advance the development of text simplification.

2.2 Mine Data for Simplification

The above models require a large number of aligned texts for training. Nevertheless, text simplification is a low-resource problem. Some works aim at designing unsupervised models (Qiang and Wu, 2019; Surya et al., 2019; Kumar et al., 2020; Laban et al., 2021). While other works try to mine aligned sentence pairs from more data to help train the models. Martin et al. (2020b) proposed unsupervised mining technology to create multi-language simplification corpora automatically. Lu et al. (2021) used the back-translation approach to construct a large-scale pseudo sentence simplification corpus.

2.3 Relationship with Text Summarization

For a long time, studies on text simplification and text summarization have been conducted separately. Nevertheless, there exist circumstances where complex texts not related to the main idea are removed when summarizing a document, and multiple sentences can be compressed and rewritten into a single sentence. Such a summarization can also be regarded as a simplification. Ma and Sun (2017) proposed a semantic relevance-based model to improve the results of simplification and summarization. Zaman et al. (2020) pointed out some similarities between the two tasks and defined the new task of generating simplified summaries. Up to now, none of the work has specifically analyzed the relationship between summarization and simplification. It is still worth investigating whether the data from summarization can help simplification.

3 Mine Sentence Pairs for Simplification from Summarization Datasets

In this section, we will elaborate on how to extract sentence pairs that are suitable for text simplification from text summarization datasets. Text summarization is a document-level task while text simplification refers to a sentence-level task. Thus, we proposed an algorithm to extract aligned sentence pairs at first. Then, since not all aligned sentence pairs are suitable for text simplification, we chose four attributes and defined a set of rules to filter the appropriate sentence pairs. The whole process is shown in Figure 1.

3.1 Sentence Alignment Algorithm

Previous sentence alignment algorithms such as CATS (Štajner et al., 2018) aim at sentence compression (one complex sentence corresponds to one simple sentence) or sentence splitting (a complex sentence is split into several simple sentences). They do not satisfy the requirement to align sentence pairs from summarization datasets, where one sentence in the summary corresponds to multiple sentences in the document. Thus, we proposed an alignment algorithm to address this problem.

Assume that there are m sentences in the document and n sentences in the summary. For each sentence d_i in the document and each sentence s_j in the summary, we first compute the similarity between the two sentences. We use SBERT (Reimers and Gurevych, 2019) to achieve this. SBERT is a pre-trained model based on BERT (Devlin et al., 2019), in which the similarity of two input sentences will be calculated rapidly. Then, we define the upper threshold of similarity S_{max} and the lower threshold of similarity S_{min}. S_{max} is greater than S_{min} and they are in the range [0,1]. Assume that the maximum value of similarity between any sentence in the document and s_j is D_{max}. If D_{max} is greater than S_{max}, we consider that the sentence corresponding to D_{max} is very similar to s_j. Therefore, we keep s_j as the target sentence and the sentence corresponding to D_{max} as the source sentence, and they form an aligned sentence pair. If D_{max} is smaller than S_{min}, we consider that there is no sentence in the document that is similar to s_j.

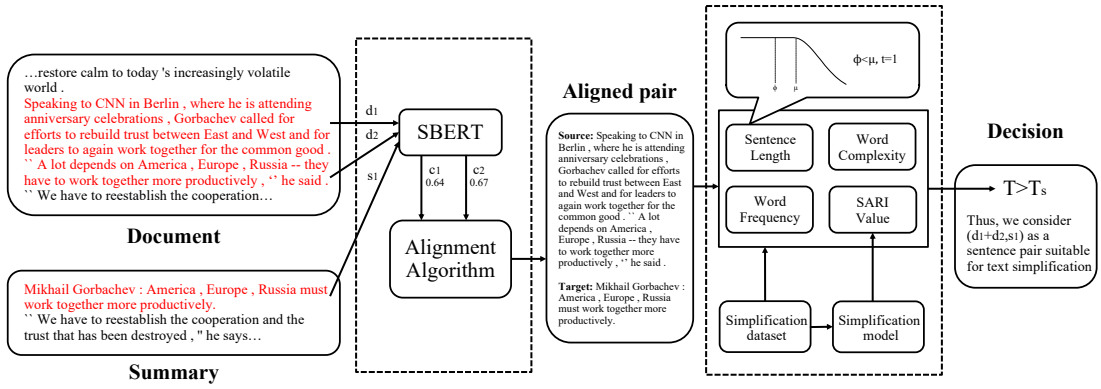

Figure 1: The process of mining suitable sentence pairs from summarization datasets.

Thus, we do not keep sentence pairs related to s_j.

Algorithm 1 Sentence alignment algorithm

```
Initialization: F and C are empty sets
for d_i in d_1,d_2,...,d_n do
    c_i = SBERT(d_i,s_j)
    C.append(c_i)
end for
if max(C)>S_max then
    F.append(corresponding d_i of max(C))
else if S_max>max(C)>S_min then
    F.append(corresponding d_i of max(C))
    C.remove(max(C))
    repeat
        c_i = SBERT(stitch(F,corresponding d_i of max(C)),s_j)
        if c_i>S_add then
            F.append(corresponding d_i of max(C))
            C.remove(max(C))
        end if
    until c_i ≤ S_add or len(C)≥ L_max
end if
Output: (F,s_j) as an aligned sentence pair
```

If D_{max} is greater than S_{min} and smaller than S_{max}, we consider this to be the case where multiple sentences in the document correspond to s_j. We temporarily save the sentences corresponding to D_{max}, and then find the sentence with the largest similarity among the remaining sentences of the document. We stitch this sentence with the sentence we just saved according to the order of the sentences in the document. We repeat this operation until the similarity between the stitched sentences and s_j is less than a threshold. We define this threshold as S_{add}, which takes values in the range [S_{min},S_{max}]. To prevent the problem of imbalance where the length of the source sentence far exceeds the length of the target sentence caused by extracting too many sentences from the document, we set L_{max}. When the number of stitched sentences reaches L_{max}, we save these stitched sentences as source sentences and s_j as the target sentence.

3.2 Four Attributes to Characterize Simplification

Aligned sentence pairs obtained from Algorithm 1 are not always complex-simple ones, and an example is given below:

Source sentence: Analysts say the Arab Spring has made Dubai a safe haven for people in the Middle East who worry about the turmoil elsewhere.

Target sentence: Analysts say the Arab Spring has made Dubai a safe haven for those who worry about the turmoil elsewhere.

This example is a real sentence pair mined from the summarization data. It is an aligned sentence pair but neither the attributive clause nor the complex words such as “turmoil” are simplified. Thus, it is not a good instance for text simplification. We design four attributes to characterize whether the source sentence is simplified or not, which are:

Sentence Length Intuitively, the longer the sentence, the more complex the sentence is likely to be. We calculate the length of the target sentence minus the average length of the source sentences.

Word Complexity We believe that the lower the average complexity of words, the simpler the sentence. We use a lexicon of word complexity created by Maddela and Xu (2018). Each word is scored by humans. The higher the score, the more complex the word. We calculate the value of the average word complexity of the target sentence minus the average word complexity of the source sentences.

Word Frequency Some words appear more frequently in complex sentences, while some words appear more frequently in simple sentences. The more frequently a word appears in a simple sentence, the more likely it is to be a simple one. We calculate the odds ratio (Monroe et al., 2008) to

represent the frequency of word occurrence. For two corpus, namely i and j, their sizes are n_i and n_j, respectively. For a word w, the occurrences in corpus i and corpus j are w_i and w_j, respectively. Then, the odds ratio r of word w between corpus i and corpus j can be defined as:

$$r = \frac{w_i/w_j}{n_i/n_j} \tag{1}$$

We use the simplification dataset to construct a dictionary containing the odds ratios of the words. For example, if we want to conduct experiments on WikiLarge (Zhang and Lapata, 2017), we calculate the odds ratio of the words occurring in the WikiLarge training set. We calculate the value of the average odds ratio of the target sentence minus the average odds ratio of the source sentence.

SARI Value SARI (Xu et al., 2016) is an essential evaluation method for text simplification. It takes the original sentence, the simplified sentence, and reference sentences into consideration. The SARI value is an average of F1 scores of add and keep operation and precision of delete operation. The score for each operation is obtained by averaging n-gram scores.

$$\begin{aligned} SARI &= \frac{1}{3}F_{add} + \frac{1}{3}F_{keep} + \frac{1}{3}P_{del} \\ P_{operation} &= \frac{1}{4}\sum_{n=1,2,3,4} p_{operation}(n) \\ R_{operation} &= \frac{1}{4}\sum_{n=1,2,3,4} r_{operation}(n) \\ F_{operation} &= \frac{2 \times P_{operation} \times R_{operation}}{P_{operation} + R_{operation}} \\ operation &\in [add, keep, del] \end{aligned} \tag{2}$$

We consider the source sentence of the aligned sentence pairs as the original sentence and the target sentence as the simplified sentence. We need to train a simplification model at first. For example, we trained a model like ACCESS (Martin et al., 2020a) on the WikiLarge training set. Then, we input the source sentences into the simplification model and generate simplified sentences. These simplified sentences are used as reference sentences. Finally, the SARI values are calculated.

3.3 Quantify Simplicity and Filter Suitable Sentence Pairs

For each attribute, we propose a method to quantify the simplicity of a sentence. Our method is based on a hypothesis: a reference simplification dataset performs approximately normally distributed on each attribute. Simplification datasets can contain hundreds of thousands of instances, in line with the concept of large samples in statistics. Therefore, we believe this hypothesis is reasonable.

Take the sentence length attribute as an example. We first calculate the mean μ and standard deviation σ of the sentence length of the training set of a reference dataset (e.g. WikiLarge). For a random variable X, the probability density function $f(x)$ can be obtained. If the ratio of sentence length for a sentence pair is ϕ, its score t on this attribute is:

$$t = \begin{cases} 1, & \phi <= \mu \\ 2 \times (0.5 - \int_{\mu}^{\phi} f(x)dx), & \phi > \mu \end{cases} \tag{3}$$

$$t = \begin{cases} 2 \times (0.5 - \int_{\phi}^{\mu} f(x)dx), & \phi < \mu \\ 1, & \phi >= \mu \end{cases} \tag{4}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{5}$$

The mathematical significance is that if $\phi <= \mu$, the simplification degree of the sentence pair is greater than the average simplification degree of the simplification dataset on this attribute. Thus, we give a score of 1 to t. If $\phi > \mu$, we subtract the proportion of sentence pairs with a ratio greater than μ and lower than ϕ that is in the simplification dataset. Then, we perform a normalization operation to obtain t. For attributes sentence length (len), word complexity (comp), and word frequency (freq), a lower ϕ indicates a greater degree of simplification. We use Equation (3) to calculate t. For attribute SARI value (sari), a higher ϕ indicates a greater degree of simplification. We use Equation (4) to calculate t.

To make a final decision, the scores on each attribute are weighted with α and summed to obtain T for a sentence pair, indicating the extent of simplification of the source sentence. We set a threshold value $\mathrm{T_s}$ to control the extent of simplification. When $\mathrm{T} > \mathrm{T_s}$, we consider the sentence pair suitable for the task of text simplification.

$$\begin{aligned} \mathrm{T} &= \sum_{i \in Attr} \alpha_i t_i \\ Attr &= [len, comp, freq, sari] \end{aligned} \tag{6}$$

We exploit and filter sentence pairs from the CNN/Daily Mail summarization dataset (Nallapati

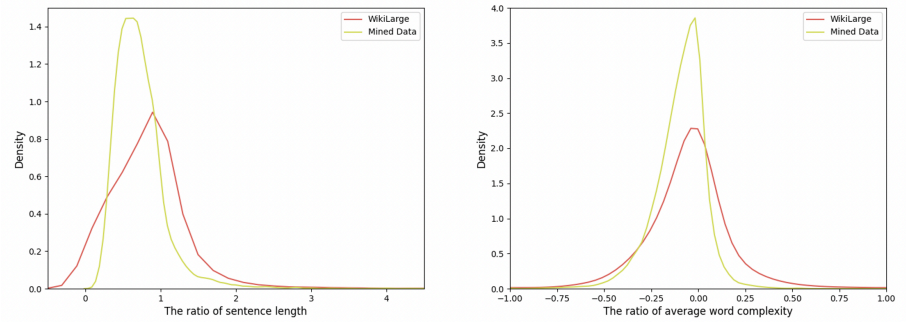

Figure 2: Distributions of the ratio of sentence length and average word complexity. We smoothed the results by using a Gaussian kernel. Sentences from S4S are more compressed than in WikiLarge. Sentences where the words become more complex are also less than in WikiLarge.

et al., 2016), which contains more than 300,000 documents and corresponding summaries from news stories in CNN and Daily Mail. We name these mined sentence pairs Sum4Simp (S4S).

4 Quantitative Analysis

In this section, we want to show that Sum4Simp (S4S) is high-quality. We conducted two human evaluations and performed statistics on S4S, comparing it with real simplification datasets.

4.1 Human Evaluations

First, we want to evaluate the alignment quality of the sentence pairs obtained in Section 3.1. Following Hwang et al. (2015), we defined the quality of alignment into four classes: Good, Good partial, Partial, and Bad. Due to the space limit, details and examples are demonstrated in Table 10.

We randomly selected sentence pairs from the aligned pairs obtained by our proposed alignment algorithm. Then, we designed a baseline that does not use our proposed alignment algorithm. When the similarity calculated by SBERT between a sentence in the document and a sentence in the summary is greater than 0.6, we kept this sentence in the document. As we introduced in Section 3.1, the CATS method (Štajner et al., 2018) may not be suitable for aligning sentence pairs from summarization datasets. However, we used it as a baseline.

We used the two baseline methods described above to obtain aligned sentence pairs from summarization datasets. What's more, we randomly selected sentence pairs from a simplification dataset named WikiLarge (Zhang and Lapata, 2017) for comparison. The results are shown in Figure 3.

We considered **Good** and **Good partial** to be acceptable quality. The sentence pairs obtained by

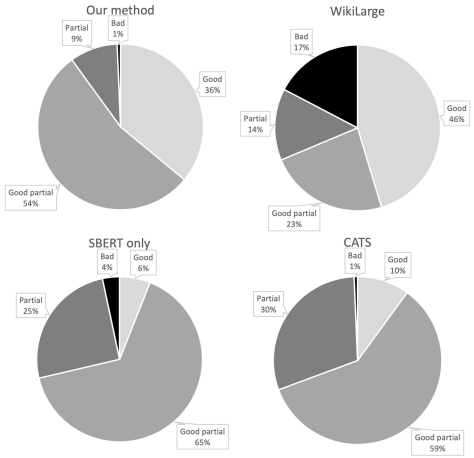

Figure 3: Human evaluation results of data obtained by three alignment methods and WikiLarge. We randomly selected 50 sentence pairs from each source of data. Then, we hired three workers to evaluate the 200 sentence pairs individually.

our proposed alignment algorithm have the highest percentage in these two levels. While WikiLarge has the most sentence pairs with a Good level, it also has the most sentence pairs with a Bad level. Xu et al. (2015) pointed out that data mined from Wikipedia is not always of high quality.

Then, we want to show that the final sentence pairs obtained in Section 3.3 are more suitable for simplification. We randomly selected 50 sentence pairs that are only aligned and 50 sentence pairs from S4S. We also randomly selected 50 sentence pairs from WikiLarge for comparison.

Following Dong et al. (2019), we used two indicators as the criteria: (1) **Simplicity**: Is the target sentence simpler than the source sentence? (2) **Adequacy**: Are the source sentence and target sentence fluent and grammatically correct? Another indicator, Meaning, can be regarded as the eval-

uation of alignment quality, so we did not repeat it. The results are shown in Table 2. The sentence pairs from S4S receive the highest Simplicity score, significantly higher than the aligned-only pairs and WikiLarge, indicating the effectiveness of the proposed filtering method.

	Simplicity↑	Adequacy↑
WikiLarge	3.11**	4.6**
Aligned only	3.2**	4.81
S4S	**3.49**	**4.94**

Table 2: Human evaluation results of data obtained by two methods and WikiLarge. We hired three workers to evaluate individually. Student t-tests were performed and results significantly different from S4S were marked with **(p<0.01).

4.2 Statistics and Comparison

We used three dimensions, sentence length, average word complexity, and odds ratio of cue words, to compare the sentence pairs from S4S with those from WikiLarge. The ratio of sentence length is calculated by dividing the length of the simplified sentence by the length of the original sentence. The ratio of average word complexity is calculated by subtracting the average word complexity of the original sentence from the average word complexity of the simplified sentence.

We randomly selected 10,000 sentence pairs from WikiLarge and S4S, respectively. From Figure 2, in S4S, the number of sentence pairs with a length ratio greater than one has been significantly decreased compared to WikiLarge, indicating that sentences are more compressed. What's more, the vast majority of the ratios of average word complexity are less than zero, suggesting a general simplification at the word level in S4S.

Sentence splitting, a common operation in text simplification, can be represented by the odds ratio of conjunctions and cue words (Siddharthan, 2003). The definition of the odds ratio is detailed in Equation (1). When the odds ratio of conjunctions is much less than 1, and the odds ratio of cue words is much greater than 1, a complete degree of simplification is involved. Following Xu et al. (2015) and Sun et al. (2021), we calculated the odds ratio of conjunctions and cue words in WikiLarge and S4S, as shown in Table 3.

WikiLarge		S4S	
cue words	odds ratio↑	cue words	odds ratio↑
also	1.15	also	1.13
then	1.16	then	**1.21**
still	1.01	still	**1.41**

Wikilarge		S4S	
conjunctions	odds ratio↓	conjunctions	odds ratio↓
and	0.87	and	0.95
as	0.72	as	0.80
since	1.01	since	**0.96**
because	2.59	because	**1.05**
when	1.32	when	**1.09**
if	1.30	if	1.38
but	1.18	but	**1.11**
though	0.71	though	**0.62**
although	0.46	although	**0.40**

Table 3: The odds ratio of cue words and conjunctions. The bolded parts indicate that S4S performs better than WikiLarge. Some words, such as "hence", occur too infrequently to be statistically meaningful.

5 Experimental Setup

5.1 Datasets

We used two commonly used simplification datasets, **WikiLarge** (Zhang and Lapata, 2017) and **WikiSmall** (Zhu et al., 2010), to demonstrate the usefulness of the sentence pairs mined from summarization data. The training set of WikiLarge contains more than 296k sentence pairs, which is larger than that of WikiSmall containing 88k sentence pairs. We used **Turkcorpus** (Xu et al., 2016) as the validation and the test set for WikiLarge. Each of the 2000 validation instances and the 359 test instances has 8 reference sentences. We used the original validation set and test set for WikiSmall, with 205 validation instances and 100 test instances.

5.2 Evaluation Metrics and Models

We took **SARI** (Xu et al., 2016) and **BERTScore** (Zhang et al., 2019) as the evaluation metric in this paper. SARI is the most popular automatic evaluation metric for text simplification. The SARI value is obtained by averaging the F_{keep}, P_{delete}, and F_{add} score. We used the **EASSE** package (Alva-Manchego et al., 2019) to get SARI values. A recent study recommends using BERTScore$_{precision}$ to evaluate the quality of the system outputs prior to using SARI to measure simplification (Alva-Manchego et al., 2021). FKGL (Kincaid et al., 1975) was used to measure text readability but was proven to be inappropriate for evaluating text simplification recently (Tanprasert and Kauchak,

Models	WikiLarge				S4S				WikiLarge+OA				WikiLarge+S4S			
	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}
Transformer	36.95*	70.80	36.91	3.15	34.43**	58.54	43.68	1.08	36.75*	70.79	36.38	3.06	**37.85**	71.11	39.15	3.27
BART	37.99**	72.53	37.85	3.59	36.21**	64.70	42.60	1.34	37.71**	73.02	36.81	3.31	**39.20**	70.99	42.31	4.30
ACCESS	39.67*	71.20	42.69	5.12	36.20**	65.62	41.53	1.44	39.46*	69.39	43.96	5.03	**40.71**	71.26	44.06	6.81

Models	WikiSmall				S4S				WikiSmall+OA				WikiSmall+S4S			
	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}
Transformer	36.35*	66.69	40.53	1.82	36.75	60.23	49.49	0.53	36.38*	64.46	40.54	4.15	**38.57**	66.56	43.69	5.46
BART	35.13*	64.94	35.86	4.59	34.13*	61.06	39.95	1.39	34.65*	67.09	31.92	4.93	**36.58**	67.39	37.14	5.22
ACCESS	35.35*	65.01	38.50	2.53	34.63**	51.07	51.76	1.05	35.67*	60.95	44.29	1.77	**38.28**	58.45	53.64	2.73

Table 4: Results of three simplification models trained on four different training sets. The test sets in the upper and lower tables are Turkcorpus and WikiSmall, respectively. "+" represents the operation to mix the two datasets and sort them randomly. OA is a set of sentence pairs with a similar size to S4S drawn from aligned but not filtered sentence pairs. The bolded part indicates the training set that achieves the best result for each model. Student t-tests were performed, and SARI values that were significantly different from WikiLarge+S4S and WikiSmall+S4S were marked with *(p<0.05) or **(p<0.01).

2021). BLEU (Papineni et al., 2002) has been proven to be unsuitable for evaluating text simplification (Sulem et al., 2018). Therefore, we did not report FKGL values and BLEU values.

We selected three representative models - **Transformer** (Vaswani et al., 2017), **BART** (Lewis et al., 2020), and **ACCESS** (Martin et al., 2020a) to conduct experiments. Transformer and BART perform strongly for many generation tasks. ACCESS is a simplification model proposed recently and it uses explicit tokens related to different attributes to control the process of simplification.

5.3 Training Details

We used the Huggingface Transformers (Wolf et al., 2020) to implement the Transformer model and the BART model. We used the original code to implement the ACCESS model. We used four Nvidia A40 GPUs for training. We reported the results of the model on the test set which has the best SARI value on the validation set.

More details can be found in Appendix A.

6 Experimental Results

6.1 Results on Existing Test Sets

We designed four types of training sets and tested the three simplification models on existing test sets. We first measured the outputs of each model using BERTScore$_{precision}$ and found that the values are very close to 1, indicating that the outputs are of high quality. Then, the SARI values are shown in Table 4.

From the upper table, Sum4Simp (S4S) mixed with the WikiLarge training set improves the performance of all three simplification models on Turkcorpus. To be more specific, in terms of the SARI metric, ACCESS is improved by 1.04 points, BART is improved by 1.21 points, and Transformer is improved by 0.90 points. We have used the original codes and followed the original hyper-parameter settings, but the SARI value of the ACCESS model trained on WikiLarge is lower than the results reported by Martin et al. (2020a). We think this is because we lowered the training data and used the NLTK package to split the words. Meanwhile, seen from the lower table, S4S mixed with the WikiSmall training set also improves the performance of all three models on the test set of WikiSmall. The improvement on the WikiSmall test set is more significant than that on the Turkcorpus test set. In terms of the SARI metric, ACCESS is improved by 2.93 points, BART is improved by 1.45 points, and Transformer is improved by 2.22 points. Example outputs are given in Table 11. It may seem strange that the SARI value of Transformer is higher than that of BART. However, we noticed that the SARI value of BART is approximately 3 points higher than that of Transformer on the validation set, making the experimental results remain convincing.

The size of the training set of WikiLarge is much larger than that of WikiSmall. Therefore, the models were more fully trained on WikiLarge. While the size of the training set of WikiSmall is comparatively smaller, S4S helps the model learn to simplify sentences better and results in a more significant improvement.

OA was designed to verify that the improvement of the results comes from high-quality mined sentence pairs rather than mere data expansion. Compared with the original training set, the per-

Models	S4S				WikiLarge				S4S+WikiLarge			
	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}
Transformer	**44.75**	53.32	74.72	6.19	32.59	45.38	51.78	0.61	43.61	52.24	73.91	4.68
BART	46.42	57.20	76.62	5.43	32.98	47.12	50.10	1.70	**46.51**	57.24	73.91	4.68
ACCESS	**40.19**	45.85	72.82	1.88	30.10	44.30	43.99	2.01	38.45	43.35	70.71	1.30

Table 5: Results on three simplification models trained on three different training sets. The valid and test sets come from S4S.

formances on WikiLarge+OA and WikiSmall+OA were not improved and even dropped for the model like BART. The results illustrate that the method for filtering suitable sentence pairs for simplification purposes is essential.

If we only used S4S as the training set, the SARI values obtained are 2.5 points lower than the model trained with WikiLarge and 0.5 points lower than the model trained with WikiSmall on average. We believe the performance gap is due to domain differences: S4S comes from news stories written by professional journalists, while WikiLarge and WikiSmall come from Wikipedia. Overall, though S4S comes from a different domain, it can still be beneficial to the existing simplification datasets.

6.2 Results on S4S Test Set

In this subsection, we treat S4S as a standard simplification dataset that contains more than 243K sentence pairs. We divided the train/dev/test set as 240k/2k/1k, respectively. We would like to see the performance of simplification models on the S4S dataset and we want to know if the WikiLarge dataset from a different domain can improve the performance. We designed three types of training sets. Then, we conducted experiments with each of them to train the three simplification models.

According to Table 5, all three simplification models trained on the S4S dataset have significantly higher SARI values compared to the results in Table 4. When we mixed the training set of S4S and WikiLarge, the SARI values dropped by 1 point on average compared to using the S4S training set alone. Besides, when we only used the WikiLarge training set, the SARI values dropped by an average of more than 10 points. We also gave example outputs in Table 12. Above all, we believe the quality of the S4S dataset is higher than that of the Wikipedia-based datasets. The S4S dataset was given in the supplementary materials.

6.3 Results on Extremely Low-resource Scenarios

In many cases simplification data is hard to obtain (Aprosio et al., 2019; Maruyama and Yamamoto, 2019), and we took a small amount of sentence pairs from the training set of WikiLarge to simulate an extremely low-resource situation. We reduced the size of the WikiLarge training set to 50%, 20%, 10%, 5%, and 1%, respectively. We then conducted experiments using the ACCESS model trained on the size-reduced WikiLarge data and the mixture of size-reduced WikiLarge and S4S. The results are shown in Figure 4.

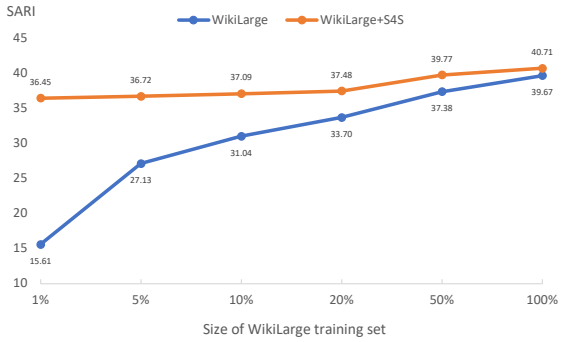

Figure 4: Experimental results of extremely low-resource experiments on Turkcorpus test set.

When the size of the training set is relatively small (less than 20%, about 60,000 sentence pairs), S4S can improve the results significantly. The results prove that the S4S is effective in helping text simplification when data is difficult to obtain.

6.4 Ablation Study

In our proposed sentence filtering method, we used four attributes to control the simplicity of the sentence pairs extracted from summarization datasets. We removed the attributes one by one and then used the remaining three attributes as new rules to filter simple sentence pairs. We set T_s to 2.75 in the experiment. The filtered sentence pairs are mixed with the WikiLarge training set and then used to train the ACCESS model.

Experiment	SARI↑
WikiLarge+S4S	40.71
WikiLarge	39.67
Without word complexity	39.32(-1.39)
Without sentence length	39.63(-1.08)
Without word frequency	37.70(**-3.01**)
Without SARI value	38.78(-1.93)

Table 6: Ablation study on Turkcorpus test set.

The results are illustrated in Table 6. In this experiment, the odds ratio attribute has the greatest effect on the results. When this attribute is missing, the SARI value decreases by 3.01 points. The sentence length attribute has the least effect on the results. When this attribute is missing, the SARI value drops by 1.08 points. The results also show that the four attributes of our design are meaningful. They all play a significant role in filtering the simplified sentence pairs.

7 Conclusion

In this paper, we are committed to mining data from text summarization datasets to help text simplification. We proposed an alignment algorithm and a new method to filter suitable sentence pairs. We named these pairs Sum4Simp (S4S). We conducted human evaluations on S4S and performed experiments on mainstream simplification models to illustrate that the S4S is high-quality and can help text simplification. In future work, we will apply our method to mine more simplification data from other summarization datasets.

Acknowledgements

This work was supported by National Key R&D Program of China (2021YFF0901502), National Science Foundation of China (No. 62161160339), State Key Laboratory of Media Convergence Production Technology and Systems and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

Limitations

We considered the consumption of computational resources as the major limitation of our method. To extract aligned sentence pairs from summarization datasets, we need to calculate the similarity between each sentence in the summary and each sentence in the document, which makes the time complexity of the alignment algorithm be $O(n^2)$. We ran the alignment algorithm with an Intel Xeon processor. On average, there are 40 sentences in a document and 4 sentences in a summary. There are 312K documents in total with corresponding summaries. The total running time is 42,153s. We have released the aligned sentence pairs to help future research.

Second, to calculate the SARI values in Section 3.2, we need to train a simplification model in advance, which can consume GPU resources. For example, if we train a BART model on the WikiLarge dataset and set the max epochs to 10, the training time spent on an Nvidia A40 is about 3 hours.

References

Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. Easse: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54.

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

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un) suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Alessio Palmero Aprosio, Sara Tonelli, Marco Turchi, Matteo Negri, and Mattia A Di Gangi. 2019. Neural text simplification in low-resource conditions using weak supervision. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 37–44.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnts: An neural

programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

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.

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.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. 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.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6365–6378.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xinyu Lu, Jipeng Qiang, Yun Li, Yunhao Yuan, and Yi Zhu. 2021. An unsupervised method for building sentence simplification corpora in multiple languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 227–237.

Shuming Ma and Xu Sun. 2017. A semantic relevance based neural network for text summarization and text simplification. *arXiv preprint arXiv:1710.02318*.

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

Paulo RA Margarido, Thiago AS Pardo, Gabriel M Antonio, Vinícius B Fuentes, Rachel Aires, Sandra M Aluísio, and Renata PM Fortes. 2008. Automatic summarization for text simplification: Evaluating text understanding by poor readers. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, pages 310–315.

Louis Martin, Éric Villemonte De La Clergerie, Benoît Sagot, and Antoine Bordes. 2020a. Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020b. Multilingual unsupervised sentence simplification. *arXiv preprint arXiv:2005.00352*.

Takumi Maruyama and Kazuhide Yamamoto. 2019. Extremely low resource text simplification with pre-trained transformer language model. In *2019 International Conference on Asian Language Processing (IALP)*, pages 53–58. IEEE.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *The 52nd annual meeting of the association for computational linguistics*, pages 435–445.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91.

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.

Jipeng Qiang and Xindong Wu. 2019. Unsupervised statistical text simplification. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1802–1806.

Nils Reimers and Iryna Gurevych. 2019. 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)*, pages 3982–3992.

Advaith Siddharthan. 2003. Preserving discourse structure when simplifying text. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003) at EACL 2003*.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. Cats: A tool for customized alignment of text simplification corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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

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

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068.

Teerapaun Tanprasert and David Kauchak. 2021. Flesch-kincaid is not a text simplification evaluation metric. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024.

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.

Farooq Zaman, Matthew Shardlow, Saeed-Ul Hassan, Naif Radi Aljohani, and Raheel Nawaz. 2020. Htss: A novel hybrid text summarisation and simplification architecture. *Information Processing & Management*, 57(6):102351.

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

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.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.

A More Details

In Algorithm 1, for S_{max}, S_{add}, and S_{min}, we first observed the alignment results to obtain a rough range [0.5,0.8]. In this range, we set the step size to 0.1 and then chose four combinations of parameters: (0.8, 0.7, 0.6), (0.8, 0.7, 0.5), (0.8, 0.6, 0.5), and (0.7, 0.6, 0.5). We used human evaluation on 50 sentence pairs for each combination to determine which combination is the best. Finally, we set S_{max} to 0.8, S_{min} to 0.6, and S_{add} to 0.7. Lmax is set to 3 as an empirical value. If it is too large, the model will be more concerned with deletion than simplification; if it is too small, the information in the original sentences will lose.

For the method of filtering suitable sentence pairs in Section 3.3, we set α_i to 0.25 because it is difficult to prove that one of the four attributes is more important than the other. We performed a parameter research for $\mathrm{T_s}$ from 3.5 to 3.8 with a step size of 0.05.

We have released the aligned sentence pairs obtained in Section 3.1 for future research. So future researchers only need to set $\mathrm{T_s}$ when conducting experiments.

To obtain Table 4, we first trained models with existing simplification datasets (e.g., train ACCESS with WikiLarge). Then, we selected the model that performed best on the validation set to calculate the score t for the SARI value attribute mentioned in Section 3.2. In this way, we got S4S. We then trained models with WikiLarge+S4S to obtain the results in the fourth column of the Table 4. The S4S dataset in Section 6.2 is obtained after we first trained ACCESS with WikiLarge. We will also release this version of S4S as a standard simplification dataset.

Parameter	Value	Parameter	Value
epochs	30	max source length	256
batchsize	64	max target length	256
optimizer	Adam	dropout	0.1
learning rate	5e-5	d_{model}	768
warm up steps	2000	attention heads	12

Table 7: Parameters of the Transformer model.

Parameter	Value	Parameter	Value
epochs	10	max source length	256
batchsize	64	max target length	256
optimizer	Adam	dropout	0.1
learning rate	5e-5	d_{model}	768
warm up steps	2000	attention heads	12

Table 8: Parameters of the BART model.

Parameter	Value	Parameter	Value
max epochs	100	label smoothing	0.54
max tokens	5000	clip norm	0.1
optimizer	Adam	dropout	0.2
learning rate	1.1e-4	weight decay	1e-4
warm up steps	1000	attention heads	8

Table 9: Parameters of the ACCESS model.

B Definition of Alignment Quality

C Example Outputs

Good	The semantics of the source sentence and the target sentence completely match, possibly with small omissions.
Source Target	Sets in children 's bedrooms or left on as background noise could be particularly damaging. Devices in bedrooms or left on as background noise is more damaging.
Good partial	Source and target sentence mean basically the same thing. However, source or target sentence may contain additional information that is not contained in the other sentence.
Source Target	The tape was played at a hearing Monday to determine whether or not the confession can be used as evidence at Hernandez 's murder trial - not whether the statements are true. Judge Maxwell Wiley must decide whether Hernandez was properly advised of his rights. The judge must decide not whether the confession is true, but whether it can be permitted to be used as evidence at Hernandez 's murder trial.
Partial	Source and target sentence are discussing two unrelated concepts, but share short related phrases that do not match considerably.
Source Target	A non-profit group called Women On 20s, formed to convince President Barack Obama to put a woman's image on the $20 note, already has done some polling. There is a group called Women On 20s.
Bad	Source and target sentence are discussing two unrelated concepts.
Source Target	Leicester City have lost just one of their last seven league meetings with Hull City. 88 % of British grandmothers consider themselves to be a Glam-Ma.

Table 10: Definition of the alignment quality. Example of each level of quality is also given.

Complex(input)	in computing , a protocol is a set of rules which is used by computers to communicate with each other across a network .
Simple(reference)	in computing , a protocol is the language used by computers while talking with each other .
WikiSmall	in computing , a protocol is a set of rules which is used by computers to communicate with each other across a network .
S4S	the process is a set of rules which is used by computers to communicate with each other across a network
WikiSmall+OA	in computing , a protocol is a set of rules which is used by computers to provide with each other across a network .
WikiSmall+S4S	in computing , a protocol is used by computers to communicate with each other across a network .

Table 11: An example of sentences generated by ACCESS. When the training set is WikiSmall, the complex sentence is not simplified. When the training set is S4S or WikiSmall+OA, the generated sentences contain grammatical errors and change the meaning of the complex sentence. The sentence generated by ACCESS trained on WikiSmall+S4S can be regarded as a simplified sentence.

Complex(input)	barcelona manager luis enrique -lrb- pictured -rrb- insisted afterwards he was right to start uruguay striker suarez
Simple(reference)	barcelona boss luis enrique says he was right to start the uruguay player
S4S	barcelona boss luis enrique says he was right to start uruguay striker
WikiLarge	barcelona manager luis enrique -lrb- pictured - pictured he wanted to start uruguay striker suarez .
S4S+WikiLarge	barcelona manager luis enrique said he was right to start right to start uruguay suarez suarez

Table 12: An example of sentences generated by ACCESS when S4S is regarded as a standard simplification dataset. When the training set is WikiLarge, the generated sentence contains grammatical errors and changes the meaning of the complex sentence. When the training set is S4S+WikiLarge, the generated sentence also contains grammatical errors and is less simple than the generated sentence when the training set is S4S only. This example illustrates that the quality of S4S is higher than that of WikiLarge.