Improving Factual Consistency in Abstractive Summarization with Sentence Structure Pruning

Dingxin Hu[1], **Xuanyu Zhang**[2], **Xingyue Zhang**[1], **Dongsheng Chen**[1]
Yiyang Li[1], **Marina Litvak**[3,4], **Natalia Vanetik**[3], **Qing Yang**[2]
Dongliang Xu[2], **Yingqi Zhu**[1], **Yuze Li**[1], **Yanquan Zhou**[1], **Lei Li**[1]
Beijing University of Posts and Telecommunications[1], Du Xiaoman Financial[2]
Sami Shamoon College of Engineering[3], Karlsruhe Institute of Technology[4]
leili@bupt.edu.cn

Abstract

State-of-the-art abstractive summarization models still suffer from the content contradiction between the summaries and the input text, which is referred to as the factual inconsistency problem. Recently, a large number of works have also been proposed to evaluate factual consistency or improve it by post-editing methods. However, these post-editing methods typically focus on replacing suspicious entities, failing to identify and modify incorrect content hidden in sentence structures. In this paper, we first verify that the correctable errors can be enriched by leveraging sentence structure pruning operation, and then we propose a post-editing method based on that. In the correction process, the pruning operation on possible errors is performed on the syntactic dependency tree with the guidance of multiple factual evaluation metrics. Experimenting on the FRANK dataset shows a great improvement in factual consistency compared with strong baselines and, when combined with them, can achieve even better performance. Code and data are availabel at `https://github.com/Anthonyhu2333/SSC`.

Keywords: abstractive summarization, factual inconsistency problem, post-editing method

1. Introduction

Abstractive summarization models can generate a concise summary that captures the salient ideas of the article(Lewis et al., 2019; Zhang et al., 2020). However, researchers find that these models are prone to generating non-factual and sometimes entirely fabricated content(Cao et al., 2018; Goodrich et al., 2019; Maynez et al., 2020).

Recently, various approaches have been proposed to evaluate or improve the factual consistency of generated summaries. There are two ideas for improving factual accuracy: directly correcting potentially incorrect summaries (post-editing methods) or avoiding models from producing incorrect summaries (model-level methods). Model-level methods like filtering out improper training data are effective in improving factual consistency. However, the post-editing method is even more crucial as it is not only model-independent but also can serve as the final step in ensuring factual consistency.

Various post-editing methods have been proposed recently, but they share a common flaw. Most of these methods use entities as the correction targets and replacement as the correction means. Due to fact that the intrinsic cause of factual errors is not simply the use of the wrong words but also an error in the collocation of different concepts in the sentence, these methods fail to identify and modify incorrect content hidden in the sentence structures, and the replacement operation proves insufficient for certain categories of errors. An incorrect summary generated by the abstractive summarization model is shown in Table 1, demonstrating the limitations of the ER (**E**ntity **R**eplacement) method. A comprehensive examination of this issue will be provided in Section 4.2.

Source	Text
Document	[...] The Art Fund, which led the campaign, will gift the collection to the Victoria and Albert Museum before it is loaned to the Wedgwood Museum. [...]
Incorrect Summary	A collection of art worth more than 100,000 has been donated to a County Durham Museum.
SSP	A collection of art has been donated to a County Durham Museum.
SSP+ ER Method	A collection of art has been donated to the Victoria and Albert Museum.

Table 1: Example of an incorrect summary cannot be corrected by replacement method in the FRANK dataset. The generated summary has an incorrect modifier "worth more than 100,000" and an incorrect target "a County Durham Museum". The value of art is never mentioned in the document. As a result, the first error can not be corrected by the ER method. This example also demonstrates that our approach has a correction ability different from the ER method and can work with the existing ER method to achieve greater correction capability.

In this paper, we propose a post-editing method named SSP (**S**entence **S**tructure **P**runing method), which takes sentence structures as the correction targets and prunes the errors under the guidance of factual evaluation metrics. The main contributions are highlighted as follows:

(1) We are the first, to the best of our knowledge, to propose sentence structure-level post-editing methods. We identify one limitation of previous post-editing methods on summarization factual consistency and prove the necessity of sentence structure pruning operation.

(2) We introduce a post-editing method to compensate for the deficiency of strong baselines and achieve SOTA performance in the FRANK dataset. We further analyze in detail what kind of sentence parts are removed in the correcting process and prove the soundness of our methods.

(3) We have widely evaluated the effects of our method working together with existing methods for improving factual consistency. The experimental results show that our method can help both post-editing and model-level methods further improve factual consistency.

2. Related work

2.1. Factual evaluation metrics

Commonly employed metrics for assessing summary quality, such as Rouge (Lin, 2004), primarily focus on attributes such as fluency and conciseness but do not encompass an evaluation of factual consistency. Existing factual evaluation metrics can be broadly categorized into three distinct types based on their fundamental ideas: natural language inference (Kryściński et al., 2019; Laban et al., 2022), syntactic dependency relationships (Goyal and Durrett, 2021), and QA or cloze models (Durmus et al., 2020; Nan et al., 2021; Li et al., 2022).[1] Due to the differences in implementation, later in the paper, we find through experiments that these metrics also vary in their sensitivity to different types of errors.

2.2. Existing post-editing methods and their limitations

All existing post-editing methods use entities as the correction target, and a huge portion of them are ER methods. Fabbri et al. (2022) applies a sentence-compression dataset to facilitate the training of a model designed to remove inaccurate entities. Apart from that, (Zhu et al., 2020; Cao et al., 2018; Balachandran et al., 2022; Dong et al., 2020; Chen et al., 2021) are all RE methods that correct errors by replacing incorrect entities. These methods have the following limitations.

Firstly, it is far from enough to correct errors only at the entity level. A large portion of the errors are not entity-related, and many incorrect summaries do not even contain recognizable entities. **Secondly**, most errors cannot be corrected by replacement means. For example, an event is given the wrong occurrence time in the summary, while the correct time of occurrence is never mentioned in the article. To correct such errors in the absence of introducing external knowledge, we can only delete the incorrect content.

In our opinion, it is more appropriate to correct errors at the SSU(**S**entence **S**tructure **U**nit) level. The nodes and relations in the syntactic dependency tree cover all the text and represent all the semantic information in the sentences. Any subtree of a syntactic dependency tree can be viewed as an SSU. As a result, it is the SSU that carries all the incorrect content. The sentence structure pruning operation has the ability to flexibly adjust SSU and is an effective way to correct errors.

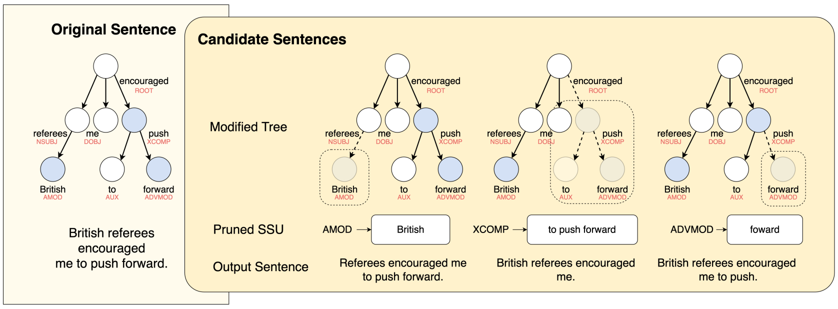
2.3. Model-level methods

The research focus of model-level methods has undergone a shift in recent years from model structure to model training data. Falke et al., 2019 added a candidate selection mechanism in the summary generation process to improve the factual consistency of generated summaries. In recent years, more work has been proposed to ensure the quality of the training data. The training data from the abstractive summarization datasets such as XSUM(Narayan et al., 2018) are known to contain hallucinations in gold summaries(Maynez et al., 2020). Goyal and Durrett, 2021; Wan and Bansal, 2022 filter and correct the training data to improve the factual consistency. Chaudhury et al., 2022 have proved that these improvements can work in conjunction with post-editing methods to further improve the factual consistency of generated summaries. We will later compare our method with them and evaluate their joint work.

2.4. Structure pruning operation

Structure pruning operation is a widely used method of modifying sentences. To get a more concise summary, Gagnon and Da Sylva (2005) prune sentence structures labeled with targeted relations. Subsequent work(Filippova and Strube, 2008; Perera and Kosseim, 2013) also applies sentence pruning to make controllable modifications to the sentence. These papers demonstrate that the sentences obtained by this method maintain a certain level of grammatical accuracy and readability.

[1] EntFA (Cao et al., 2021) is also an important work. However, it is not incorporated into our analysis due to its reliance on posterior probabilities for factual accuracy evaluation, which can no longer accurately be obtained after the pruning operation.



R (Relation)	$\mathbf{SSU_R}$
NSUBJ	British referees
AMOD←	British
DOBJ	me
XCOMP←	to push forward
AUX	to
ADVMOD←	forward

Figure 1: It shows how to get pruned summaries. We first list all SSUs in the summary. Pruning SSUs marked with ← has a smaller probability of causing grammatical errors. We prune the subtrees separately and obtain the pruned summaries. The definition of dependency relations can be seen at Description.

3. Methods

3.1. Identification of prunable SSUs

To guarantee grammatical accuracy in the corrected summaries while simultaneously optimizing computational efficiency, we adopt the following two steps to identify prunable SSUs.

Firstly, we identify specific dependency relations associated with prunable SSUs. Due to the intricate nature of the grammar rules, the utilization of quantitative metrics becomes imperative for informed decision-making. We randomly select 1000 reference summaries from the XSUM dataset, followed by the pruning of SSUs governed by diverse dependency relations. Subsequently, we quantify their impact on grammatical accuracy (**IGA**) by counting the number (**n1**) of linguistically unacceptable sentences generated after pruning with the help of a related evaluation method and normalizing it by the number (**n2**) of effective pruned samples ($\mathrm{IGA} = \mathrm{n1}/\mathrm{n2}$). The larger the value of the IGA, the more likely the pruning operation for related SSU will result in a grammar error.

We pick the top 20 syntactic dependencies that have the least impact on grammatical accuracy.[2] Common linguistic knowledge acknowledges that the pruning of certain modifiers and clauses will not break the grammatical correctness, which is consistent with the type of selected relations.

Secondly, we use linguistically acceptability metric as a final guarantee of grammatical accuracy. The process of obtaining the pruned summaries is illustrated in Figure 1. Not all the pruned summaries will be added to the candidate set. Rather, an additional evaluation is conducted to assess the grammatical accuracy of each pruned summary, and exclusively, only those deemed grammatically accurate are added to the candidate set.

3.2. Strategies to combine multiple factual evaluation metrics

Different factual evaluation metrics (as described in Section 4.4) differ in their sensitivity to errors due to their various implementation forms. Through experiments in Figure 2, we demonstrate that the correlation between different metrics is insignificant, and single evaluation metrics may have insensitivity problems to specific factual errors. Different metrics can complement each other, and combining multiple metrics in a joint evaluation is meaningful.

We use the strategy of **voting** to combine multiple metrics. Specifically, a consensus is reached when all five metrics vote that the current summary has superior factual accuracy (the number of valid votes is not less than 5). The determination of the requisite number of valid votes is determined through a comparative experimental study. The presently employed strategy has exhibited superior performance with regard to factual accuracy. The results of the experiment will be placed in the appendix when permitted.

Additionally, we want to explain the reasons why we don't adopt the combination strategy that weights scores given by different metrics. Firstly, although most of the factual metrics give scores between 0 and 1, there are large differences in the distribution of scores. This problem can not be solved by normalization or weighting. Secondly, some metrics will only give discrete scores of 0 or 1, such as FactCC (Kryściński et al., 2019), which makes changes in the scores of some metrics decisive for the final score.

3.3. Correction process

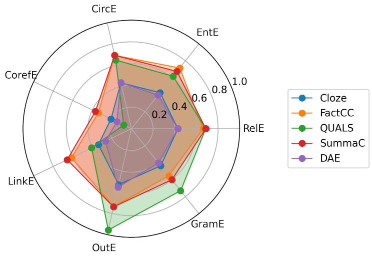
The error correction process is divided into two steps: candidate summaries **generation**, and output summary **selection**. The specific error correction process can be found at Algorithm 1.

Generation: To capture all possible candidate summaries, we go through the various possible results of sentence structure pruning. As shown in

[2] Only the top 20 are chosen because the statistics indicate that pruning the SSUs lead by rest grammatical dependencies has a high probability of causing grammatical errors (more than 20%). Selection results and statistics can be seen in Appendix A

	SummaC	ClozE	FactCC	DAE	QUALS
SummaC	1.0 *	0.0754 *	0.0567	0.1029 *	0.0409
ClozE	\	1.0 *	0.0807 *	0.3258	-0.0837 *
FactCC	\	\	1.0 *	0.2178 *	-0.0440
DAE	\	\	\	1.0 *	-0.0241
QUALS	\	\	\	\	1.0 *

(a)



(b)

Figure 2: Person correlation coefficients between different evaluation metrics and their sensitivity to different error types. The summaries sampled from the FRANK dataset as the evaluation targets to calculate the person correlation coefficient used in (a) are generated based on XSUM. Values marked with * are statistically significant with p < 0.05. We find that there is no strong correlation between the metrics. (b) are carried out on the FRANK dataset, in which errors are classified. Values are obtained from the deviation of the mean scores from different evaluation metrics and a constant. Higher values mean the metrics are more sensitive to this type of error.

Figure 1, we traverse each node in the dependency tree in a Pre-order traversal[3]. After checking for grammatical accuracy, suitable summaries will be added to the candidate set.

Selection: We use a variety of factual evaluation metrics to pick up the summaries with the best factual accuracy. By using the strategy of combining factual evaluation metrics presented in Section 3.2, we select the summaries that are the most factual as the final output summary.

Algorithm 1 Sentence structure pruning method

```
Initialize evaluation metrics $E$, dependency tree $D$. Initialize output $O$, same as summary $S$.
scores = [metric.score($S$) for metric in $E$]
for $R$ in $D$ do
    if $R$ is prunable then
        $S'$ = $S$.prune($SSU_R$)
        if $S'$ is not grammatical accurate then
            continue
        end if
        scores' = [metric.score($S'$) for metric in $E$]
        if scores is better than scores' then
            continue
        end if
        scores=scores'
        $O$=$S'$
    end if
end for
return $O$
```

* SSU_R means a SSU led by the syntactic dependency relation R

4. Experiments

4.1. Dataset and dependency parsing method

We conduct our experiments on the XSUM dataset (Narayan et al., 2018). This dataset consists of articles from the British Broadcasting Communication (BBC) and a one-sentence summary of the article. We also report results on the CNN/DM dataset (Nallapati et al., 2016) to show that our analysis generalizes across datasets. For testing, we use a subset of these datasets contained in the FRANK dataset (Pagnoni et al., 2021) that combines incorrect outputs from 9 models with a total of 2250 annotated model outputs, which can be used to benchmark factuality metrics.

We use Spacy[4] for syntactic dependency parsing. Spacy is a widely used tool for syntactic dependency parsing. It achieves state-of-the-art accuracy of over 95% on the parsing task[5]. Our approach also introduces linguistic acceptability metrics to ensure that the very small probability of errors introduced by Spacy does not result in grammatically incorrect summaries being output.

4.2. Validation of limitation for entity replacement methods

To prove the limitation of the entity replacement, we first count the number of incorrect summaries containing identifiable entities. As shown in Tabel 2, 22.5% sentences in FRANK do not even have recognizable entities. Moreover, for those error summaries that contain identifiable entities, the entities are not necessarily the cause of the errors.

We then randomly select 100 summaries from the FRANK dataset and artificially correct them

[3] We additionally do controlled experiments to prove that traversal order has virtually no effect on the results.

[4] https://spacy.io/

[5] https://spacy.io/usage/facts-figures#benchmarks

Dataset	Total Number	No-E Number	Percentage
F-C	3915	921	23.5%
F-X	1027	192	18.7%
FRANK	4942	1113	22.5%

Table 2: The percentage of incorrect summary sentences with no recognizable entities. F-C represents the FRANK dataset sampled from CNN/DM, and F-X represents the Frank dataset sampled from XSUM. No-E Number represents the number of summary sentences with no recognizable entities.

only using entity replacement. For each case, rigorous efforts are made to correct errors, and the extent to which these corrections reduce errors is evaluated. The annotation result can be seen in Table 5. 47% of the incorrect summaries can be partially corrected, and only 21% of those can be completely corrected by entity replacement.

As a comparison, we also use sentence structure pruning to correct errors. By introducing it, the percentage of errors that post-editing methods can correct significantly increases. The number of FI (summaries where **F**actual accuracy can be **I**mproved) increases by **74%** (83% relative to 47%), and the number of CC (summaries that can be **C**ompletely **C**orrected) increases by **186%** (60% relative to 21%) than only applying entity replacement in our samples, which further proves the necessity of adopting sentence structure pruning.

4.3. Post-editing methods baselines

The following baselines are used to compare the performance and complement ability of our sentence structure pruning (SSP) methods:

BART: Research in factual error correction first begins with Zhu et al. (2020), with their contributions specifically in using data augmentation methods to train a model based on the UniLM (Dong et al., 2019) to output the corrected summaries directly. Cao et al. (2020) use the BART (Lewis et al., 2019) model instead and achieve better performance.

Factedit: Factedit (Balachandran et al., 2022) improves the data enhancement process and can generate more representative synthetic examples of non-factual summaries.

Compedit: By using the training data outputted by a sentence-compression model, Fabbri et al. (2022) further improve the data enhancement process and make the correction model trained on these data has the ability to remove erroneous entities. However, it still has the limitation of only using entities as correction targets, and discrepancies between the dataset constructed using data augmentation and the real incorrect summaries generated by the summarization model can affect the model's performance.

SpanFact: Dong et al. (2020) use a QA-based model named SpanFact to check and correct entity words in the summary sentence. Codes of this model are not available, and we reproduce them with the help of Transformers[6] library.

Cogcomp: Chen et al. (2021) generate alternative candidate summaries where entities in the generated summary are replaced with ones with compatible semantic types from the source document. They train a discriminative correction model named Cogcomp to correct errors by selecting candidate summaries with the best factual accuracy.

4.4. Evaluation metrics

In this section, we summarize the evaluation metrics that are used in the experiments. All evaluation metrics are implemented through publicly available code and weighting files.

Rouge: Rouge (Lin, 2004) is a common metric for evaluating fluency and conciseness for text generation tasks, including abstractive summarization. It measures the overlapping text regarding n-grams and word sequences between the gold summary and the model outputs.

FactCC: Kryściński et al. (2019) train a BERT (Devlin et al., 2018) model named FactCC to score the factual consistency of the summaries. The model is trained on data generated by transforming ground truth with paraphrasing, swapping entities, numbers, pronouns, etc.

SummaC: Laban et al. (2022) divide the document and the summary into multiple blocks and use an NLI model directly to score, avoiding the granularity mismatch problem. According to the different ways of calculating the final score, the scoring models are divided into two types: SummaCZS and SummaCConv. We use SummaCConv because it has better performance.

DAE: Goyal and Durrett (2021) use dependency arc entailment to realize the evaluation of factual consistency. We use its publicly available code and checkpoint files directly.

QUALS: Durmus et al. (2020) propose a method named FEQA, which checks the factual consistency using question-generation and question-answering models. QUALS is proposed by Nan et al. (2021), which runs substantially faster compared to FEQA, and the evaluation results are proved to have a strong correlation with FEQA. [7]

ClozE: Li et al. (2022) use the cloze model as the base model, which inherits strong interpretability and reduces time consumption.

[6] https://huggingface.co/docs/transformers

[7] We also use FEQA instead of QUALS for error correction and factual evaluation in subsequent experiments. Our approach achieves similar SOTA performance.

post-editing Method		Factual Evaluation Metrics					Rouge			Linguistic Acceptabilty
		ClozE	FactCC	DAE	QUALS	SummaC	R-1	R-2	R-L	
Original Summary		0.4282	0.2122	0.3878	-1.7965	0.2318	**0.2939**	**0.1060**	**0.2512**	0.9260
Baseline	Random	0.4151	0.2425	0.3562	-1.8650	0.2323	0.2707	0.0960	0.2340	0.8773
	BART	0.4558	0.1636	0.3889	-1.6783	0.2507	0.2647	0.0903	0.2237	0.8675
	SpanFact	0.4325	0.2152	0.3859	-1.7899	0.2330	0.2929	0.1038	0.2487	0.9143
	Compedit	0.4718	0.2541*	0.4151	-1.6791	0.2478*	0.2650	0.0902	0.2218	0.9649
	Factedit	0.4282	0.2122	0.3878	-1.7966	0.2318	0.2658	0.0967	0.2315	0.9260
	Cogcomp	0.4265	0.1967	0.3659	-1.7568	0.2332	0.2869	0.1014	0.2454	0.8967
SSP		0.4838*	0.2726	0.4535*	-1.6731*	0.2373	0.2769	0.0994	0.2395	0.9464
SSP works with Baselines	BART+	0.4989	0.2259	0.4416	-1.5482	0.2517	0.2497	0.0840	0.2137	0.8948
	SpanFact+	0.4831	0.2706	0.4486	-1.6717	0.2383	0.2758	0.0977	0.2374	0.9367
	Compedit+	**0.5173**	**0.3232**	**0.4802**	**-1.5579**	**0.2579**	0.2485	0.0840	0.2103	**0.9727**
	Factedit+	0.4840	0.2736	0.4536	-1.6726	0.2327	0.2543	0.0921	0.2231	0.8782
	Cogcomp+	0.4850	0.2629	0.4296	-1.6321	0.2398	0.2694	0.0942	0.2333	0.9250

Table 3: The performance of the SSP method in the FRANK dataset sampled from XSUM. Values marked with * are the highest scores for the factual evaluation metrics achieved by a single post-editing method. The bolded numbers are the highest scores achieved when considering the combination of multiple post-editing methods. We can see that the SSP method achieves SOTA performance and, in combination with existing methods, can further improve fact consistency.

CoLA: CoLA (Warstadt et al., 2019) is a model trained to judge grammatical accuracy to test linguistic competence. It is trained on 10,657 English sentences labeled as grammatical or ungrammatical from published linguistics literature and has available code and checkpoint files.

4.5. Performance

Our main results are shown in Table 3, which measures the performance of various post-editing methods and our method compensating for baseline post-editing methods. For corrected summaries evaluation, we evaluate three aspects: the summary's basic quality, its factual consistency, and linguistic acceptability.

In the baseline method, Compedit is a newly proposed post-editing method and has the ability to remove incorrect entities. As a result, they obtain a high score in some factual evaluation metrics like FactCC and SummaC. In comparison, our method shows the best performance on metrics like ClozE, DAE, and QUALS, reaching SOTA performance. When we combine the existing post-editing method with the SSP method, the factual accuracy of the generated summaries is further steadily improved. The performance improvement is significant, probably because the SSP method complements the shortcomings of the previous post-editing method in terms of error correction range.

There is a small decrease on the Rouge scores after the post-editing correction process. This may be because the training target of the original summarization models indirectly maximizes the Rouge scores. Modifications to the output can hardly further improve them. In addition, pruning some erroneous content from the summary can sometimes result decreased Rouge score. We will analyze the pruned content in subsequent experiments. As for linguistic acceptability, we can see from the table that the baseline method, in addition to Compedit, may lead to a decrease in linguistic acceptability. Because our sentence structure pruning operation is based on a syntactic dependency tree, it does not cause a decrease in linguistic acceptability. Sometimes it has even resulted in an improvement in linguistic acceptability by pruning some grammatically incorrect sentence structures.

Method	ClozE	FactCC	DAE	QUALS	SummaC
Original Summary	0.8565	0.7340	0.9349	-0.7082	0.6850
SSP	0.8739*	0.7637	0.9546*	-0.6209	0.7185
BART	0.8656	0.7144	0.9359	-0.5800	0.6855
+SSP	**0.8754**	0.7476	0.9441	-0.5401	0.7107
SpanFact	0.8363	0.6610	0.9155	-0.7671	0.6395
+SSP	0.8575	0.7080	0.9308	-0.6753	0.6855
Compedit	0.8273	0.5655	0.8785	-0.4796*	0.7340*
+SSP	0.8350	0.5780	0.8877	**-0.4668**	**0.7529**
Factedit	0.8566	0.7341*	0.9350	-0.7083	0.6850
+SSP	0.8733	**0.7642**	**0.9449**	-0.6193	0.7188
Cogcomp	0.8472	0.6628	0.9208	-0.6768	0.6503
+SSP	0.8670	0.7108	0.9349	-0.5896	0.6973

Table 4: The performance of the SSP method in the FRANK dataset sampled from CNN/DM. The meaning of the markers is the same as in Table 3. From the table, we can see that the SSP method's enhancement effect on the factual consistency is also significant in the CNN/DM dataset.

To demonstrate the wide applicability of our research, we do similar experiments on the CNN/DM part of the FRANK dataset. The summaries in CNN/DM dataset differ in language style and length from those on the XSUM dataset, but our SSP method demonstrates the same excellent performance in factual enhancement. In Table 4, by putting together the performance related to the same baseline, we demonstrate more clearly the enhancing effect of the SSP method when it cooperates with baseline methods.

4.6. Analysis of pruned contents

What did SSP prune?

The distribution of dependency relations pruned SSC can be seen in Figure 3. The type of pruned SSUs is consistent with commonly recognized manifestations of errors. When performing human labeling, we found that many error summaries had the wrong place or time in them. These words often need to be joined by pronouns to form a complete SSU. prep (prepositional modifier) also makes up a large percentage of the pruned SSUs. The remaining dependency relations, which make up the larger proportion, are also in line with our perceptions.

This distribution is influenced by the factual evaluation metrics and is not determined solely by the frequency of appearance. For example, prep accounts for 43% of all pruned SSUs, but only 28% of all appeared ones.

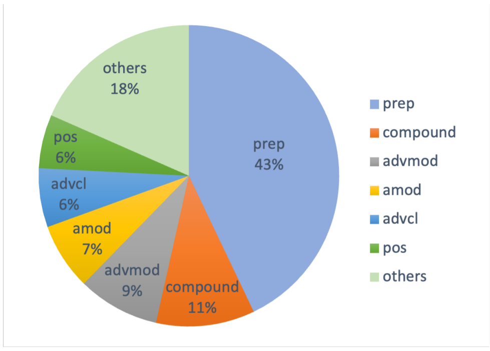


Figure 3: The relations related to the SSUs pruned by SSC in the FRANK dataset. The definition of these relations can be seen at Description.

How much did SSP prune?

It is imperative to acknowledge the inevitable loss of content during the pruning process. Our model's primary objective is to uphold factual accuracy, rather than preserving the richness of content. It is relatively straightforward to extract content from the document, and content can be added to the summary in subsequent steps.

As depicted in Figure 4, compared to CompEdit, our approach has the ability to correct errors that need a greater degree of pruning, but its pruning operation is prudent and appropriate. **Firstly**, as illustrated in Figure 3, pruned SSUs relatively short. prep consists of an average of 5 words, 1.38 words for compound (compound noun modifier), and 1.09 words for advmod (adverbial modifier). SSC more frequently favors the pruning of modifiers over clauses. **Secondly**, the distribution of word numbers pruned by the SSC is similar to the distribution of pruned word numbers when human beings make the necessary corrections.

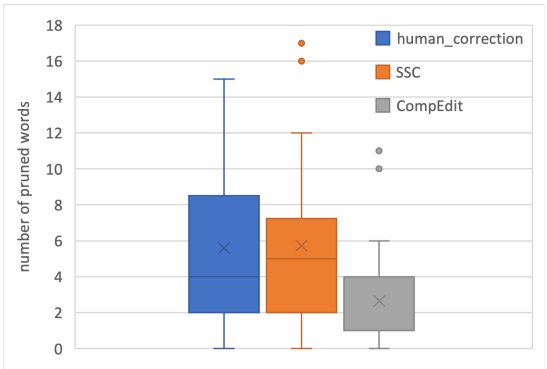


Figure 4: The number of words in the pruned SSUs in the 100 huaman annotation summaries. We compared human modification, SSC and CompEdit.

4.7. Human annotaion

Method	**FI(%)**	**CC(%)**
Manually correction $R^1/P^2/R+P^3$	47/73/82	21/48/60
Original/+SSP^2	0/19	0/3
BART^1/+SSP^3	7/13	1/3
SpanFact^1/+SSP^3	6/23	0/5
Factedit^1/+SSP^3	8/20	1/3
Compedit^2/+SSP^2	13/**28**	1/**6**
Cogcomp^1/+SSP^3	1/18	0/5

Table 5: Results of manual annotation of error correction results. The definition of FI and CC can be found in Section 4.2. In the case of Manual Correction, R and P stands for using **R**eplacement operation or **P**runing operation to correct errors. "Original" in the table represents the original uncorrected summaries. "+SSP" in the table represents the results obtained by applying the SSP method on top of the original method. The superscripted numbers express the type of error correction operation used in these methods. We have highlighted the best data values in bold in the table.

We perform human annotation to evaluate the correction performance of our method and baselines. We sample 100 cases from the XSUM part of the FRANK dataset. We first manually correct these summaries under a limited method (only using replacement operation, only using pruning operation, using replacement and pruning operation). Then, we evaluate the performance of all the post-editing methods. We use open-source annotation tools doccano[8] for text data annotation. In the evaluation process, for the very same article and summary pair, we will show the output of each method in turn. The annotator will determine whether the error has been entirely, partially, or not corrected.

[8] https://github.com/doccano/doccano

Model-level Method		Factual Evaluation Metrics					Rouge			LA
		ClozE	FactCC	DAE	QUALS	SummaC	R-1	R-2	R-L	
Reranking	BART	0.7230	0.2265	0.6616	-1.0643	0.2447	**0.4106**	**0.2025**	**0.3544**	**0.9955**
	Reranking	0.7253	0.3480	0.6639	-1.0679	0.2472	0.4096	0.2017	0.3537	**0.9955**
	SSP(BART)	0.7397	0.2820	0.6984	**-0.9988**	**0.2661**	0.3877	0.1861	0.3364	**0.9955**
	SSP(Reranking)	**0.7534**	**0.3865**	**0.6989**	-1.0055	0.2651	0.3879	0.1862	0.3369	**0.9955**
DAEs	baseline	0.5513	0.1996	0.5908	-1.6252	0.2320	**0.2709**	**0.0886**	**0.2275**	0.9964
	DAEs	0.5596	0.2171	0.5831	-1.2897	0.2326	0.2654	0.0821	0.2218	0.9845
	SSP (baseline)	0.5916	0.2586	**0.6390**	-1.4972	**0.2455**	0.2631	0.0845	0.2214	**0.9969**
	SSP(DAEs)	**0.6059**	**0.2777**	0.6370	**-1.3721**	0.2444	0.2601	0.0799	0.2172	0.9865
Factpegasus	pegasus	0.6969	0.2305	0.6375	-1.1539	0.2497	**0.4306**	**0.2256**	**0.3756**	0.9940
	Factpegasus	0.8212	0.5885	0.8359	-0.7507	0.4054	0.1432	0.0187	0.1151	0.9865
	SSP(pegasus)	0.7171	0.2765	0.6735	-1.0925	0.2685	0.4100	0.2107	0.3600	**0.9955**
	SSP(Factpegasus)	**0.8362**	**0.6185**	**0.8581**	**-0.7466**	**0.4301**	0.1393	0.0179	0.1125	0.9875

Table 6: Comparison and joint performance of SSP method and other model-level methods on XSUM dataset. Table headings and marking rules are similar to Table 3. We select three series of factuality methods, and in each series, we have conducted experiments on four forms: baseline, model-level method, and SSP works with both methods, respectively. The bolded numbers are the highest scores for the factual evaluation metrics achieved in a range of correlated methods.

We invite four proficient English experts to participate in the annotation. To ensure the accuracy of the annotation, at least two experts annotate each output, and we will discuss together when the two annotated results differ.

The outcomes of manual annotation are shown in Table 5. The results demonstrate that the Pruning operation can greatly increase correctable errors compared to the Replacement operation and can cooperate with the Replacement operation to achieve even better error correction performance. The result can also be considered as an upper bound for these post-editing methods.

For all the post-editing methods, we can see that it is not easy to correct the incorrect summaries. Even the best post-editing models only make improvements on a small percentage of incorrect summaries, which is because most errors are caused by multiple mismatched contents in a summary. One or two changes alone cannot make it factually consistent. Some summaries even have nothing to do with the original text but only overlap in some keywords, and it is impossible to correct them.

Consistency with the evaluation results of the factual metrics. CompEdit achieves the best performance among all the baseline methods. However, it still uses only entities as the correction target. Our method takes SSUs as the target of error correction. From the results of manual annotation, the SSP method achieves SOTA performance and has a solid complementary relationship with the baseline methods. The best performance is achieved when SSP and CompEdit work together. A case study can be found in Appendix B.

Fortunately, the SSP approach has the potential to easily further developing. Any factual evaluation metric can be used as a guide for its error correction actions. A superior factual evaluation metric is expected to make the SSP method better.

4.8. Further comparison and cooperation with model-level methods

We also additionally compare our method (a post-editing method) with model-level methods and test their joint effects on improving factual accuracy.

We select Reranking (Chaudhury et al., 2022), DAEs (Goyal and Durrett, 2021) and FactPegasus (Wan and Bansal, 2022) as our model-level methods. We reproduce these methods using available codes. For the reranking strategy, we use BART as a summarization model and FactCC as an evaluation metric for factual evaluation.

The performance of these methods is presented in Table 6. Unlike the previous experiments, this experiment is conducted directly on the XSUM dataset since the summarization models use articles as input. The different model-level factuality methods are also compared with different baseline models selected according to their implementations.

The factual consistency of the output summaries varies considerably between the three series of methods due to the different base models used to make the improvements. Factpegasus performs best in terms of factuality because it optimizes both the pre-training and fine-tuning processes. The improvement of factual consistency by the SSP is significant and stable. As can be seen from the table, both the summaries generated by baseline and those generated by the model-level method can be more factual after correction by SSP.

Since our method does not significantly complement the model-level method in terms of the correction ability, the best performance is not necessarily achieved by their joint efforts. On some of the factual metrics, the highest scores are obtained by using the SSP method to correct the summaries generated by baseline methods rather than model-level methods.

5. Conclusion

In this paper, we introduce an innovative post-editing method based on sentence structure pruning named SSP. This method employs a novel mean and unique targets for error correction, for which the necessity is verified in detail, raising the ability to correct errors to the sentence structure level. Our approach involves a traversal of all potential prunable sentence structures within the syntactic dependency tree, followed by a decision-making process facilitated by a voting strategy integrating multiple evaluation metrics. The correction process highlights the potential that the performance of our method can stay up-to-date as the error correction metrics are developed. Through experiments, the performance of SSP method is comprehensively analyzed, demonstrating its state-of-the-art performance. Further experiments underscore its collaborative potential with existing post-editing methods as well as model-level methods to achieve even better performance.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No.62176024) and Engineering Research Center of Information Networks, Ministry of Education.

7. Bibliographical References

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. *arXiv preprint arXiv:2210.12378*.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. *arXiv preprint arXiv:2109.09784*.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. *arXiv preprint arXiv:2010.08712*.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Subhajit Chaudhury, Sarathkrishna Swaminathan, Chulaka Gunasekara, Maxwell Crouse, Srinivas Ravishankar, Daiki Kimura, Keerthiram Murugesan, Ramón Fernandez Astudillo, Tahira Naseem, Pavan Kapanipathi, et al. 2022. X-factor: A cross-metric evaluation of factual correctness in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7100–7110.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. *arXiv preprint arXiv:2010.02443*.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *arXiv preprint arXiv:2005.03754*.

Alexander R Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022. Improving factual consistency in summarization with compression-based post-editing. *arXiv preprint arXiv:2211.06196*.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32.

Michel Gagnon and Lyne Da Sylva. 2005. Text summarization by sentence extraction and syntactic pruning. *Proceedings of Computational Linguistics in the North East*.

Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 166–175.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yiyang Li, Lei Li, Qing Yang, Marina Litvak, Natalia Vanetik, Dingxin Hu, Yuze Li, Yanquan Zhou, Dongliang Xu, and Xuanyu Zhang. 2022. Just cloze! a fast and simple method for evaluating the factual consistency in abstractive summarization. *arXiv preprint arXiv:2210.02804*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Feng Nan, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Kathleen McKeown, Ramesh Nallapati, Dejiao Zhang, Zhiguo Wang, Andrew O. Arnold, and Bing Xiang. 2021. Improving factual consistency of abstractive summarization via question answering. 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 6881–6894, Online. Association for Computational Linguistics.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Prasad Perera and Leila Kosseim. 2013. Evaluating syntactic sentence compression for text summarisation. In *International Conference on Application of Natural Language to Information Systems*, pages 126–139. Springer.

David Wan and Mohit Bansal. 2022. Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2020. Enhancing factual consistency of abstractive summarization. *arXiv preprint arXiv:2003.08612*.

Category	Relations	Explanation	Valid samples	Acceptable	Unacceptable	Effect
Original summary			1000	821	179	0
A	acl	clausal modifier of noun	91	821	179	0
	dep	unspecified dependency	714	820	180	0.001
	appos	appositional modifier	32	820	180	0.031
	advcl	adverbial clause modifier	108	817	183	0.037
	agent	agent	26	820	180	0.038
	nmod	nominal modifier	24	820	180	0.042
	advmod	adverbial modifier	221	811	189	0.045
	prt	phrasal verb particle	43	819	181	0.047
	prep	prepositional modifier	787	782	218	0.05
B	amod	adjectival modifier	460	797	203	0.052
	compound	compound noun modifier	382	801	199	0.052
	neg	negation modifier	54	816	184	0.093
	relcl	relative clause modifier	86	812	188	0.105
	punct	punctuation mark	503	754	246	0.133
	ccomp	clausal complement	285	780	220	0.144
	poss	possession modifier	170	792	208	0.171
	conj	conjunct	185	787	213	0.184
	pcomp	prepositional complement	65	809	191	0.185
	xcomp	open clausal complement	120	798	202	0.192
C	case	case marking	102	799	201	0.216
	npadvmod	noun phrase as adverbial modifier	27	815	185	0.222
	attr	attribute	83	802	198	0.229
	det	determiner	544	694	306	0.233
	oprd	object predicate	20	816	184	0.25
	nummod	numeric modifier	12	818	182	0.25
	cc	coordinating conjunction	192	772	228	0.255
	dobj	direct object	487	693	307	0.263
	aux	auxiliary	405	704	296	0.289
	mark	marker	72	800	200	0.292
	pobj	object of preposition	624	639	361	0.292
	acomp	adjectival complement	87	793	207	0.322
	auxpass	auxiliary (passive)	164	768	232	0.323
	nsubj	nominal subject	672	575	425	0.366
	nsubjpass	passive nominal subject	120	748	252	0.608

Table 7: The classification of syntactic dependency relations and statistics of changes in linguistic acceptability after pruning. "Valid sample" represents the number of summaries that contains this syntactic dependency relation and where the corresponding sentence structure is pruned. "Effect" is obtained by dividing the number of additional linguistically unacceptable summaries by the number of valid samples.

A. Classification of syntactic dependency relations

The classification of syntactic dependency relations is based on their effect on linguistic acceptability. We sample 1000 golden summaries from the XSUM dataset and try to prune the sentence structures related to the specified syntactic dependencies. We calculate the number of linguistically unacceptable sentences caused by the pruning operation. Since different syntactic dependency relations occur at different frequencies, the number of valid samples for which modifications are made varies. We quantify its effect on linguistic acceptability by dividing the increase in the number of linguistically unacceptable sentences by the number of valid samples.

The classification results and statistical information can be found in Table 7. It is worth mentioning that some grammatical dependencies, such as "dative" and "preconj", are not included in the table because the valid sample data were too small.

B. Case study

We select a representative example to show the difference between the post-editing methods. As we can see, the generated summary has an incorrect modifier "worth more than 100,000" and an incorrect target object "a county durham museum". The value of art is never mentioned in the document, and the correct target object is "the Victoria and Albert Museum". We must remove the incorrect modifier and replace or remove the incorrect target to correct this error. Details of the different output contents are shown in Table 8.

Using the baseline method to correct will encounter many problems. First of all, not all entities can be correctly identified. When we use the most common tool, Spacy, for entity extraction, the museum's name is not successfully extracted. This results in the baseline method, which uses the entity as the error correction object, not checking the museum name. Second, except for Compedit, the baseline methods all use replacement as a means of error correction, making them powerless against incorrect modifiers. Compedit successfully removes the incorrect modifier but causes a syntax error during the removal process.

The SSP method can correct these errors. For example, when using the SSP method alone, the incorrect modifier is successfully deleted. When combined with the baseline method, the SSP method can further remove errors on top of the original output. However, the SSP method itself has shortcomings. It can only handle one error at a time. It is also limited by the fact that the existing factual evaluation metrics are sensitive, and its error correction action may change significantly when there are only minor changes in a summary sentence. Fortunately, the SSP approach has the potential to easily further developing. Any factual evaluation metric can be used as a guide for its error correction actions. We expect more outstanding evaluation metrics to make the SSP method even better.

– Basic information of summary –

Source article: *About 80,000 works of art, ceramics, manuscripts, letters, and photographs faced being auctioned to help pay off the pottery firm's pension debt. But a public fundraising campaign launched in September hit its target in just a month. Administrator Begbies Traynor said the collection will remain on display at the museum in Barlaston, Staffordshire. The Art Fund, which led the campaign, will gift the collection to the Victoria and Albert Museum before it is loaned to the Wedgwood Museum. Administrator Bob Young said it had been "incredibly satisfying" to sign off on the sale on Monday. "Today's fantastic outcome wouldn't have been possible without the spirit of goodwill and determination shown during the often complex negotiations," he said. The Wedgwood Museum inherited Waterford Wedgwood plc's pension bill after the firm collapsed in 2009. In 2010 the museum also went into administration, and in 2011 a high court judge ruled its collection could be sold to reimburse the Pension Protection Fund. Alison Wedgwood, whose husband Tom is a direct descendant of company founder Josiah Wedgwood, said the collection was "important" for Staffordshire. The collection risked being "sold and scattered around the globe" had the money not been raised, she added. Tristram Hunt MP, who was involved in the campaign, said the sale was "fantastic news". "The items contained within the Wedgwood collection chart a significant part of Britain's cultural development over centuries and play a crucial part in defining our national identity today," he said."*

Summary: *A collection of art worth more than 100,000 has been donated to a county durham museum*.

Recognized entities: *More than 100,000*

Manual corrected summary: *A collection of art has been donated to the Victoria and Albert museum*.

– Summaries corrected by baselines –

BART (2 errors): *A collection of art worth more than 100,000 has been donated to a county durham museum*.

SpanFact (2 errors): *A collection of art worth more than 100,000 has been donated to a county durham museum*.

Compedit (1 error): *A collection of art donated to a county durham museum*.

Factedit (2 errors): *A collection of art worth more than 100,000 has been donated to a county*.

Cogcomp (2 errors): *A collection of art worth more than 100,000 has been donated to a county durham museum*.

– Summaries further corrected by SSP –

SSP (1 error): *A collection of art has been donated to a county durham museum*.

BART+ (2 errors): *A collection of art worth more than 100,000 has been donated to a county durham museum*.

SpanFact+ (1 error) : *A collection of art has been donated to a county durham museum*.

Compedit+ (1 error): *A collection of art donated to a durham museum*.

Factedit+ (1 error): *A collection of art worth more than 100,000 has been donated*.

Cogcomp+ (1 error): *A collection of art has been donated to a county durham museum*.

Table 8: Examples of a summary with factual errors and the correction results for each post-editing method.