Improving Sequence-to-Sequence Pre-training via Sequence Span Rewriting

Wangchunshu Zhou[1*] **Tao Ge**[2] **Ke Xu**[1] **Furu Wei**[2]
[1]Beihang University, Beijing, China
[2]Microsoft Research Asia, Beijing, China
zhouwangchunshu@buaa.edu.cn, kexu@nlsde.buaa.edu.cn
{tage, fuwei}@microsoft.com

Abstract

In this paper, we generalize text infilling (e.g., masked language models) by proposing Sequence Span Rewriting (SSR) as a self-supervised sequence-to-sequence (seq2seq) pre-training objective. SSR provides more fine-grained learning signals for text representations by supervising the model to rewrite imperfect spans to ground truth, and it is more consistent than text infilling with many downstream seq2seq tasks that rewrite a source sentences into a target sentence. Our experiments with T5 models on various seq2seq tasks show that SSR can substantially improve seq2seq pre-training. Moreover, we observe SSR is especially helpful to improve pre-training a small-size seq2seq model with a powerful imperfect span generator, which indicates a new perspective of transferring knowledge from a large model to a smaller model for seq2seq pre-training.

1 Introduction

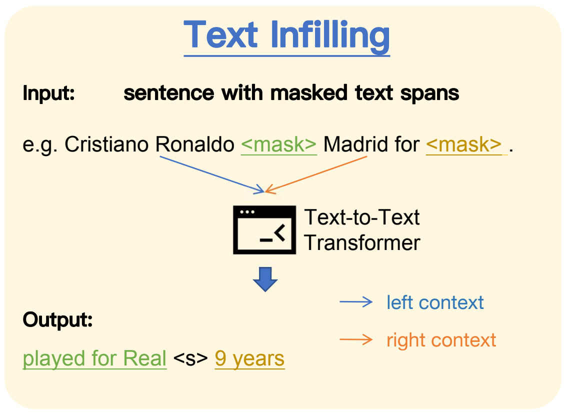
Text infilling (e.g., masked language models) has become a prevalent learning objective for pre-training models in Natural Language Processing (NLP) (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019; Lan et al., 2019; Lewis et al., 2019; Raffel et al., 2019). It provides self-supervision for the model by infilling masked parts for reconstruction of plain text, as Figure 1 shows, and enables the model to effectively learn text representation with the prediction of the masked content based on contexts.

In this paper, we generalize text infilling by text rewriting. Specifically, we propose Sequence Span Rewriting (SSR) as sequence-to-sequence (seq2seq) pre-training objective. As Figure 1 shows, SSR offers self-supervision of rewriting imperfect text spans to ground truth. Compared with text infilling which is a special case of SSR where the imperfect spans are empty text (i.e., masked), SSR has two advantages: **1)** it provides more diverse and fine-grained learning signals regarding how to improve text spans by rewriting since the model's prediction is not only based on contexts but also conditioned on the imperfect spans; **2)** it is more consistent with downstream seq2seq tasks (e.g., text summarization) where a source sentence is mapped to the target sentence following some rewriting patterns.

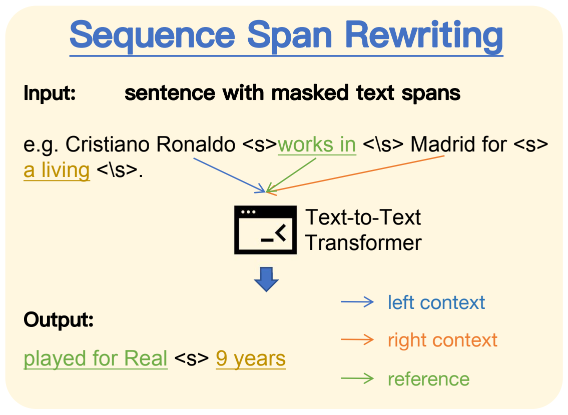
In contrast to text infilling where a text span in plain text is always replaced with empty text (i.e., masked), SSR replaces the text spans with imperfect text, as mentioned above. While there are several straightforward ways (e.g., random or rule-based noising and corruption) to generate imperfect spans, most of them cannot generate diverse and informative text for the spans. As a result, the model could only learn limited and valueless rewriting patterns, making SSR degrade into text infilling. To fully exploit SSR, we propose to use a pre-trained text infilling model as the imperfect span generator, inspired by ELECTRA (Clark et al., 2020), for generating imperfect spans to guarantee both their quality and diversity, as shown in Figure 1. In this way, SSR enables the model to not only learn to reconstruct a masked sequence but also learn meaningful and diverse rewriting patterns including paraphrase, correction of grammatical, commonsense and even factual errors, to improve a text sequence.

In our experiments, we apply SSR to the T5 models (Raffel et al., 2019) that are pre-trained with the text infilling objective and use the T5-large model as our imperfect span generator. We show SSR improves both the original T5 models and their continual training variations on monolingual seq2seq tasks including text summarization, question generation, and grammatical error correction. Moreover, we observe SSR is especially helpful to improve

*This work was done during the first author's internship at Microsoft Research Asia.


Text Infilling
Input: sentence with masked text spans
e.g. Cristiano Ronaldo <mask> Madrid for <mask> .
Text-to-Text Transformer
Output:
played for Real <s> 9 years
left context
right context

(a) *Text infilling* pre-trains a sequence-to-sequence language model to infill masked text spans.


Sequence Span Rewriting
Input: sentence with masked text spans
e.g. Cristiano Ronaldo <s>works in <\s> Madrid for <s> a living <\s>.
Text-to-Text Transformer
Output:
played for Real <s> 9 years
left context
right context
reference

(b) *Sequence span rewriting* pre-traineds a sequence-to-sequence language model to rewrite imperfect text spans into groundtruth.

Figure 1: Illustration of text infilling and sequence span rewriting objectives.

pre-training smaller seq2seq models with a powerful imperfect span generator, indicating a potential perspective of transferring knowledge from a large model to a smaller model for seq2seq pre-training.

2 Related Work

NLP pre-training Early pre-training methods like ELMo (Peters et al., 2018), GPT (Radford et al., 2018) were based on language modeling. More recently, Radford et al. (2019); Brown et al. (2020) show that very large language models can act as unsupervised multi-task/few-shot models.

BERT (Devlin et al., 2018) introduced the masked language modeling objective by masking out certain tokens in a text and predict them based on their left and right side contexts. Recent work has shown that BERT's performance can be further improved by training for longer (Liu et al., 2019), by tying parameters across layers (Lan et al., 2019), and by discrimination intead of generation (Clark et al., 2020). However, in BERT-like masked language models, predictions are not made auto-regressively, reducing their effectiveness for NLG tasks. To enable mask language models for natural language generation tasks, Song et al. (2019) used a decoder to generate the masked tokens autoregressively instead of directly connecting an MLP after the encoder.

UniLM (Dong et al., 2019) fine-tunes BERT with an ensemble of masks, some of which allow only leftward context. This allows UniLM to be used for both generative and discriminative tasks.

More recently, BART (Lewis et al., 2019) and T5 (Raffel et al., 2019) propose to use text infilling as a self-supervised objective to pre-train text-to-text transformers that are boardly applicable to a wide range of both NLU and NLG tasks. Specifically, they remove text spans in the input texts and train the models to recover the original texts in an auto-regressive fashion. CALM (Zhou et al., 2020b) proposed to pre-train text-to-text transformers by training them to compose sentences based on keywords or concepts. All aforementioned methods can be viewed as denoising auto-encoders that are trained to receive noisy inputs as source sequences and recover the original inputs. However, the noisy sentence they receives during pre-training are incomplete sentences, therefore suffer from the aforementioned gap between pre-training and fine-tuning.

Pre-trained model compression/acceleration Recently, many attempts have been made to compress and speed up a large pre-trained language model. For example, Shen et al. (2020) quantized BERT to 2-bit using Hessian information. Michel et al. (2019) pruned unnecessary attention heads in the transformer layers to reduce the parameters of a BERT model. DistilBERT (Sanh et al., 2019) uses knowledge distillation (Hinton et al., 2015; Romero et al., 2014) to compress BERT. More recently, Xu et al. (2020) introduced progressive module replacing to train more compact BERT models. In addition, Zhou et al. (2020c); Schwartz et al. (2020) proposed to accelerate the inference stage of pre-trained models via input-adaptive inference. However, up to now, few work has been done for compressing large pre-trained text-to-text

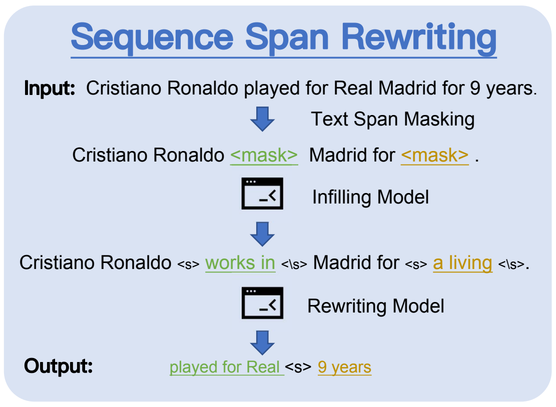

Sequence Span Rewriting
Input: Cristiano Ronaldo played for Real Madrid for 9 years.
Text Span Masking
Cristiano Ronaldo <mask> Madrid for <mask> .
Infilling Model
Cristiano Ronaldo <s> works in <\s> Madrid for <s> a living <\s>.
Rewriting Model
Output: played for Real <s> 9 years

Figure 2: Workflow ot the proposed text span rewriting objective. We first randomly mask a number of text spans and then use a pre-trained sequence-to-sequence language model to infill the masked spans. We then train the rewrite model to rewrite the machine-generated text spans into groundtruth text spans.

transformers for downstream NLG tasks.

3 Sequence Span Rewriting

The key idea of the sequence span rewriting objective is to train a text-to-text transformer to rewrite machine-generated texts that may contain diverse kinds of noises such as paraphrase, lexical substitution, grammatical errors, wrong world knowledge/commonsense knowledge, text expansion and simplification, into human-written texts that are correct, appropriate, and make sense. Specifically, self-supervised training with the sequence span rewriting objective involves three steps: (1) text span masking (2) text infilling, and (3) sequence span rewriting. We will describe these steps in detail and discuss the advantages of the sequence span rewriting objective in terms of general pre-training and distilling large text-to-text transformers.

Text span masking To generate training data of sequence span rewriting in a self-supervised fashion, we first randomly sample a number of text spans and mask them. Specifically, each span is replaced with a single `[MASK]` token with span lengths drawn from a Poisson distribution ($\lambda = 3$). The number of spans is controlled so that approximately 30% of all tokens are masked. 0-length spans correspond to the insertion of `[MASK]` tokens. These choices are made to ensure the corrupted input texts for the text infilling step matches pre-training data distribution of BART (Lewis et al., 2019) and T5 (Raffel et al., 2019) that are used for data generation.

Text infilling The second step is using a text-to-text transformer pre-trained with the text infilling objective to fill the masked text spans. Concretely, we feed the corrupted sentences into the text infilling model to generate the predicted text spans in an auto-regressive fashion. To improve the diversity of rewrite patterns, we use Nucleus Sampling (Holtzman et al., 2020) that truncates the unreliable tail of the probability distribution and samples from the dynamic nucleus of tokens containing the vast majority of the probability mass.

sequence span rewriting With the data generated in the previous steps, we then train our model to rewrite the machine-generated text spans into original texts. Specifically, we use special tokens $<\mathrm{s}_i>$ and $</\mathrm{s}_i>$ to denote the starting and ending of i-th text span to be rewritten in the source sequences, and use $<\mathrm{s}_i>$ to separate different original text spans in the target sequences. We train the model to generate target text spans from left to right auto-regressively by maximum likelihood estimation. The entire pre-training procedure is presented in Figure 2.

Pre-training via rewriting We discuss several key advantages of the proposed sequence span rewriting objective over the conventional text infilling objective for pre-training text-to-text transformers here.

- First, the task of sequence span rewriting is closer to the downstream sequence transduction tasks since there exists references in source sequences for generation of target text spans, reducing the gap between pre-training and fine-tuning. Indeed, many important NLG tasks such as machine translation, text summarization, grammatical error correction, and text style transfer, can be viewed as text rewriting problems that rewrite the input text into another language, compact text, correct sentence, and another style, respectively.

- Second, sequence span rewriting introduces more diverse noise patterns, including paraphrasing and simplification of the text span, missing or redundant information, grammatical errors, and errors in terms of world knowledge or commonsense knowledge. In contrast, conventional self-supervised pre-training of text-to-text transformers is generally based on rule-based noise functions like text span masking, token masking/deletion, sentence shuffling, etc.

- Finally, sequence span rewriting objective enables the model to learn from mistakes made by the infilling model used for data generation (or the model itself if the rewriting model is initialized by the infilling model), thus providing more informative self-supervision.

Distilling via rewriting Self-supervised sequence span rewriting also provides a new perspective of exploiting the knowledge of a large text-to-text pre-trained model to improve smaller models. This can be achieved by using a large-size teacher model pre-trained with the text infilling objective to generate infilling data and pre-train a small-size student model with the generated data and the sequence span rewriting objective. Different from conventional knowledge distillation (Hinton et al., 2015) or sequence-level knowledge distillation (Kim and Rush, 2016), sequence span rewriting enables the student model to exploit both teacher outputs and ground-truth at the same time. It is also related to boost learning (Schapire, 2003) and residual learning (He et al., 2016) in a sense that the model only need to learn how the teacher model fails to predict in the first place. In addition, the sequence span rewriting objective reduces the multi-modality issue described in Zhou et al. (2020a) by providing source-side references via outputs generated by a high-capacity teacher model. Providing source-side reference also reduces the capacity requirement to succeed the pre-training task, since the model can refer to the machine-generated text spans during generation, instead of doing unconditional generation like in the case of text infilling.

4 Experiments

4.1 SSR

SSR is implemented as a text-to-text transformer model with a bidirectional encoder over texts with text spans rewritten by machine, and a left-to-right auto-regressive decoder. For pre-training, we optimize the negative log likelihood of the original human-written text spans. We describe details of architecture, pre-training, and fine-tuning of SSR in this section.

Architecture We use the same text-to-text transformer architecture compared with T5 (Raffel et al., 2019). Specifically, our model is roughly equivalent to the original Transformer proposed by Vaswani et al. (2017), with the exception of removing the Layer Norm bias, placing the layer normalization outside the residual path, and using a different position embedding scheme, where each "embedding" is simply a scalar shared across all layers that is added to the corresponding logit used for computing the attention weights.

Motivated by recent success of T5 (Raffel et al., 2019), we train three versions of SSR: SSR-small, SSR-base, and SSR-large. Specifically, SSR-small model has around 60 million parameters, consists of 6 transformer layers, each of which has 8 attention heads and a dimensionality of 512. SSR-base has around 220 million parameters with 12 transformer layers, each of which has 12 attention heads and a dimensionality of 768. SSR-large is the largest model we pre-trained, it uses a dimensionality of 1024, a 16-headed attention, and 24 transformer layers each in the encoder and decoder, resulting in around 770 million parameters.

Pre-training SSR All these models are initialized with the corresponding pre-trained T5 model with same architecture, and continually pre-trained with the text rewriting objective. Since the text rewriting objective is formulized as a sequence-to-sequence task, we continually pre-train SSR using seq2seq in huggingface's transformers[1] (Wolf et al., 2020). We use T5-large for infilling data generation. Therefore, SSR-large can be regarded as continual pre-training while SSR-base and SSR-small can be considered as distilling or exploiting the knowledge of a larger pre-trained model for training smaller text-to-text transformers in a task-agnostic setup. More precisely, we use nucleus sampling with $p = 0.9$ to sample machine-generated text spans. For pre-training data, we select portions from Wikipedia corpus, BookCorpus (Zhu et al., 2015), and RealNews (Zellers et al., 2019), which are commonly used for pre-training language models. The resulting pre-training corpus consists over 4GB of text.

We use text sequences containing approximately 256 tokens, which may contain multiple sentences, to sample masked text spans and generate infilling data. We then continually pre-train different variants of SSR for 100k updates, with a maximum sequence length of 256, a batch size of 512 text sequences, and a learning rate of 5e-5 with linear

[1]https://github.com/huggingface/transformers

Model	Architecture	CNN/DM			XSum		
		RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
Performance of models without pre-training							
Lead-3	-	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	-	36.44	15.66	33.42	29.70	9.21	23.24
Performance of state-of-the-art models based on pre-trained models of comparable sizes							
MASS (Song et al., 2019)	L=6, H=1024	42.12	19.50	39.01	39.75	17.24	31.95
BERTSUMABS (Liu and Lapata, 2019)	L=12, H=768	41.72	19.39	38.76	38.76	16.33	31.15
UniLMv2 (Bao et al., 2020)	L=12, H=768 (shared)	43.16	20.42	40.14	**44.00**	21.11	**36.08**
Performance of comparable models based on T5-base							
T5-base (Raffel et al., 2019)	L=12, H=768	42.25	20.22	39.45	43.12	20.84	34.98
T5-base-cont	L=12, H=768	42.49	20.33	39.65	43.32	20.94	35.21
DistilT5-base	L=12, H=768	42.37	20.25	39.53	43.25	20.89	35.14
SSR-base	L=12, H=768	**43.38**	**20.65**	**40.27**	43.99	**21.13**	35.60

Table 1: Abstractive summarization results on CNN/DailyMail and XSum. We also present the transformer architecture for the methods using pre-trained models. For example, L=12, H=768 means both the encoder and decoder is built with 12 transformer layers with a hidden size of 768. "Shared" means the encoder and decoder parameters are shared

warmup for the first 8000 updates. We empirically find 100k updates to be enough since the models' performance on downstream begin to saturate. We think this is because pre-trained T5 models provide a good initialization and the pre-training corpus is relatively small. We leave larger-scale pre-training from scratch over larger text corpus for future work.

4.2 Tasks and Datasets

We fine-tune SSR on multiple sequence-to-sequence natural language generation tasks that are closely related to the sequence span rewriting objective. We describe the selected tasks, their relation to text rewriting, and corresponding datasets here.

Summarization The task of abstractive summarization requires the model to receive a long article as input and output a short summary of the input. This task can be viewed as rewriting the input article into a more compact form.

To provide a comparison with the recent work in pre-trained models for summarization, we present results on two summarization datasets, CNN/DailyMail (Hermann et al., 2015) and XSum (Narayan et al., 2018), which have distinct properties. Summaries in the CNN/DailyMail tend to resemble source sentences. Extractive models do well here, and even the baseline of the first-three source sentences is highly competitive. In contrast, XSum is highly abstractive and extractive models perform poorly. Therefore, text rewriting ability is needed to succeed the XSum dataset.

Question Generation The goal of question generation is to generate a valid and fluent question according to a given passage and the target answer. It can be considered as rewriting the target answer and its surrounding context into a question form. Following previous work (Dong et al., 2019), the input of the sequence-to-sequence problem is defined as the concatenation of a paragraph and an answer. We fine-tune the pre-trained models to predict output questions. We use SQUAD (Rajpurkar et al., 2016) dataset for training and testing question generation following the data split in Du and Cardie (2018).

Grammatical Error Correction In grammatical error correction (GEC), the model receives a potentially erroneous input sentence and generates an output sentence that is grammatically correct while retaining the original meaning of the input sentence. This task can be considered to rewrite erroneous text spans so that they are grammatically correct. Following the recent work (Grundkiewicz et al., 2019; Kiyono et al., 2019; Zhou et al., 2019) in GEC, the GEC training data we use is the public Lang-8 (Mizumoto et al., 2011), NUCLE (Dahlmeier et al., 2013), FCE (Yannakoudakis et al., 2011) and W&I+LOCNESS datasets (Bryant et al., 2019; Granger, 1998). We do not use any synthetic GEC data before fine-

Model	Architecture	Question Generation			GEC		
		BLEU-4	METEOR	CIDEr	P	R	$F_{0.5}$
Performance of models without pre-training							
Zhang and Bansal (2019)	-	18.37	22.65	46.68	-	-	-
Transformer-big (Chen et al., 2020)	L=12, H=1024	-	-	-	64.9	26.6	50.4
Performance of state-of-the-art models based on pre-trained models of comparable sizes							
UniLMv2 (Bao et al., 2020)	L=12, H=768 (shared)	24.43	26.34	51.97	-	-	-
Performance of comparable models based on T5-base							
T5-base (Raffel et al., 2019)	L=12, H=768	23.74	25.95	51.61	68.6	33.5	56.7
T5-base-cont	L=12, H=768	23.93	26.11	51.78	69.6	33.6	57.3
DistilT5-base	L=12, H=768	23.86	25.93	51.64	69.3	33.1	56.9
SSR-base	L=12, H=768	24.29	**26.46**	**52.05**	**70.3**	**34.8**	**58.4**

Table 2: Question generation and grammatical error correction results. We also present the transformer architecture for the methods using transformer models. For example, L=12, H=768 means both the encoder and decoder is built with 12 transformer layers with a hidden size of 768. "Shared" means the encoder and decoder parameters are shared

tuning on GEC datasets.

4.3 Compared Models

We compare SSR against the following models to test the effectiveness of the sequence span rewriting objective.

- **T5**: the original pre-trained text-to-text transformer based on the text infilling objective.
- **T5-cont**: text-to-text transformers initialized by T5 and continually pre-trained with the original text infilling objective with additional training steps. The total number of additional training steps is equal to that of SSR.
- **DistilT5**: the variant that continually pre-train T5 by text infilling with sequence-level knowledge distillation (Kim and Rush, 2016). This is realized by using the infilling data generated by T5-large as target outputs for text infilling. DistilT5-small and DistilT5-base is similar to conventional sequence-level knowledge distillation while DistilT5-large can be viewed as continually pre-trained with self distillation.

For reference, we also compare against several strong pre-trained models for natural language generation including MASS (Song et al., 2019), UniLM (Dong et al., 2019), and UniLMv2 (Bao et al., 2020).

4.4 Experimental Results

We first present experimental results of SSR-base and comparable baselines on different datasets. Then we present results of SSR-small and SSR-large for analysis.

Summarization Results For summarization datasets, we report ROUGE (Lin, 2004) scores of different compared models in Table 1. We can see that SSR-base substantially improve the original T5-base model, as well as T5-base-cont, a competitive baseline that continually pre-train T5 for the same steps, on both CNN/DM and XSum datasets. Surprisingly, continually pre-train T5-base with sequence-level knowledge distillation does not improve the performance significantly. In fact, it underperforms the baseline that continually pre-train T5-base with the text infilling objective. SSR-base also outperforms several models based on base-size pre-trained models including MASS and BERTSUMABS, and performs competitively with UniLMv2, the state-of-the-art pre-trained model of base-size on text summarization benchmarks.

Question Generation and GEC Results We present the results on question generation and GEC datasets in Table 2. Similarly, we find that SSR-base substantially outperforms all T5-base based baselines. Interestingly, we find the improvement of SSR on GEC task is larger than that on question generation and summarization datasets. We think this is because the task of GEC has a closer relation with the TSR objective.

Impact of Model Size To analyze the effectiveness of the proposed TSR objective for sequence-to-

Model	**CNN/DM**		
	RG-1	RG-2	RG-L
T5-large	42.50	20.68	39.75
T5-large-cont	42.54	20.71	39.77
DistilT5-large	42.45	20.66	39.71
SSR-large	**42.65**	**20.84**	**39.92**

Table 3: Abstractive summarization results on CNN/DailyMail for T5-large based models.

Model	**CNN/DM**		
	RG-1	RG-2	RG-L
T5-small	41.12	19.56	38.35
T5-small-cont	41.33	19.75	38.58
DistilT5-small	41.28	19.69	38.51
SSR-small	**41.95**	**20.06**	**39.01**

Table 4: Abstractive summarization results on CNN/DailyMail for T5-small based models.

sequence pre-trained models with different sizes, we report the performance comparison of small-size and large-size SSR and different T5-based baselines. The results of large-size models and small-size models in Table 3 and Table 4 respectively. We find that the sequence span rewriting objective improves both large-size and small-size models. However, the improvement upon small-size models is significantly larger than that upon large-size models. This suggests that our method is more effective when the infilling model is significantly larger than the rewriting model. The performance of SSR-small is also significantly better than DistilT5-small sequence-level knowledge distillation. This opens a new perspective of distilling the knowledge from large pre-trained sequence-to-sequence transformers into smaller models in a task-agnostic fashion.

5 Conclusion

We present sequence span rewriting (SSR), a novel self-supervised objective for improving sequence-to-sequence transformers pre-trained with conventional text infilling objectives. SSR introduces more diverse and fine-grained learning signals while also bridge the gap between self-supervised pre-training and task-specific fine-tuning on common NLG datasets. Experiments on continually pre-training T5 show the effectiveness of SSR on improving pre-trained T5 models of different sizes. Specifically, the improvements on using imperfect spans generated by T5-large to continually pre-train T5-small show that SSR can effectively exploit the knowledge of a large pre-trained sequence-to-sequence transformer to improve a smaller model. This work is in progress and for future work, we plan to use more data for pre-training, as well as investigate the performance of SSR when generating impefect data using smaller pre-trained models.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. *arXiv preprint arXiv:2010.03260*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 22–31.

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. In *Advances in Neural Information Processing Systems*, pages 13063–13075.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Sylviane Granger. 1998. *The computer learner corpus: a versatile new source of data for SLA research*. na.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. *arXiv preprint arXiv:1909.00502*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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*.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning sns for automated japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155.

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*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Robert E Schapire. 2003. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer.

Roy Schwartz, Gabi Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. 2020. The right tool for the job: Matching model and instance complexities. *arXiv preprint arXiv:2004.07453*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *AAAI*, pages 8815–8821.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

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 *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, 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.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. *In EMNLP 2020*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 180–189.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in neural information processing systems*, pages 9054–9065.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. *arXiv preprint arXiv:1909.06356*.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020a. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2019. Improving grammatical error correction with machine translation pairs. *arXiv preprint arXiv:1911.02825*.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. 2020b. Pre-training text-to-text transformers for concept-centric common sense. *arXiv preprint arXiv:2011.07956*.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020c. Bert loses patience: Fast and robust inference with early exit. *In NeurIPS 2020*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.