Abstract

Few-shot abstractive summarization has become a challenging task in natural language generation. To support it, we developed a novel soft prompts architecture coupled with a prompt pre-training plus prompt fine-tuning paradigm, which is effective and tunes only extremely light parameters. To meet the structure of the generation models, the soft prompts comprise continuous input embeddings across an encoder and a decoder. Importantly, a new inner-prompt placed in the text is introduced to capture document-level information. The aim is to devote attention to understanding the document that better prompts the model to generate document-related content. In the training process, the prompt pre-training with self-supervised pseudo-data firstly teaches the model basic summarizing capability. Then, with few-shot examples, only the designed lightweight soft prompts are fine-tuned. Experimental results on the CNN/DailyMail and XSum datasets show that our method, with only 0.1% of the parameters, outperforms full-model tuning where all model parameters are tuned. It also surpasses Prompt Tuning by a large margin and delivers competitive results against Prefix-Tuning with 3% of the parameters.

1 Introduction

Given the high labor-costs of obtaining quality abstractive summaries, few-shot abstractive summarization is very demanding and highly challenging. A widely accepted paradigm for almost all NLP tasks is to fine-tune the entire set of parameters for a large pre-trained language model to suit the target task (Liu and Lapata, 2019; Liu et al., 2020).

However, the fine-tuning with few-shot examples usually leads to disappointing results, especially with generation tasks like abstractive summarization (Fabbri et al., 2020; Yu et al., 2021). The likely outcome is an overfit model. Further, for every specific task, a large number of pre-trained parameters need to be updated and stored, which is not efficient to use.

Pre-trained language models are few-shot learners, i.e., GPT-3 (Brown et al., 2020) that surprisingly perform generation tasks from a few examples without any further gradient updates. Although it lacks a rigorously theoretical proof, prompt learning inherits the few-shot property (Li and Liang, 2021; Schick and Schütze, 2020; Jin et al., 2021; Liu et al., 2021). Commonly, this type of learning is considered to retrieve relevant knowledge from frozen language models, only tuning continuous prompts to quickly adapt to new tasks with very few examples.

More recently, Prompt Tuning (Lester et al., 2021) has received much attention. With large frozen language models (say, >10 billion parameters), Prompt Tuning simply adds a tunable soft prompt to the input of the encoder, achieving results that are comparable to full-model tuning. Yet, our empirical results, in Section 2, demonstrate that Prompt Tuning for abstractive summarization yields simply abysmal performance. Prefix-Tuning (Li and Liang, 2021) extends the use of prompt learning in the natural language generation area. With this technique, continuous prompts are applied to every layer of the pre-trained model and even shows increase in few-shot generation tasks over fine-tuning. Yet the training process is not stable and updates are required that add to the memory and training costs.\(^1\)

Given the shortcomings of these two methods, we have developed a soft prompts tuning method that is specifically designed for summarization. The structure is given in Figure 1. The method is capable of performing few-shot language generation task (i.e., abstractive summarization) with an efficient amount of training parameters. Prompt tokens are added before the decoder input tokens.

\(^1\)See more related work in Section 5.
to guide the generation process toward the target summary. Moreover, we have designed three inner prompts – interval, sequential, and fixed-length – one of which is placed among the source input tokens. The aim is to capture the structure in the source document and aid in understanding its semantics, so as to better prompt the model to generate document-related content. Each kind of inner prompts focuses on different semantic units (e.g., phrases, sentences, and etc.), differentiating important units from non-informative ones. To bolster the summarization ability of the model and assist the prompts to understand the documents, prompt pre-training is performed before the tuning process, and leveraged by self-supervised pseudo data. As a last step, all the prompts are fine-tuned with few-shot training examples. Experiments conducted on two commonly used datasets – CNN/DM (See et al., 2017) and XSum (Narayan et al., 2018) – demonstrate that our method outperforms full-model tuning under few-shot settings only with 0.1% of the parameters. It also surpasses naive Prompt Tuning by a large margin. Our model also yields a performance competitive to Prefix-Tuning with 3% of the trainable parameters. A detailed analysis shows that the designed prompt-pre-training phase and the inner prompts are effective for few-shot text summarization. Thus, the major contributions of this work include: 1) A novel soft prompt architecture for few-shot abstractive summarization. With the well-designed prompts in embedding layer, our model fulfills the task effectively and efficiently; 2) It is necessary to perform prompt pre-training strategy which benefits soft prompts model for few-shot summarization and shows excellent zero-shot capabilities; 3) Experiments that investigate the effect of different prompts by probing the attention weights. The results show our model is able to: extract knowledge from the encoder language model; understand the discourse in the document; and guide the decoder language model to generate fluent summaries.

2 Pilot Experiments

In a pilot study, we experimented with using Prompt Tuning under 300-shots settings to find reasonable clues as to how to design summary-prompts for the task. Our findings follow.

Consider an encoder-decoder language model $p_\theta(y|x)$ based on the Transformer architecture (Vaswani et al., 2017) (e.g., BART (Lewis et al., 2020)) and parameterized by $\theta$. To conduct a few-shot summarization task, we have some few-shot training pairs of a document $X = \{x_1, x_2, \ldots, x|X|\}$ and a corresponding summary $Y = \{y_1, y_2, \ldots, y|Y|\}$. Specifically, we divided $X$ into different subsets with sentences\(^2\) as our unit, $X = \{x^1_1, x^1_2, \ldots, x^n_m\}$, where $x^i_j$ denotes the $j_{th}$ token in the $i_{th}$ sentence.

First, original Prompt Tuning is applied by concatenating a series of prompt tokens $P_{en}$, parameterized by $\theta_{p_en}$, to the encoder input $X_{en} = \{e^1_1, e^1_2, \ldots, e^n_m\}$, where $e$ represents the embedding of each token (the leftmost structure in Figure 1). The gradients are backpropagated through the prompts and the weights $\theta$ of language model are frozen (Lester et al., 2021). In this way, the model maximizes the likelihood of the output $Y$:

$$p_{\theta; \theta_{p_en}}(Y|P_{en}; X_{en})$$ (1)

The result of original Prompt Tuning is shown on the first line in Table 1, where we see it severely underperforms versus full-model tuning. In further experiments, we added a series of prompts $P_{de}$ to the decoder inputs $X_{de}$ following the generation $p_{\theta; \theta_{p_de}}(Y|X_{en}, P_{de})$. Here, we found the results to be even worse than the last.

Necessary Prompts for Generation For generation-based tasks, prompts in both the encoder and decoder are equivalently useful. Therefore, our model employs a combination of the two series of prompts mentioned above, and generates $Y$ conditioning on $X_{en}$, $P_{en}$ and $P_{de}$:

$$p_{\theta; \theta_{p_en}; \theta_{p_de}}(Y|P_{en}; X_{en}; P_{de})$$ (2)

The result on the third line in Table 1 again verify our hypothesis. Prompts across the encoder and

\(^2\)Note that, throughout this work, a "sentence" can be an arbitrary span of contiguous text (e.g., fixed length of 10 tokens), or an actual linguistic sentence.
Table 1: Results of BART-base on CNN/DailyMail Datasets. Best results are bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompt in encoder</td>
<td>32.87</td>
<td>11.92</td>
<td>21.73</td>
</tr>
<tr>
<td>Prompt in decoder</td>
<td>26.77</td>
<td>11.73</td>
<td>16.71</td>
</tr>
<tr>
<td>Prompt in en.&amp;de.</td>
<td>36.37</td>
<td>14.41</td>
<td>24.46</td>
</tr>
<tr>
<td>Full-Model Tuning</td>
<td>37.01</td>
<td>14.49</td>
<td>23.91</td>
</tr>
</tbody>
</table>

Figure 2: Visualization of the encoder-decoder attention weights. The x-axis are the encoder input, including prompts across the encoder $P_{en}$ and the source document $X$. The y-axis are the decoder input, including prompts across the decoder $P_{de}$ and the target summary $Y$. The area in the red box represents the attentions of $P_{de}$ assigning to $P_{en}$. The area in the yellow box represents the attentions of $Y$ assigning to $X$. Darker color shows the more highly related associations between tokens.

decoder even achieve comparable results with full-model tuning under few-shot settings. This verifies two things for us. First, prepending simple prompts to only the input embedding layer is effective and efficient for few-shot abstractive summarization. Second, prompts across the encoder and decoder are both necessary for generation tasks.

Lack of Attention on the Document We further explored the encoder-decoder attention to investigate the effect of the prompts and freezing the language model. From Figure 2, we find the generating output is mainly focused on the soft prompts to come with little attention given to the document itself. This outcome is detrimental to summarization that requires to understand the semantics and inner discourse structure of documents (Wang et al., 2019). Without the associations of target summaries and source documents, it is impossible to obtain high-quality summaries using current prompt architectures.

From Figure 2, we can observe that prompts in the encoder and the ones in decoder are consistently and directly associated with each other. We speculate that the mechanism is that encoder prompts retrieve relevant knowledge from the frozen encoder language model as a document representation, and decoder prompts copy the encoder’s behaviour, guiding the decoder language model to generate text.

3 Method

In light of our findings about the current architectures, we developed a new architecture of pre-trained soft prompts, for few-shot abstractive summarization called PSP. The framework includes continuous prompts across the encoder and decoder inputs, as well as inner-prompts to capture the dependencies between documents and target summaries. To better understand a given document, we add a prompt pre-training process before few-shot tuning. It also brings a good initialization for the prompting. The overall architecture and training scheme are illustrated in Figure 3.

3.1 Encoder-Decoder Basic Prompts

As mentioned in Section 2, in the training phase of current architectures, $P_{en}$ is responsible for extracting knowledge from the encoder’s frozen language model as a document representation. Meanwhile, $P_{de}$ mostly copies the behavior of $P_{en}$ and guides the frozen decoder’s language model to generate fluent text as a summary.

To strengthen the model’s ability to understand a document, the dependencies and attentions given to the source document need to be embodied in the prompt architecture.

3.2 Inner-Prompts for Document Understanding

To achieve our goal, we propose the notion of adding inner-prompts within the source document,
is manipulated into a new "sentence" and a
tactic units, a text span with a fixed length
To discover more fine-grained se-
Fixed-length
In this way, the model can identify important sen-
tences to encode the document at sentence level.

Following Liu and Lapata (2019), the
Interval
For the first...what does best. Contenders told...of the proceedings.
Instead, there was Bob Barker, in 2007...Looking very...finished up....
Sequential
For the first...what does best. Contenders told...of the proceedings.
Instead, there was Bob Barker, in 2007...Looking very...finished up....
Fixed-length (k=10)

Figure 4: Different inner prompts for one example

Figure 4: Different inner prompts for one example

Figure 4: Different inner prompts for one example

Figure 4: Different inner prompts for one example

Figure 4: Different inner prompts for one example

Figure 4: Different inner prompts for one example

denoted as P_{in} = \{p_{in}^{1}, p_{in}^{2}, \ldots, p_{in}^{n}\} with the pa-
parameters \theta_{P_{in}} to be updated. Each p_{in}^{i} corresponds
to a single sentence. These inner-prompts are added
to the corresponding token embedding, which gives
rise to a new X'_{in}:

X'_{in} = \{e_{1}^{i} + p_{in}^{1}, e_{2}^{i} + p_{in}^{2}, \ldots, e_{n}^{i} + p_{in}^{n}\} \quad (3)

We believe that by prompting different semantic units (e.g., sentences, phrases, etc.), more attention
can be given to understanding the document’s discourse. Furthermore, the inner-prompts help the model to
clearly interpret the document by strengthening the associations between outputs and documents. What follows are three different strategies for incorporating the three different inner-prompts. Note that there is more discussion on this point in Section 4.2.

Interval Following Liu and Lapata (2019), the interval inner-prompts comprise two inner-prompt tokens are assigned to each sentence sent_{i}, depending on whether \(i\) is odd. Specifically,

P_{in} = \{p_{in}^{1}, p_{in}^{2}, \ldots, p_{in}^{(n-1) \text{mod} 2 + 1}\} \quad (4)

In this way, the model can identify important sentences to encode the document at sentence level.

Sequential To highlight the complex discourse structure of documents, sentence positions need to be considered. Therefore, different tokens are set in sentences by their sequences, formulated as:

P_{in} = \{p_{in}^{1}, p_{in}^{2}, \ldots, p_{in}^{n}\} \quad (5)

Fixed-length To discover more fine-grained semantic units, a text span with a fixed length \(k\) is manipulated into a new “sentence” and a

3.3 Self-supervised Prompt Pre-training

To improve the ability of the prompts to understand the documents and to help the model to adapt to the summarization tasks, soft prompts are further pre-trained on the corpus using summarization-oriented self-supervised objectives. Doing this also means that the prompts are well initialized for few-shot tuning.

We tested two strategies for constructing the self-supervised data. Each strategy was designed to suit a particular type of writing bias in the document. These are “lead” and “gap sentences generation”.

Lead Lead bias is common in news articles, which usually follow an inverted pyramid structure where the first few sentences contain the most salient information (See et al., 2017; Yang et al., 2020). With this type of bias, we initially select the first three sentences as our target summary, and the rest of the document as the source text.

With this type of prompt pre-training process, the model was able to infer the salient information based on the remaining text.

GSG Gap sentences generation applies to all documents that do not follow the lead bias structure (e.g., XSum (Narayan et al., 2018)). The strategy used here follows Zhang et al. (2020), where we used ROUGE1-F1 (Lin, 2004) between each sentence \(x_{i}\) and the rest of the document as a proxy for the principal score, \(s_{i} = \text{rouge}(x_{i}, D \setminus \{x_{i}\}), \forall i\). The top-\(m\) most important sentences were selected according to \(s_{i}\), and removed from the document. Then these \(m\) sentences are concatenated in the same order as the original text in the form of a pseudo summary. The remainder of the text is treated as a pseudo document.

With the constructed data, our designed prompts can be pre-trained and further tuned with few-shot examples.

3.4 Training Objective

The model is trained with maximum likelihood estimation (MLE). Given a ground-truth summary \(Y = [y_{1}, y_{2}, ..., y_{|Y|}]\) for an input passage \(X\), the objective is to minimize the negative log-likelihood
work. Following Lester et al. (2021), we used 100 with 5 different random seeds. We used the origi-

Table 2: Datasets statistics. “Avg.Passage” means the value and the standard deviation. Table 2 shows the statistics of the pre-processed corpus. The base version of BART was used in our Setup. To tune the hyper-parameters and se-

Table 3 presents the results of all PSP variants and baselines across CNNDM and XSum datasets. With the exception of the ROUGE-2 and ROUGE-L scores for the Prefix-Tuning on the CNNDM dataset, our proposed PSP, outperforms the others. However, PSP delivered a competitive result with only 3% of the parameters, which is an acceptable place to start. To our surprise, we observe that

Baseline Models We compared PSP to: Prompt Tuning (Lester et al., 2021), which only concatenates soft prompts into the encoder input; Prefix Tuning (Li and Liang, 2021), which adds a prefix to all the encoder layers, cross-attention layers, and the decoder layers; and Full-Model Tuning, which does not have any prompts and fine-tunes all the parameters of the pre-trained language model.

4.1 Experimental Results of Our Method

Table 3 presents the results of all PSP variants and baselines across CNNDM and XSum datasets. With the exception of the ROUGE-2 and ROUGE-L scores for the Prefix-Tuning on the CNNDM dataset, our proposed PSP, outperforms the others. However, PSP delivered a competitive result with only 3% of the parameters, which is an acceptable place to start. To our surprise, we observe that

We made our division at 85% to ensure all embeddings of inner-prompt tokens could be fully trained, because sentences after the n-th only exist in 15% of the data.

More information about implementation details are shown in Section A.2 in the appendix.
Table 3: Results on CNNDM and XSum Datasets. The experiments are conducted with 300 training samples and 300 validation samples on each dataset. We report the mean value and the standard deviation over 5 sampled datasets. $k = 10$ is chosen for PSP$_{fixed-k}$. “PPL” represents the perplexity of generated summaries. A low perplexity indicates the summaries are fluent. Best results are bold and underline means our models outperform Full-model tuning.

50% of PSP’s results surpass the full-model tuning, especially on XSum, as underlined in the table. Besides, results on the PPL metric show that PSP can generate more fluent summaries than other models. These results indicate that fine-tuning large language models is not necessarily a good or efficient idea with few-shot generation. It also shows that soft prompts with frozen language models are effective for few-shot abstractive summarization. Moreover, it statistically verifies that PSP with its three inner-prompt strategies is effective.

**Efficiency v.s. effectiveness.** We gave an overall comparison to baseline models on effectiveness and memory-efficiency, evaluated by ROUGE and the number of parameters, respectively. The results are shown in Table 4. Prompt Tuning has the least number of parameters, while its capacity is limited to this and lacks control over the decoder side, hence it can not perform natural language generation tasks well. We can see that substantial gains are made when going from vanilla Prompt Tuning to PSP. However, even if Prefix-Tuning is nearly thirty times more parameters than ours, there is either a marginal improvement or even performance decrease on some metrics. Besides, Prefix-Tuning relies on reparameterization tricks to stabilize the training, i.e., adds a MLP with large number of parameters to the training stage. Our method provides the best effectiveness-efficiency trade off, and outperforms full-model tuning with only 0.1% parameters, and presents competitive results against Prefix-Tuning with 3% parameters.

**Human Evaluation** We conducted a human evaluation study. To this end, we randomly selected 20 instances from the test set of each dataset. Ten graduate students with high levels of fluency in English were asked to assess the generated summaries and golden summaries from independent perspectives (Wang et al., 2021): **Informativeness** (how much useful information does the summary provide?), **Relevance** (how well does the summary reflect the input document?), and **Fluency** (how grammatically correct are the summary sentences and how easy are they to read?). Scoring followed the Best-Worst Scaling method (Kiritchenko and Mohammad, 2017). Participants were asked to select the best and worst summaries from each perspective. The scores were computed as the percentage of times a summary was chosen as the best minus the times it was selected as the worst. The scores ranged from -1 (worst) to 1 (best). Results are shown in Table 5. Qualitatively, we show several examples generated by different models and the reference in Table 1 and Table 15 in the appendix. Compared with all baselines, the summaries generated by PSP are always more fluent and relevant to the source document, consistent with the results of human evaluation. Further more, we found summaries generated by PSP and Prefix-Tuning are always similar in sentence patterns and expressions. However, Prefix-Tuning tends to generate texts shorter than PSP, which often leads to lack of information.
Table 5: Human evaluation results. Best results are bold.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CNNDM</th>
<th>XSum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IF</td>
<td>RL</td>
</tr>
<tr>
<td>PSP</td>
<td>0.500</td>
<td>0.708</td>
</tr>
<tr>
<td>Prompt Tuning</td>
<td>-0.317</td>
<td>-0.758</td>
</tr>
<tr>
<td>Prefix-Tuning</td>
<td>-0.233</td>
<td>0.067</td>
</tr>
<tr>
<td>Full-Model Tuning</td>
<td>0.067</td>
<td>-0.025</td>
</tr>
</tbody>
</table>

Table 6: Results of different fixed length $k$ on validation set $D_{dev}$ and test set $D_{test}$ of XSum. “R-1” is short for “ROUGE-1”, the same for “R-2” and “R-L”.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$D_{dev}$</th>
<th>$D_{test}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-2</td>
</tr>
<tr>
<td>5</td>
<td>34.27</td>
<td>11.90</td>
</tr>
<tr>
<td>10</td>
<td>35.31</td>
<td>12.88</td>
</tr>
<tr>
<td>15</td>
<td>34.98</td>
<td>11.68</td>
</tr>
<tr>
<td>30</td>
<td>34.48</td>
<td>12.57</td>
</tr>
</tbody>
</table>

Selection of fixed length $k$. As shown in Table 3, PSP$_{Fixed-k}$ performs consistently well on both datasets. So we further explored the influence of different length $k$, i.e., $k = 5, 10, 15, 30$, for inner-prompt tokens of the PSP$_{Fixed-k}$.

4.2 Analyses on Soft Prompts

Whether our model attends to understand documents? According to Figure 2, we further present the encoder-decoder attention distribution of the PSP. The comparison visualization is shown in Figure 5. We find the following enhancement of our model by introducing the inner prompts. First, the PSP model strengthens the associations between the encoder prompts and the decoder prompts compared to the original model. Second, the soft prompt $P_{en}$ has more opportunities to be related to the output $Y$, indicating the semantic relations between them. Third, the output $Y$ assigns more attention to the source document $X$. This suggests that the hidden structure of the document is emphasized, increasing the capability of understanding its semantics. As such, these prompts can properly select salient information from the document and prompt the model to generate the output.

Do inner prompts assist the model to understand the content of documents or simply increase the model's capacity? Instead of using inner-prompts, we prepended additional tunable tokens (i.e., 150 tokens) in front of the encoder and the decoder inputs. Comparison results are shown in Table 7. Despite the larger capacity, soft prompts with 150 tunable tokens before the input performed the worst, denoted as soft prompts (en.&de., 150). This suggests the inner-prompts with a few parameters do help to understand the document by prompting the structures, rather than simply add more trainable parameters to increase the model’s capacity.

Further insight on soft prompts across the encoder and the decoder. To verify our hypothesis that the decoder prompts largely copy the behaviour of the encoder prompts, we shared similar embeddings of the soft prompts before the encoder and the decoder. In Table 8, we observe the Soft prompts (en.&de., shared) and (en.&de., separate) almost...
A crucial strategy for PSP is the pre-training of soft prompts. To give a fair comparison, we performed prefix pre-training for Prefix-Tuning in the same way with the PSP. The results are shown in Table 10. We can find that the Prefix model obtains improvements on the XSum dataset after adopting the pre-training strategy, but underperforms the original one on the CNNDM dataset. It indicates that Prefix-Tuning shows limited potential compared to our model. We induce that the pre-training for Prefix-Tuning raises over-fitting risk due to its sensitivity to different data or parameter settings.

### 4.5 Ablation Study

We conducted experiments to examine the effectiveness of the major components of our model, and Table 11 shows the ablation results across the two datasets. We observed both the prompt pre-training operation and the inner-prompts component contribute to the main model. Notably, with the removal of each component, the model becomes considerably unstable, indicated by the variance shown in the ablation results. Comparably, prompt pre-training in our model accounts for more importance on the XSum dataset whose summaries have a higher abstract level (we assume it’s more “difficult”) than the CNNDM. In sum, these two components support the performance and stability of our model in terms of summarization adaption (by prompt pre-training) and structural documents understanding (by inner-prompts).

### 5 Related Work

**Few-Shot Abstractive Summarization**  
In practical application scenarios, the lack of manual constructed document-summary pairs or labeled data makes data-driven neural models perform badly (Hu et al., 2021, 2020). Fabbri et al. (2020) condense characteristics of the target dataset into Wikipedia data to construct pseudo-summaries. Bražinskas et al. (2020) introduce plug-in networks to reproduce characteristics of the target dataset with only a small set of labeled examples. Bai et al. (2021) conduct cross-lingual summarization
in a low-resource setting. Yu et al. (2021) design the second phase of pre-training on large-scale generative models before fine-tuning. In this paper, we construct pseudo-summary corpus with heuristic rules, providing a better parameter initialization for soft prompts under few-shot settings. More importantly, we design summarization-oriented soft prompts to help the model produce few-shot summaries.

**Prompt Learning**

The emergence of GPT-3 (Brown et al., 2020) introduces the concept of “prompting”. One only needs to assemble a task description and few examples into a prompt, and then prepend it to the task input. With the large-scale frozen parameters, a pre-trained model can generate the output without any task-specific tuning. However, task description is error-prone while there is no unified, explicit, and effective way to build these hard prompts manually (Logan IV et al., 2021). Hence, several works (Gao et al., 2020; Jiang et al., 2020; Shin et al., 2020) are proposed to generate prompts automatically, but they all restrict prompts to discrete spaces. These discrete prompts are less expressive and sub-optimal. To overcome the shortcomings of hard prompts, Li and Liang (2021) propose “Prefix-Tuning”. This method only tunes prefix activation prepended to all transformer layers, and keeps the LM parameters frozen. To further simplify, Prompt Tuning (Lester et al., 2021) only prepends tunable tokens to the encoder input, and keeps all other parameters frozen. Logan IV et al. (2021) and Gu et al. (2021) propose to use pre-training to boost the low performance of Prompt Tuning for few-shot learning. In this work, we fit the structure of Prompt Tuning to text generation models, proposing encoder prompts, decoder prompts, and inner prompts. We successfully apply prompt tuning methods to few-shot abstractive summarization task.

6 Conclusion

In this paper, we present a novel pre-trained soft prompts architecture (PSP) specifically designed for few-shot abstractive summarization. We design continuous input embeddings across an encoder and a decoder alongside several kinds of inner-prompts placed in the text, assisting the model better to understand documents and guide accurate generation. Empirical results find the necessity of using prompt pre-training for few-shot/zero-shot abstractive summarization. Extensive experiments and analyses show that the proposed PSP provides the best effectiveness-efficiency trade off among all the baseline methods.

7 Acknowledgments

The research presented in this publication was sponsored by CCF Fund For Young Scholars, and Joint Funds of the National Natural Science Foundation of China (Grant No. U21B2009).

References


A Appendix

A.1 Constructing Pseudo Data for Pre-training

We constructed the pseudo data for CNNDM with Lead. We also conducted a simple data cleaning procedure to the self-supervised pre-train corpus. First, we cleaned away irrelevant information, such as media names, reporter names or dates from the summaries. Second, for those summaries with less than 50 tokens, we iteratively collected the first sentence of the remaining text to the pseudo summary, until the length of summary reaches 70. This procedure was set up to prevent the target text from being too short to form a meaningful summary. Third, for those samples in which the source document is shorter than its summary, we filtered them out.

For XSum, we constructed the pseudo data for pre-training following GSG. The top-1 most important sentence was selected as the pseudo summary. Then we filtered out those pseudo summaries that are not relevant enough to the pseudo passages. In particular, we leveraged hand-written summaries in the few-shot dataset to determine the filtering threshold of pseudo data. We calculated the ROUGE-1 F1 between each ground-truth summary and its corresponding passage, represented as $R_i$. Then we calculated the mean and variance of $R_i$: $\epsilon = \frac{1}{n} \sum_{i=1}^{n} R_i$, $\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (R_i - \epsilon)^2$, and $\epsilon - \sigma^2$ was used as a lower-bound threshold to filter out low quality pseudo data. For those pseudo samples where ROUGE-1 F1 between the pseudo summary and the pseudo passage is lower than the threshold $\epsilon - \sigma^2$, we filtered them out. Finally, we conducted pre-training on our soft prompts with these filtered pseudo-data. Table 12 shows the statistics for the pre-training data corpus.

A.2 Implementation Details

We first split sentences with the Stanford CoreNLP toolkit (Manning et al., 2014), and the input documents were truncated to 1024 BPE tokens. We adopted BART-base for all the experiments. Our implementation was based on the Hugging Face Transformer models (Wolf et al., 2020). We used a mini-batch size of 8 with a gradient accumulation for 10 iterations. We used Adam optimizer with momentum $\beta_1 = 0.9$, $\beta_2 = 0.998$ and noam decay. In the stage of pre-training, the peak value of learning rate was 1e-3, and we set the warm up ratio to 10%. During fine-tuning, the peak value of learning rate was 3e-4, and we set the warm up steps to 100 with 400 epochs. In the decoding stage, we used beam search with a beam size of 4. The decoding process will not stop until an end-of-sequence (EOS) token was emitted or the length of the generated summary reached to 256 tokens. All models were trained on 4 TITAN RTX GPUs.

A.3 The Universality of GSG to Construct Pseudo-data

To demonstrate the universality of using the GSG method to construct pseudo-data for prompt pre-training, we conducted a complimentary experiment to testify its effect on the CNNDM. Specifically, we selected $m = 3$ important sentences. Results in Table 13 indicate that the PSP model pre-trained by GSG is equally effective with the original PSP Lead, showing that the GSG can be universally employed to pre-train soft prompts for abstractive summarization.

---

Table 12: Pseudo-summarization corpus statistics. “# of Original Passages” means the number of original passages in the training set, “# of Pre-training Data” means the number of pseudo data after data cleaning.

<table>
<thead>
<tr>
<th></th>
<th>CNNDM</th>
<th>XSum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pseudo Corpus</td>
<td>Pseudo Corpus</td>
</tr>
<tr>
<td># of Original Passages</td>
<td>287,113</td>
<td>204,017</td>
</tr>
<tr>
<td># of Pre-training Data</td>
<td>284,177</td>
<td>158,499</td>
</tr>
</tbody>
</table>

Table 13: Results on CNNDM by using the Lead and the GSG to construct pseudo-data for prompt pre-training.

<table>
<thead>
<tr>
<th></th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPLead (w/o inner-prompts)</td>
<td>37.66</td>
<td>15.07</td>
<td>24.52</td>
</tr>
<tr>
<td>PSG (w/o inner-prompts)</td>
<td>37.04</td>
<td>15.04</td>
<td>25.20</td>
</tr>
</tbody>
</table>

---

We do not conduct ablation experiments on XSum, as there is no “lead bias” in this dataset. So it is inappropriate to take the first sentences of the passage as the pseudo summary.
Manchester City are monitoring Juventus striker Alvaro Morata. The Spain international was a target for Arsenal a year ago when he left Real Madrid but opted for a move to Italy instead. Real have a buy-back option set at around £15 million but are unlikely to take that up.

Source: Nairobi, Kenya (CNN); University of Nairobi students were terrified Sunday morning when they heard explosions – caused by a faulty electrical cable – and believed it was a terror attack, the school said. Students on the Kikuyu campus stampeded down the halls of the Kimberly dormitory, and some jumped from its fifth floor, the university said. Hundreds were injured and were taken to hospitals. One person died, according to the school. The confusion and panic came less than two weeks after Al-Shabaab slaughtered 147 people at a college in Garissa, Kenya. Kenyan teachers and students have said they fear being targeted by the Somalia-based terrorists.

On Sunday, as many as 108 students from the University of Nairobi were admitted to Kenyatta National Hospital. Among them, at least 63 students have been discharged, and at least four are slated for surgery, the school said. Almost all of the 54 students being treated at PCEA Kikuyu Hospital have been released, the university said.

Table 14: Qualitative examples of CNNNDM.

| Source | Manchester City are keeping tabs on Juventus striker Alvaro Morata. The Spain international was a target for Arsenal a year ago when he left Real Madrid but opted for a move to Italy instead. Real have a buy-back option set at around £15 million but are unlikely to take that up. Manchester City are keeping tabs on Juventus striker Alvaro Morata ahead of possible summer bid. Morata closes down Lazio goalkeeper Federico Marchetti during Juventus’ Serie A win on Saturday. City also remain keen on Juve midfielder Paul Pogba but face competition from Paris Saint Germain. Morata has been a success at Juventus but the Italians are always susceptible to offers at the right price and are making plans to bring in Edinson Cavani from Paris Saint Germain despite the French club’s resistance. PSG also are one of the frontrunners to sign Juve midfielder Paul Pogba. City remain one of the clubs pushing to sign the France international too and Juventus have City’s Edin Dzeko, Aleksandar Kolarov and Stevan Jovetic on their list of potential targets for the summer. |
| PSP | Manchester City are keeping tabs on Juventus striker Alvaro Morata. The Spain international was a target for Arsenal a year ago when he left Real Madrid but opted for a move to Italy instead. Juven |
Brunon Kwiecien, 48, was convicted of planning a terrorist attack, illegal weapons possession and inciting two students to carry out an attack. He suggested he had been manipulated by Polish intelligence agents. Kwiecien was said to be fascinated with Norwegian mass killer Anders Behring Breivik. Right-wing extremist Breivik killed 77 people in a bombing and shooting rampage in Norway in July 2011. Kwiecien, a former professor at Krakow’s University of Agriculture, was arrested in 2012. Investigators believe he wanted to target parliament with four tonnes of explosives while then-President Bronislaw Komorowski and former Prime Minister Donald Tusk were due to attend, the trial heard. "If Brunon Kwiecien hadn’t been stopped, we would be talking amid the ruins of the state today," said judge Aleksandra Almert, according to the AFP agency. While admitting he planned to carry out the attack, he also said he was subject to "provocation" by the intelligence services. Kwiecien is the first Pole to be accused of terrorism, Polish media reported. He has no known links to established extremist groups.

<table>
<thead>
<tr>
<th>PSP</th>
<th>A Pole has been convicted of planning a terrorist attack in Poland, a court heard.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix-Tuning</td>
<td>A Pole has been convicted of planning to carry out a terrorist attack in Poland.</td>
</tr>
<tr>
<td>Prompt Tuning</td>
<td>A Polish man has been convicted of planning a terrorist attack in the Polish capital, Warsaw, on Thursday.</td>
</tr>
<tr>
<td>Full-Model Tuning</td>
<td>A Pole has been found guilty of planning a terrorist attack in the Polish state of Krakow.</td>
</tr>
<tr>
<td>Reference</td>
<td>A Polish university lecturer has been sentenced to 13 years in jail for plotting to ram a car packed with explosives into parliament.</td>
</tr>
</tbody>
</table>

Source

Schmidt was sent off by the referee for insulting Hoffenheim’s Julian Nagelsmann in Saturday’s 3-0 home loss. "That was nothing, what sort of a nutcase are you? Just shut your mouth," Schmidt shouted after going 2-0 down. The 49-year-old has been banned for two games and handed a 15,000 euros (£13,373) fine. The German was sanctioned after triggering a suspended sentence from February this year. He had been banned for three games, with a further two in the event of a repeat offence before June 2017, for refusing a referee’s order to leave the sidelines during a 1-0 defeat to Borussia Dortmund. Schmidt will be unable to have any contact with the team for half an hour before, during and after Tuesday’s German Cup second-round match against Lotte and Saturday’s league match against Wolfsburg. Leverkusen’s director of sport Rudi Voller has sought a meeting with the head of the disciplinary committee.

<table>
<thead>
<tr>
<th>PSP</th>
<th>Leverkusen defender Christian Schmidt has been banned for two games for insulting the referee.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix-Tuning</td>
<td>Leverkusen midfielder Matthias Schmidt has been banned for two games after refusing to leave the sidelines during a match against Wolfsburg.</td>
</tr>
<tr>
<td>Prompt Tuning</td>
<td>A Leverkusen midfielder Christian Schmidt has been banned for two games for insulting the referee in a game against Hoffenheim on Saturday.</td>
</tr>
<tr>
<td>Full-Model Tuning</td>
<td>Averkusen manager Gerhard Schmidt has been banned for two games for insulting the head of the German national team.</td>
</tr>
<tr>
<td>Reference</td>
<td>Bayer Leverkusen head coach Roger Schmidt has been banned and fined for calling an opposing manager &quot;a nutcase&quot; during a Bundesliga game.</td>
</tr>
</tbody>
</table>

Table 15: Qualitative examples of XSum.