fit@hcmus

Seminar

On Transferability of Prompt Tuning for Natural Language Processing

Indicator: Anh-Thu Le, Nhat-Truyen Nguyen, Minh-Dat Nguyen

Table of contents

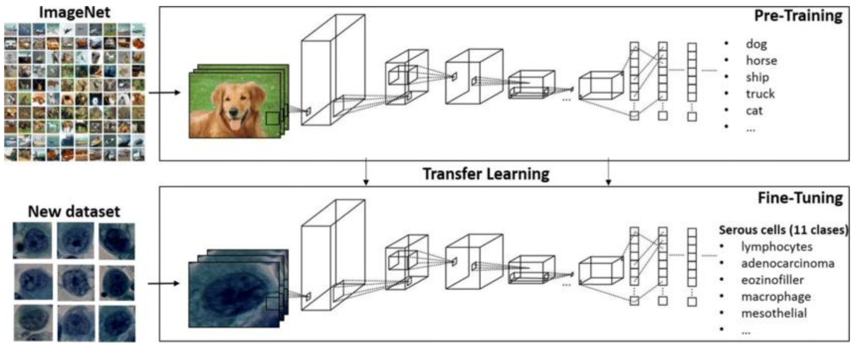
01

Background

Fine Tuning, Prompt Tuning

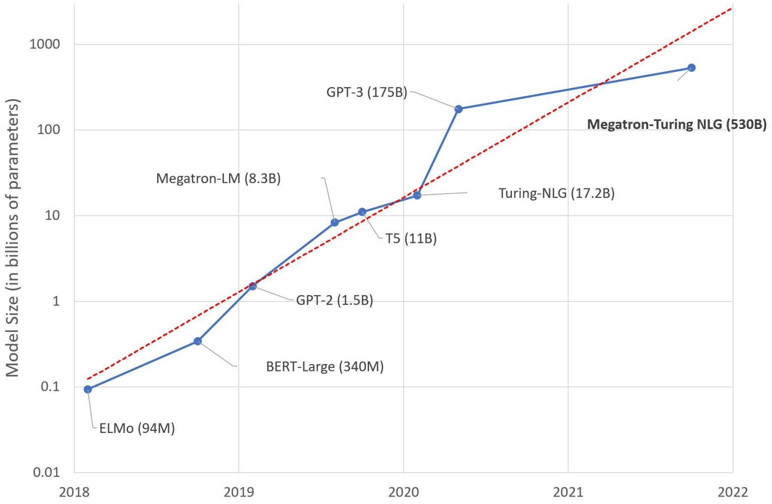
Definition

- Transfer learning method
- Use the pre-trained weights to train new dataset
- Better performance compare with directly train on our small dataset



Challenge

- Pre-trained Language Models (PLMs) development
- Require a large number of parameters
- Take a lot of computational resources

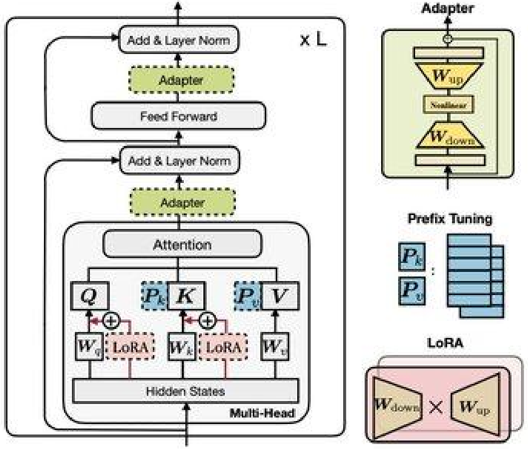


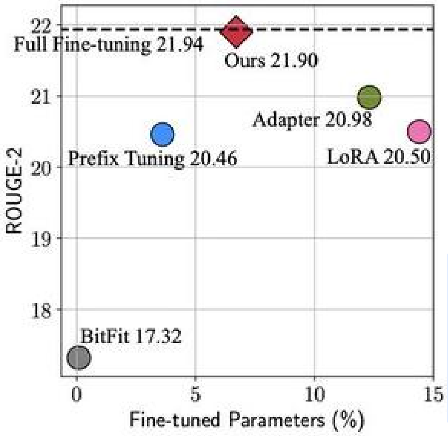
Overcome this challenge

Parameter-Efficient Fine Tuning (PEFT) Methods

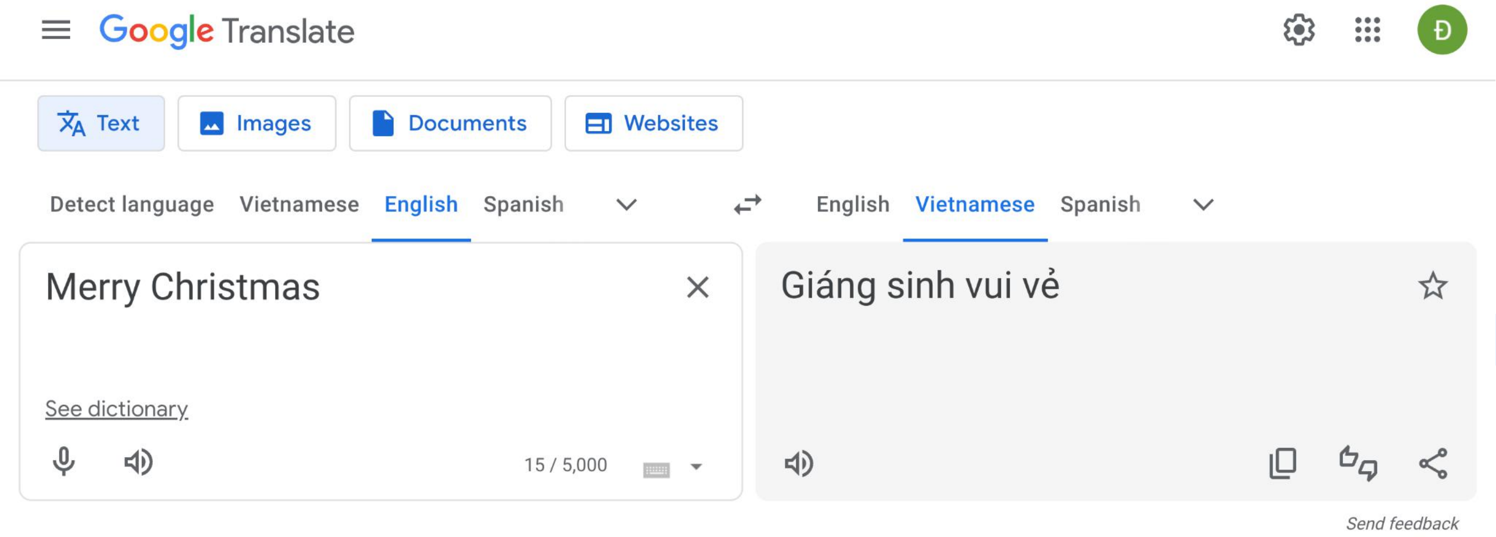
- Optimize a small part of parameters for downstream tasks while freezing the rest of the parameters of the PLM

Adapter	Neil, et al., Parameter-Efficient Transfer Learning Learning for NLP, ICML, 2019.
Prefix	Li, et al., Prefix-Tuning: Optimizing Continuous Prompts for Generation, ACL, 2021.
LoRA	Hu, et al., LoRA: Low-Rank Adaptation of Large Language Models, ICLR, 2022.
BitFit	Zaken, et al., BitFit: Simple Parameter-efficient Fine-Tuning for Transformer-based Masked Language-models, ACL, 2022.
Prompt	Lester et al., The Power of Scale for Parameter-Efficient Prompt Tuning, EMNLP, 2021.
Combine	He at al, Towards a Unified View of Parameter-Efficient Transfer Learning, ICLR, 2022.





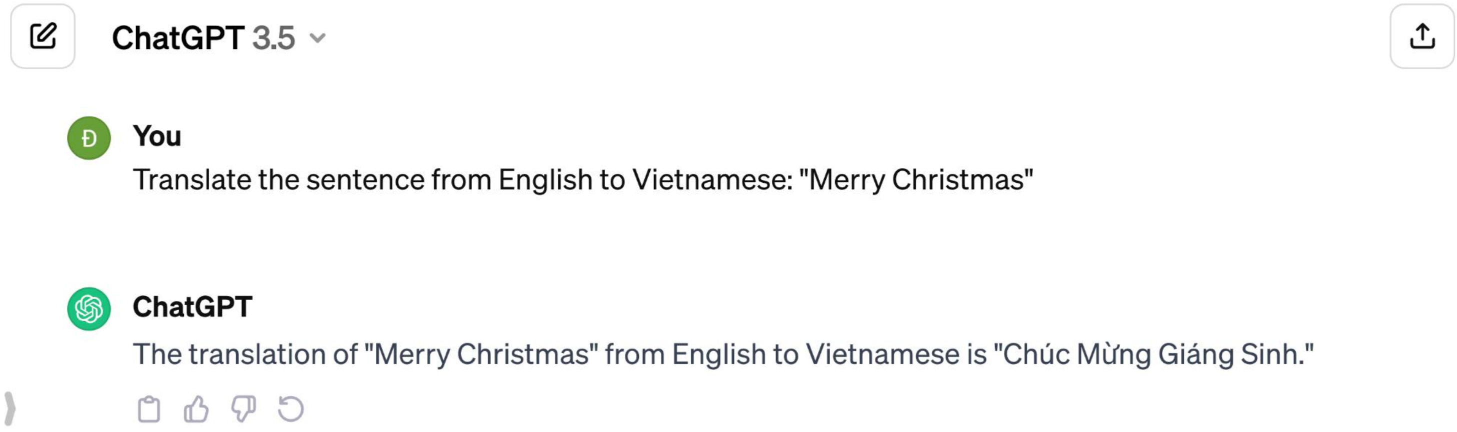
Fine Tuning



How to improve?

Update the model itself!

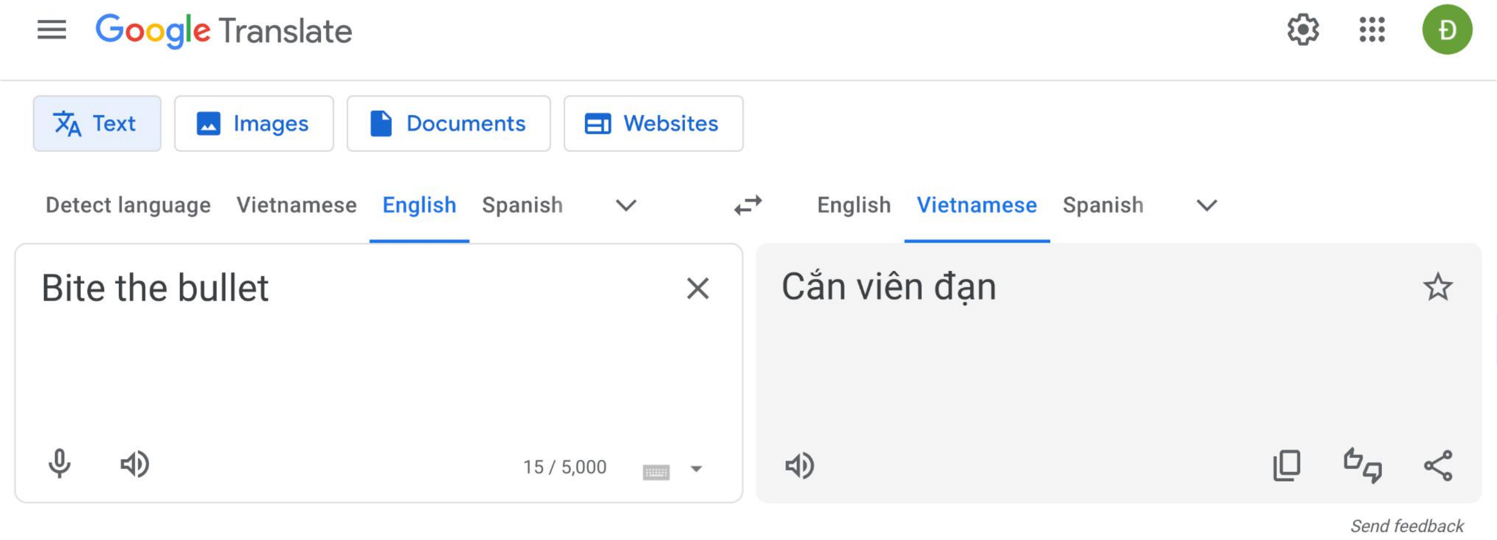
Prompt Tuning



How to improve?

Just tune the prompt!

When Google Translate fails



When ChatGPT fails

N **You**

Translate the sentence from English to Vietnamese: "Bite the bullet"

ChatGPT

The phrase "Bite the bullet" translates to "Nhai viên đạn" in Vietnamese.

N **You**

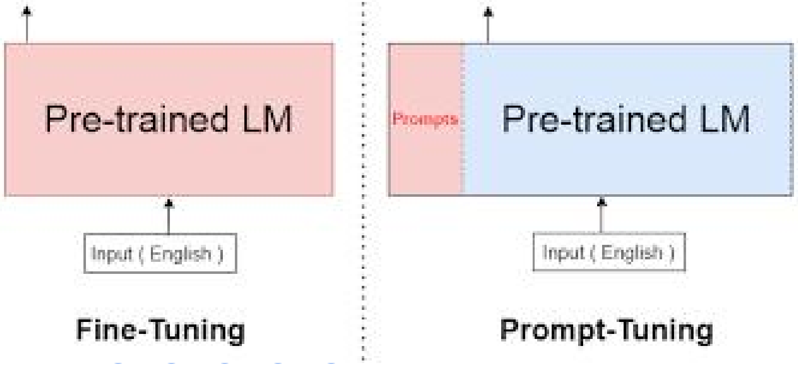
The sentence "Bite the bullet" is actually a collocation in English

ChatGPT

You're absolutely right! "Bite the bullet" is an English idiom meaning to endure a painful or difficult situation with courage and determination. In Vietnamese, this expression could be translated as "Nhẫn nhịn vượt qua khó khăn" or "Đối mặt với khó khăn một cách dũng cảm."

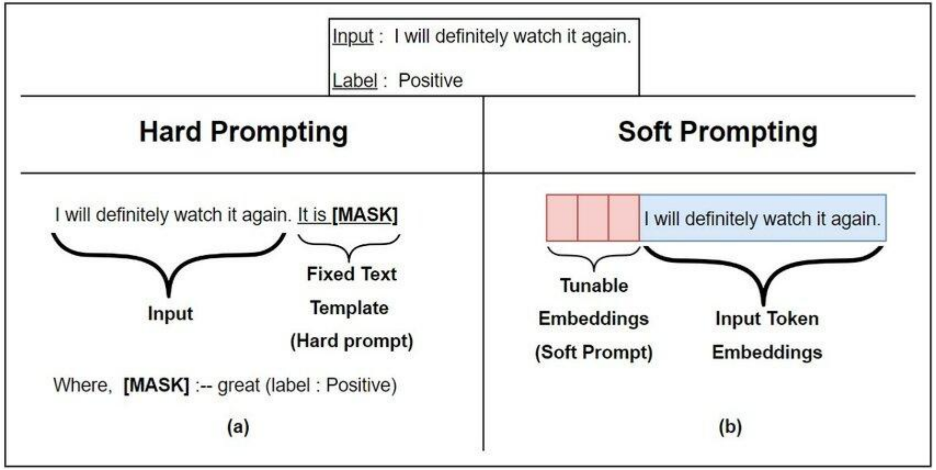
Definition

- Effectively fine-tuning pre-trained models on small datasets
- Use soft prompt to tune the output of a pre-trained model



Hard / Soft Prompt

<Prompt><Instance>



Pros/Cons

- **Advantage**: Lowest computation costs
- **Challenge**: Slow Convergence

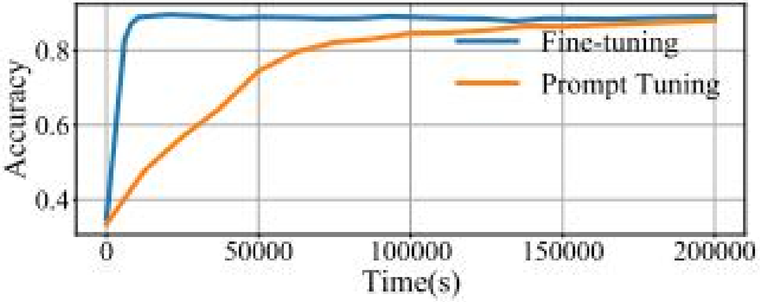


Figure 2: Validation accuracies against training time of fine-tuning and PT for RoBERTa$_{LARGE}$ on MNLI. PT takes much more training time.

Proposed Method

Key idea of this paper

- **Should we look at knowledge transfer?**
  - Because only soft prompts have been learned anyway, wouldn't it be possible to transfer knowledge by simply detaching and pasting the tokens? **Prompt Transfer**
  - Accordingly, this study investigates whether **Prompt Transfer** can contribute to improving **Prompt Tuning** effectively

02

Prompt Transfer

Cross-Task Transfer, Cross-Model Transfer

Settings for the experiment

Cross-Task Transfer:

- **17 NLP Tasks in 6 Categories:**
  - **Sentiment Analysis:** IMDB, SST-2, Laptop, Restaurant, Movie, Regionales, TweetEvel
  - **NLI:** MNLI, QNLI, SNLI
  - **Ethical Judgment:** Deontology, Justice
  - **Paraphrase Identification:** QQP, MRPC
  - **QA:** SQuAD, NQ-Open
  - **Summarization:** Multi-News, SAMSum

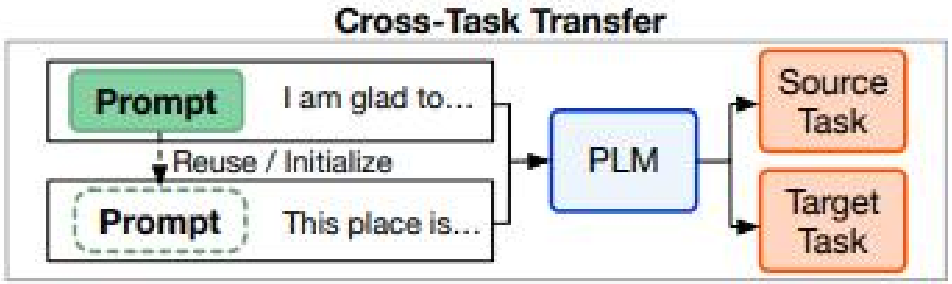
Cross-Model Transfer:

- **RoBERTa**
  - RoBERTa-large: MLM representative
- **T5**
  - T5-XXL: Representative seq2seq training

Target

- Check prompt transferability in zero-shot setting
- Check effectiveness and efficiency of prompt tuning with transfer

Implementation



- Since only the task is different, we experimented with only prompts being removed from the source task and attached to the target task

Prompt Input

- Input structure: **<soft><mask><text_a><text_b>**
- Example:

soft: Does the first sentence entails the second?

mask:

text_a: A soccer game with multiple males playing.

text_b: Some men are playing a sport.

Implementation

```
[8]  # cấu trúc input
     trainer.template.text

     [{'add_prefix_space': '', 'soft': None, 'duplicate': 100, 'same': True},
      {'add_prefix_space': ' ', 'mask': None},
      {'add_prefix_space': ' ', 'placeholder': 'text_a'},
      {'add_prefix_space': ' ', 'placeholder': 'text_b'}]
```

```
[9]  # độ dài soft prompt
     trainer.template.num_tokens

     100
```

```
[10] # kích thước prompt embedding (num_tokens, embedding_dim)
     trainer.template.soft_embeds.shape

     torch.Size([100, 768])
```

Train prompt

- Backbone: RoBERTa-base
- Dataset: **SST2**
- Task type: Sentiment analysis
- Evaluate accuracy: **77,17%**

Cross-task eval

- Zero-shot transfer
- Target dataset: **rotten_tomatoes**
- Task type: Sentiment analysis
- Evaluate accuracy: **72,13%**

PT on target dataset

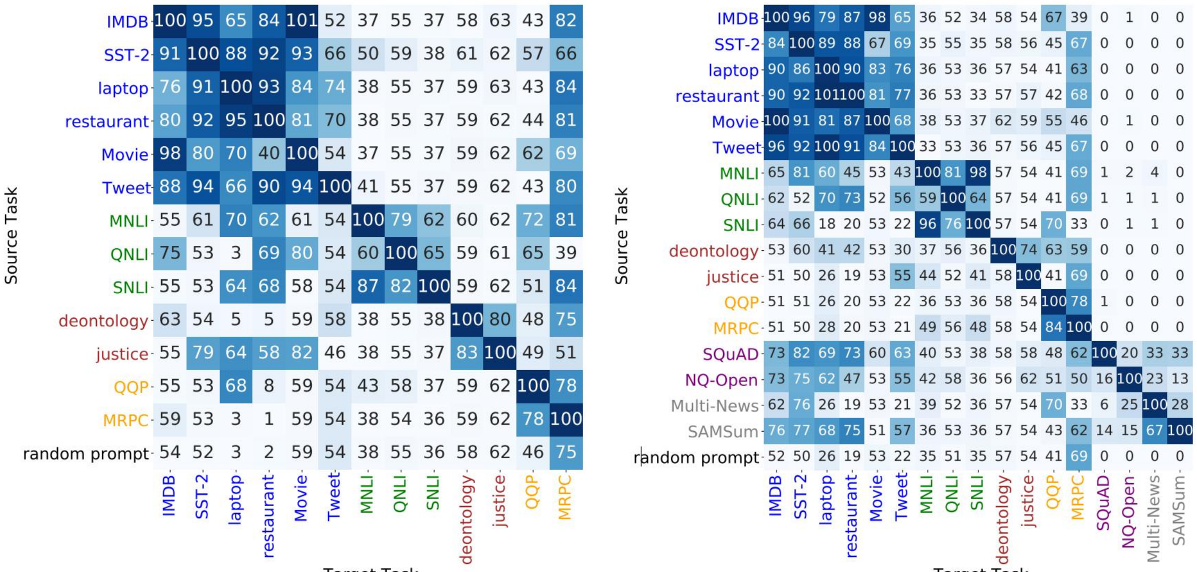
- Evaluate accuracy: **74,10%**

SST2: analyzing sentiment within sentences, at the sentence level.
rotten_tomatoes: expressing the overall sentiment towards movies.

Results

Similar or better performance than soft prompts starting from random initialization, and less training time

→ Soft prompts can be used for **similar tasks** without additional learning

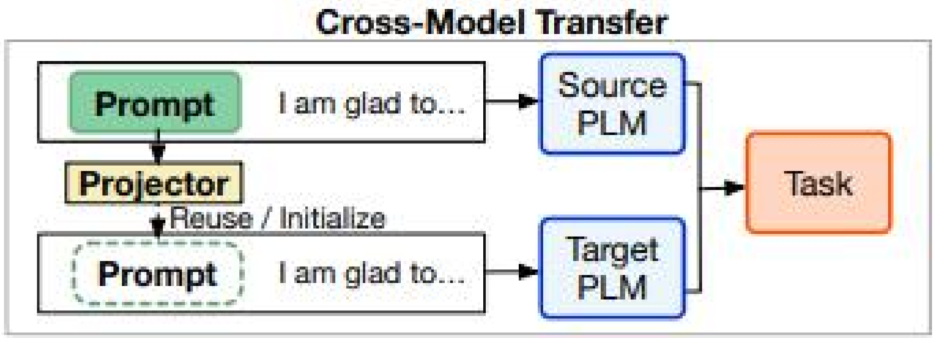


(a) RoBERTa$_{\text{LARGE}}$ (b) T5$_{\text{XXL}}$

Target

- Find an answer to whether soft prompts can be transferred from *small PLM* to *large PLM*
  - Source model: **RoBERTa** (small PLM)
  - Target model: **T5** (large PLM)

Implementation



- Because the models are different, soft prompts cannot be used directly as cross-tasks
- Prompt Projectors were used to ensure that soft prompts can be easily moved from the source model to the target model

Min distance

- Prompt trained on source model $\mathbf{P^s}$
- Prompt trained on target model $\mathbf{P^t}$
- Minimize the distance between Project($\mathbf{P^s}$) and $\mathbf{P^t}$

Example

$\mathbf{P^s}$ shape: (100, 768)

$\mathbf{P^t}$ shape: (100, 1024)

Task tuning

- Directly tune the projected prompts Project($\mathbf{P^s}$)
- Backpropagate the supervision signals to train the projector weights

Results

Task tuning is generalized even for **unseen tasks** in **similar type tasks**

Even if the model is **different**, if there is a **prompt projector**, learned soft prompts can be **used**

→ cross model prompt transfer seems **possible**

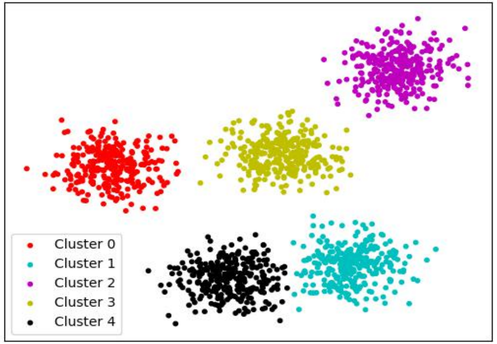
Method		SA						NLI		
		IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI
PT on $T5_{XXL}$		96.5	97.4	76.6	88.1	97.9	72.5	90.5	95.2	93.4
Random Prompt		49.7	49.0	19.8	17.0	51.6	15.5	31.8	49.3	31.9
(a) Zero-shot Transfer Performance (%)										
`laptop`	Distance Minimizing	49.6	49.0	76.6	17.5	51.5	14.4	31.8	48.1	32.8
	Task Tuning	**82.9**	**89.3**	**80.3**	**85.7**	**78.6**	**58.4**	32.4	50.7	33.6
`MNLI`	Distance Minimizing	49.6	50.1	19.8	18.3	51.2	15.0	**90.5**	49.0	32.9
	Task Tuning	49.7	48.8	19.8	17.0	51.6	16.0	89.8	**82.7**	**88.2**

Transferability indicators

Why can soft prompts transfer across tasks?

Embedding Similarity

- Compare similarities between trained prompts (Euclidean, cosine)
- Prompts of different tasks form distinguishable clusters



Model Stimulation Similarity

- Compare similarities between responses of PLM
- Compute the overlapping rate of activated neurons (ON)

→ Model stimulation is more important than embedding distances

03

Conclusion

Contribution

- This study shows the possibility that prompt transition can contribute to improving the efficiency of Prompt Transfer
    + Cross-task
    + Cross-model
- Explore possibilities for soft prompts initialization: accelerating training, ensuring efficiency & effectiveness
- Create metrics to determine which elements (and ultimately model stimulation) are effective for soft prompts in PLM

Limitation

Interpretability

Interpretability issues of soft prompts

Implementation

Incomplete implementation of prompt projector

The end

Thank you for listening
and feel free to ask any questions.

Implementation Notebooks

References:

[1]: On Transferability of Prompt Tuning for Natural Language Processing
[2]: The Power of Scale for Parameter-Efficient Prompt Tuning
[3]: Towards a Unified View of Parameter-Efficient Transfer Learning
[4]: Prompt as Parameter-Efficient Fine-Tuning