

Advanced NLP: Assignment 4

Hitesh Goel

2020115003

Motive of the assignment

Using the method of prompt tuning to finetune for various NLP tasks.

OneDrive link for pth files:

Link

It has the following contents:

- `models/`: All the `pth` files i.e. all the 3 models for the three different finetuning tasks. They have been named according to the hyperparameters used.
-

Running the files

Summarisation

```
cd summarisation/
```

```
python main.py
```

- set arguments in the `Config` class in `main.py` accordingly.
-

Question-Answering

```
cd qa/
```

```
python main.py
```

- set arguments in the `Config` class in `main.py` accordingly.
-

Machine Translation

```
cd mt/
```

```
python main.py
```

- set arguments in the `Config` class in `main.py` accordingly.

Note: Make sure the datasets are present in the `dataset` directories in each of the subdirectories of the tasks.

Theory

Question 1:

Concept of Soft Prompts: How does the introduction of “soft prompts” address the limitations of discrete text prompts in large language models? Why might soft prompts be considered a more flexible and efficient approach for task-specific conditioning?

Answer 1:

The advent of “soft prompts” provides a more adaptable and effective method for task-specific conditioning by addressing the drawbacks of discrete text prompts in big language models in multiple ways:

1. Adaptability and Flexibility

Soft prompts enable vector adjustments while maintaining the majority of the pre-trained model’s constituent parts. Because of its adaptability, the model can be used for a variety of jobs without requiring significant changes to the whole thing. Discrete text prompts, on the other hand, may require careful drafting for every activity, which limits their adaptability.

2. Modification Ease

Soft prompts are a quick and effective approach to switch between jobs because they are easily customizable for different tasks by just modifying the vectors. In contrast, single text prompts might need to be carefully thought out and crafted in order to produce the best results on particular tasks.

3. Decreased Manual Labor

With soft prompts, creating customized prompts for every task requires less physical labor. Rather than writing clear, human-readable instructions, the emphasis is on manipulating abstract vectors to increase automation. This contrasts with the time-consuming process of crafting skillfully worded separate text prompts.

4. Adjusting the Task without Changing the Model

Task adaptation can be facilitated by soft cues without requiring substantial modifications to the overall model. This helps with multi-task learning, as it enables a single model to seamlessly transition between tasks by changing the prompts. Discrete text prompts, on the other hand, can need for distinct modifications or even the development of unique models for every task.

5. More Accurate Token Length Measurement

Since soft prompts are based on vectors, they can employ fewer words overall. This accuracy in token length might be useful, particularly for more complicated operations. Conversely, discrete text prompts may be more involved and time-consuming, particularly for complex tasks.

6. Applications for a Range of Tasks

Soft prompts are useful for a variety of tasks, including as text summarization, question answering, sentiment analysis, and language translation. Their adaptability renders them appropriate for various tasks involving language comprehension. Discrete text prompts may be less flexible and more task-specific in a variety of contexts.

Question 2:

Scaling and Efficiency in Prompt Tuning: How does the efficiency of prompt tuning relate to the scale of the language model? Discuss the implications of this relationship for future developments in large-scale language models and their adaptability to specific tasks.

Answer 2:

The efficiency of prompt tuning is closely tied to the scale of the language model, and the implications of this relationship have significant ramifications for future developments in large-scale language models and their adaptability to specific tasks.

The efficiency of prompt tuning is highlighted as a competitive technique for adapting frozen pretrained language models to downstream tasks, as the model scales up in size. Implications: As language models grow larger, prompt tuning remains competitive, suggesting that the method is effective in leveraging the increased model capacity to adapt to a diverse set of tasks. This has positive implications for the scalability and versatility of large-scale models. Task Performance and Model Size:

The paper indicates that on the SuperGLUE benchmark, task performance with prompt tuning rivals traditional model tuning, with the gap diminishing as the model size increases. Implications: Larger models demonstrate improved task performance with prompt tuning, emphasizing that as models scale, their adaptability and ability to capture nuanced task-specific information become more pronounced. Improved Generalization in Zero-shot Domain Transfer:

The study suggests that prompt tuning leads to improved generalization in zero-shot domain transfer, indicating that the efficiency of this technique extends beyond traditional task-specific adaptation. Implications: Larger models, with

their enhanced generalization capabilities, benefit from prompt tuning in zero-shot domain transfer. This is crucial for applications where adaptability across diverse domains is essential. Avoidance of Overfitting to Specific Domains:

The paper suggests that freezing general-purpose language understanding parameters and restricting downstream learning to a lightweight parameter footprint through prompt tuning can help avoid overfitting to specific domains. Implications: Larger models, with their increased capacity to capture general language understanding, can use prompt tuning to avoid overfitting and achieve better generalization across tasks, indicating a potential solution for domain-agnostic model adaptation. Storage and Serving Cost Efficiency:

The discussion in the paper highlights the appeal of moving to frozen pre-trained models in terms of storage and serving costs, enabling efficient multi-task serving and high-performing prompt ensembling. Implications: As models continue to scale, the efficiency gains from prompt tuning contribute to cost-effective storage and serving. This has practical implications for real-world deployment of large-scale language models in resource-constrained environments. Task-Defining Parameters and General Language Modeling Parameters:

The paper suggests factoring out task-defining parameters as distinct from general language modeling parameters, providing new avenues for research. Implications: The relationship between prompt tuning and model scale opens up opportunities for innovative research, emphasizing the importance of understanding task-specific parameters and their interaction with general language understanding parameters in large-scale models.

Note: References used for this answer are: [link](#)

Analysis

Hyperparameters Used:

1. `batch_size`: 2 or 4. CUDA goes out of memory after this. **Gradient accumulation** has been used.
2. `num_epochs`: 10. However, **early stopping** has also been used.
3. **gradient clipping**: Gradients have been clipped with a norm of 1.0 to prevent exploding gradients.
4. `learning_rate`: 0.01
5. `optimiser`: AdamW
6. `metric`: Bleu score

Results achieved

1. Summarisation

- Best Loss achieved: 3.2
- Bleu score on test set: 0.314

2. Question Answering

- Best Loss achieved: 1.32
- Bleu score on test set: 0.281

3. Machine Translation

- Best Loss achieved: 4.56
 - Bleu score on test set: 0.246
-