Master thesis

University of Tartu

March 14, 2023

1 Meeting notes 5

1. Yova suggested that it is better to implemented prompt/prefix tuning, and adaptors ourselves since it would give complete control, which would allow us to understand why something isn't working. However, there is a big downside for this approach. My implementation probably would contain bugs, which I would need to fix, and this might take a lot of time. And we are short on time.

2. Perform instructive in-context-learning with T5-XXL(no weights update, just inference)

3. Write up about parameter efficient fine-tuning methods. This would let me understand the concepts better. It is great idea and incentive to write an articles on medium portal.

4. Regarding thesis experiments we don't need to predict multiple answers(parametric and contextual). Single answer would be enough.

5. Training won't be sequential as I thought before, instead we would train in bulk. For example, baseline would be trained on factual knowledge, 'f+cf' would be trained on combination of factual and counterfactual knowledge, and so on. However, the amount of data would be large(at most 280k).

6. We will be using data augmentation for all the experiments. For example:

   Data aug + Fine-tuning

   Data aug + Soft prompt

   Data aug + Adversarial training

   Data aug + ...

7. Implement adversarial training

8. Fine tuning changes all the weights of the model, thus this approach nudges model to memorize knowledge instead of learn algorithm. Thus by tuning only small(0.1%) subset of extra parameters we want instead algorithmic thinking emerge i.e. take an answer from the context rather then weights.

   NOTE: So, would it be true then that low capacity network should learn better algorithm better, than high capacity networks? My assumption is that low capacity networks are limited in terms of memory, thus in order to achieve good performance they should be more prone to learn algorithm. I think overall it is not true, because apparently memorize data easier, then learn algorithm how to process them. In order to reason about the language our model should have model of the world.

9. Learn about LoRA and write medium article.

Todos:

1. Prepare counterfactual data
2. Run experiment for different prefix length
3. Learn and write article about LoRA
4. Implement Prompt Tuning
5. Implement Adversarial training
6. (Maybe) Implement Prefix-Tuning, and Adapter.
7. Read HPC documentation, to understand allocation of two machines.
8. Setup wandb
9. Submit code for finetuning t5

Desirable outcome:

1. By the next week I want to have data
2. Implement adversarial training
3. T5-XXL inference for instructive in-context-learning

4. Design all the experiments
5. Try different length for prefix tuning
6. Read LoRA paper

References