Large Language Models

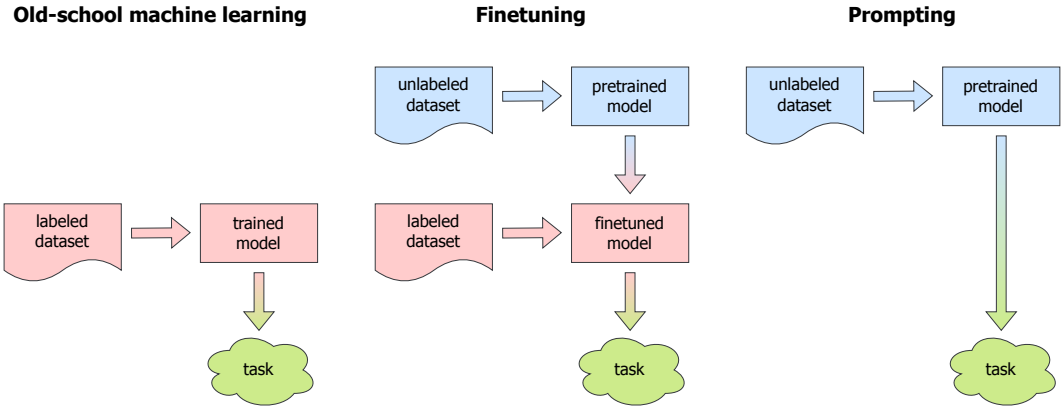
Part 6: Prompting

David Samuel

Language Technology Group, University of Oslo



Prompt-based learning



Formal definition

- Supervised learning: we have a dataset of input-output pairs and we learn to assign the correct output y to any input x
  - We directly optimize $P(y|x;\theta_S)$
- But: We don't always have a large enough annotated dataset!
- Then: Let's use a language model that learns $P(x|\theta_{LM})$ of any text x and reformulate the task so that we can approximate $P(y|x;\theta_S)$ by language modeling
  - We will use a prompt template f that can transform any input-output pair x, y into natural text: $f(x, y)$

$$argmax_y\ P(y|x;\theta_S) \approx argmax_y\ P(f(x,y)|\theta_{LM})$$

Example (filled prompt)

- $x =$ *"I really enjoyed this movie."*
- $y =$ *"positive"*, $y' =$ *"negative"*

Example (filled prompt)

- $x =$ *"I really enjoyed this movie."*
- $y =$ *"positive"*, $y' =$ *"negative"*

- $f =$ "[X] *To sum up, it was a* ['good' if $Y ==$ 'positive' else 'bad'] *movie."*

Example (filled prompt)

- $x =$ *"I really enjoyed this movie."*
- $y =$ *"positive"*, $y' =$ *"negative"*

- $f =$ "[X] *To sum up, it was a* ['good' if $Y ==$ 'positive' else 'bad'] *movie."*

- $f(x, y) =$ *"I really enjoyed this movie. To sum up, it was a good movie."*
- $f(x, y') =$ *"I really enjoyed this movie. To sum up, it was a bad movie."*

Example (filled prompt)

- $x =$ *"I really enjoyed this movie."*
- $y =$ *"positive"*, $y' =$ *"negative"*
- $f =$ "[X] *To sum up, it was a* ['good' if $Y ==$ 'positive' else 'bad'] *movie."*
- $f(x, y) =$ *"I really enjoyed this movie. To sum up, it was a good movie."*
- $f(x, y') =$ *"I really enjoyed this movie. To sum up, it was a bad movie."*
- Predict *"positive"* iff:

$$P(f(x, y)|\theta_{LM}) > P(f(x, y')|\theta_{LM})$$

Cloze and prefix prompts

- This was an example of a filled prompt, which contains both the input x and answer y
- But some tasks have to be generated for efficiency (translation, summarization) and some APIs don't allow for probability estimation

Cloze and prefix prompts

- This was an example of a filled prompt, which contains both the input x and answer y
- But some tasks have to be generated for efficiency (translation, summarization) and some APIs don't allow for probability estimation

- Then we need cloze prompts (for MLMs) or prefix prompts (for CLMS)
- Here, the prompt template f only transforms x and the missing y is generated by a language model

Example (cloze prompt)

- $x =$ *"I felt asleep in the middle."*
- $y =$ *"positive"*, $y' =$ *"negative"*

Example (cloze prompt)

- $x =$ *"I felt asleep in the middle."*
- $y =$ *"positive"*, $y' =$ *"negative"*
- $f =$ "[X] *To sum up, it was a* [Y] *movie."*

Example (cloze prompt)

- $x =$ *"I felt asleep in the middle."*
- $y =$ *"positive"*, $y' =$ *"negative"*
- $f =$ "[X] *To sum up, it was a* [Y] *movie."*
- $f(x) =$ *"I felt asleep in the middle. To sum up, it was a* [Y] *movie."*

Example (cloze prompt)

- $x =$ *"I felt asleep in the middle."*
- $y =$ *"positive"*, $y' =$ *"negative"*
- $f =$ "[X] *To sum up, it was a* [Y] *movie."*
- $f(x) =$ *"I felt asleep in the middle. To sum up, it was a* [Y] *movie."*
- Now we have to generate text to replace [Y]

Example (cloze prompt)

- $x =$ *"I felt asleep in the middle."*
- $y =$ *"positive"*, $y' =$ *"negative"*
- $f =$ "[X] *To sum up, it was a* [Y] *movie."*
- $f(x) =$ *"I felt asleep in the middle. To sum up, it was a* [Y] *movie."*
- Now we have to generate text to replace [Y]
  - *"I felt asleep in the middle. To sum up, it was a boring movie."*
- Our language model decided to generate *"boring"*, we have to prepare a mapping function that aligns *"boring"* with *"negative"*

Example (prefix prompt)

- $x =$ *"lang og svingete vei"*
- $y =$ any possible English translation

Example (prefix prompt)

- $x =$ *"lang og svingete vei"*
- $y =$ any possible English translation
- $f =$ *"The Norwegian sentence '*[X]*' can be translated as:* [Y]

Example (prefix prompt)

- $x =$ *"lang og svingete vei"*
- $y =$ any possible English translation
- $f =$ *"The Norwegian sentence '*[X]*' can be translated as:* [Y]
- $f(x) =$ *"The Norwegian sentence 'lang og svingete vei' can be translated as:* [Y]

Example (prefix prompt)

- $x =$ *"lang og svingete vei"*
- $y =$ any possible English translation
- $f =$ *"The Norwegian sentence '*[X]*' can be translated as:* [Y]
- $f(x) =$ *"The Norwegian sentence 'lang og svingete vei' can be translated as:* [Y]
- Now we have to generate text to replace [Y]

Example (prefix prompt)

- $x =$ *"lang og svingete vei"*
- $y =$ any possible English translation
- $f =$ *"The Norwegian sentence '*[X]*' can be translated as:* [Y]
- $f(x) =$ *"The Norwegian sentence 'lang og svingete vei' can be translated as:* [Y]
- Now we have to generate text to replace [Y]
  - *"The Norwegian sentence 'lang og svingete vei' can be translated as: long and winding road"*
- In the case of machine translation, the generated text can be directly used as y

Zero-shot and few-shot learning

- zero-shot == no supervised data
- few-shot == ”few” supervised samples of data
  - For example, “8-shot” means that we show a model 8 gold input-output pairs (so-called “shots”)
- Usually, no training is involved, we don’t change the weights as in fine-tuning
- The gold input-output pairs are a part of a prompt, it shows what is the expected behavior

Zero-shot and few-shot learning

- zero-shot == no supervised data
- few-shot == ”few” supervised samples of data
  - For example, “8-shot” means that we show a model 8 gold input-output pairs (so-called “shots”)
- Usually, no training is involved, we don’t change the weights as in fine-tuning
- The gold input-output pairs are a part of a prompt, it shows what is the expected behavior
- Much more efficient than fine-tuning, simpler, no overfitting, no need for large annotated datasets

Zero-shot and few-shot learning

- zero-shot == no supervised data
- few-shot == ”few” supervised samples of data
  - For example, “8-shot” means that we show a model 8 gold input-output pairs (so-called “shots”)
- Usually, no training is involved, we don’t change the weights as in fine-tuning
- The gold input-output pairs are a part of a prompt, it shows what is the expected behavior

- Much more efficient than fine-tuning, simpler, no overfitting, no need for large annotated datasets
- But we need a huge LM with emergent abilities and the performance is mediocre compared to fine-tuning on a sufficiently large dataset

Example (3-shot prefix prompt)

- x = *"ahh, what a waste of time :("*
- y = *"positive"*, y' = *"negative"*

- f = *"I felt asleep in the middle. -> negative; I really enjoyed this movie. -> positive; What an amazing ending! -> positive;* [X] -> [Y]"

- $f(x)$ = *"I felt asleep in the middle. -> negative; I really enjoyed this movie. -> positive; What an amazing ending! -> positive;* ahh, what a waste of time :(-> [Y]"

More information

- We have only scratched the surface of prompt-based learning
- If you want to learn more the the survey paper by Liu et al. (2023) is a good start!

References I

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).