Large Language Models

Part 7: Instruction-tuning and preference optimization

David Samuel

Language Technology Group, University of Oslo



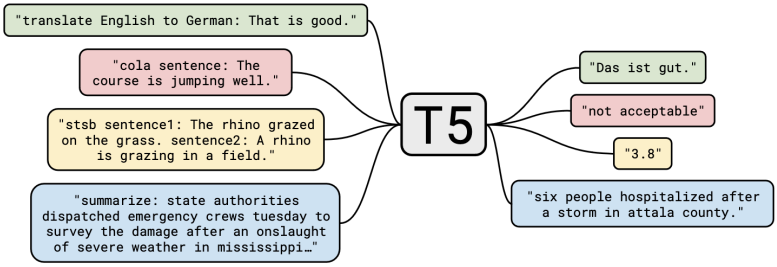
Alignment

- LMs are pretrained on enormous amounts of texts $\rightarrow$ they can imitate all kinds of characters and domains: article by a nuclear physicist, blogpost by a conspiracy theorist, or a mean comment by an internet troll
- A big part of prompting is persuading LMs to play the correct role -> we need to align their behavior with the user intent

Alignment

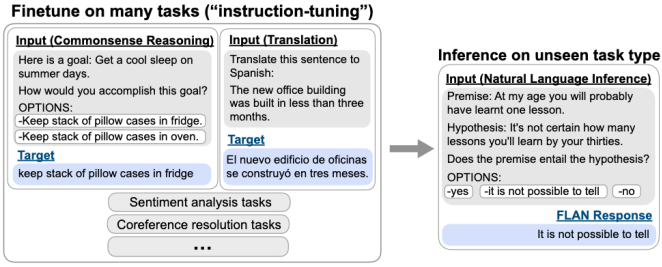
- LMs are pretrained on enormous amounts of texts $\rightarrow$ they can imitate all kinds of characters and domains: article by a nuclear physicist, blogpost by a conspiracy theorist, or a mean comment by an internet troll
- A big part of prompting is persuading LMs to play the correct role -> we need to align their behavior with the user intent
- Or we can use instruction finetuning
  - We finetune a general LM to act as an obedient expert assistant that factually answers our questions

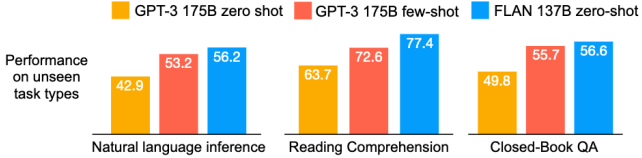
Remember T5?



- Let's take a language model and finetune it on a bunch of tasks in the T5 instruction-prompt style
- Will it generalize to unseen tasks?

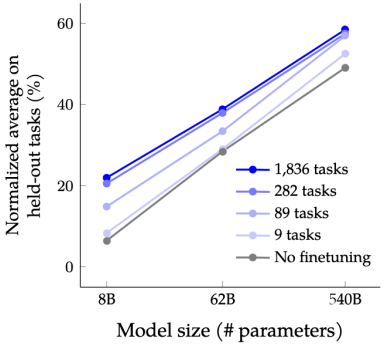
FLAN – Fine-tuned LANguage models (Wei et al., 2022)

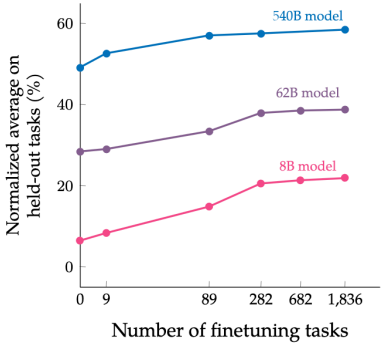




FLAN

- Will it generalize? Yes! It substantially outperforms “pure” language models
- FLAN (Wei et al., 2022) was fine-tuned on a collection of 60 tasks
- Later, FLAN-T5 and FLAN-PaLM (Chung et al., 2022) was fine-tuned on 1 836 tasks





What about ChatGPT and similar models?

- Basically instruction-finetuning with conversational responses
- But there's an additional step of preference optimization, which tries to align the responses with human preferences
- In the end, there are three stages:
  1. Supervised finetuning (SFT)
  2. Reward modeling
  3. Preference optimization via reinforcement learning

Reward modeling

- The goal is to get a model that can score a query-response pair according to human preferences
- We need a special dataset with such preferences
  - Usually, a sample consists of a query x, and a preferred and non-preferred response y_+ and y_-, respectively
- Then we model the preferences with Bradley-Terry model (BT) by minimizing:

$$\mathop{\mathbb{E}}_{x,y_+,y_-\sim\mathcal{D}} -\log\sigma(\log(\, r_\phi(x,y_+) - r_\phi(x,y_-)\,)),$$

where r_ϕ is our reward model that returns a continuous scalar score, and σ is the sigmoid function

Preference optimization via reinforcement learning

- Now we have a reward model that approximates human preferences, the next step is to finetune our chatbot to respond in a way that maximizes the rewards
- Formally:

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)} r_\phi(x, y),$$

  where π_θ is a policy (RL term) – in our case, it is a probability distribution of responses given a query
- Usually, we also want the policy to stay close to the original SFT model (to not overfit on the reward model):

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)} [r_\phi(x, y) - \beta D_{KL}(\pi_\theta(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x))]$$

Preference optimization via reinforcement learning

- Optimizing these terms is hard
- This is reinforcement learning
  - In supervised learning, we exactly know which output is correct and which is incorrect, optimizing the models is easy
  - Here we don't know what is correct or not, we only get a reward signal after trying one option
- This is a big subfield of deep learning, quite complicated, we don't have time to properly cover it
- There are multiple algorithms for learning an optimal policy, for example Proximal Policy Optimization (PPO, Schulman et al. (2017); used for the original ChatGPT models), Group Relative Policy Optimization (GRPO Shao et al. (2024), new DeepSeek reasoning models), or REINFORCE (Williams, 1992)

Preference optimization via reinforcement learning

- REINFORCE is the most straightforward one and often suffices (Ahmadian et al., 2024)
- Here, we minimize the loss function with gradient:

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)} R(y,x)\nabla_\theta \log \pi_\theta(y|x),$$

  notice how similar this is to the usual cross-entropy / negative log-likelihood loss (NLL)!
- In practice, we usually use REINFORCE with baseline, which readjusts the reward with a baseline reward $b(x)$:

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)} (R(y,x) - b(x))\nabla_\theta \log \pi_\theta(y|x),$$

Direct preference optimization (DPO)

- If we're okay with sacrificing some preformance, we don't have to use reiforcement learning
- Direct Preference Optimization (DPO) doesn't learn a separate reward model, instead it directly optimizes the policy from preference pairs:

$$\mathop{\mathbb{E}}_{x,y_+,y_-\sim\mathcal{D}} -\log\sigma\left(\beta\log\frac{\pi_\theta(y_+|x)}{\pi_{\text{ref}}(y_+|x)} - \beta\log\frac{\pi_\theta(y_-|x)}{\pi_{\text{ref}}(y_-|x)}\right)$$

References I

Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. (2024). Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.