How Can We Use Large Language Models (LLMs): A Brief Tutorial

Networked Intelligence for Comprehensive Efficiency (NICE) Lab
College of Information Science and Electronic Engineering
Zhejiang University
http://nice.rongpeng.info/

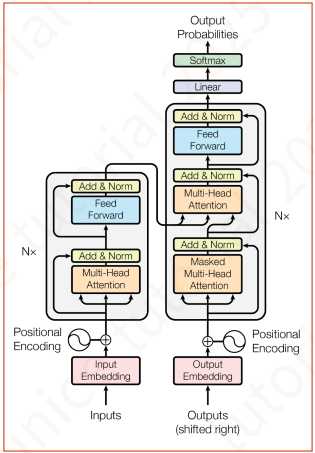
Aug. 18, 2025

Content

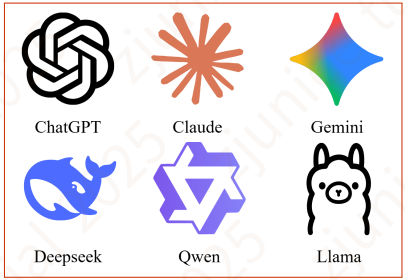
Decoder-only Architecture

Introduction to Causal Transformer

Open AI **GPT** Series: Generative Pre-trained **Transformer**[1]

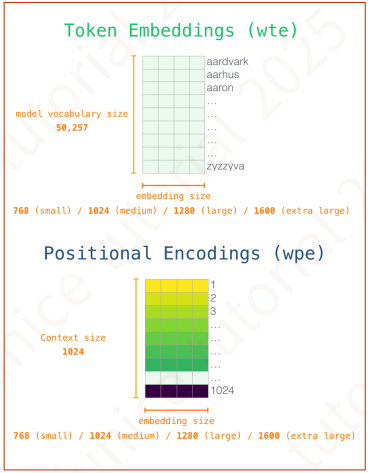


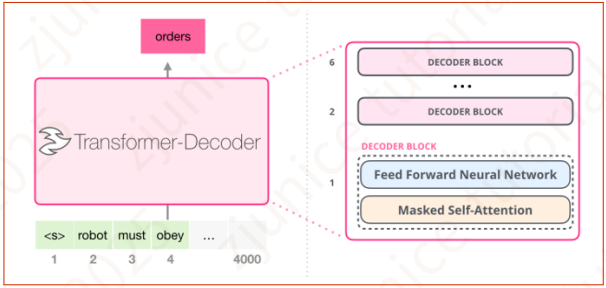
- GPT Series are **Decoder-only** transformer (**causal transformer**) architecture
- Most LLMs adopt the decoder-only architecture.



[1] A. Vaswani et al., "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

Components of Causal Transformer[2]

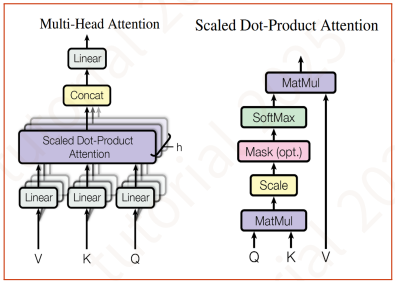


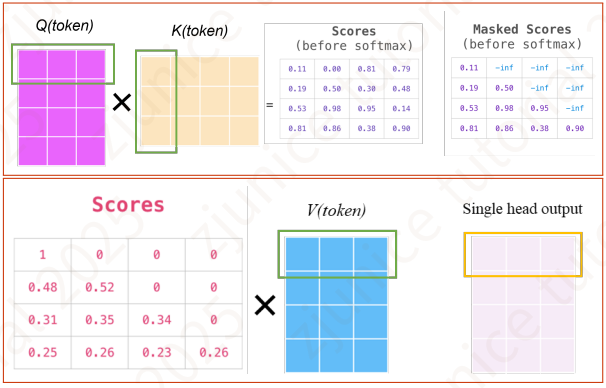


- Architecture: Decoder-only transformer
- Token embedding: Switch token and vector
- Position embedding: Add positional information to token embeddings

[2] Visualizing Transformer Language Models, https://jalammar.github.io/illustrated-gpt2

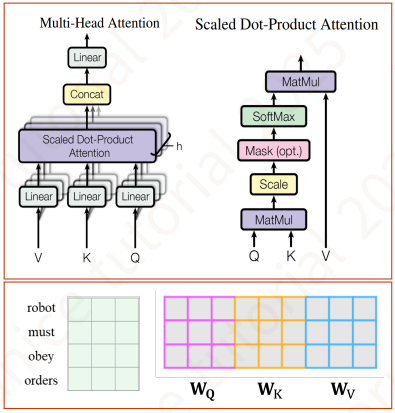
Masked Self-Attention



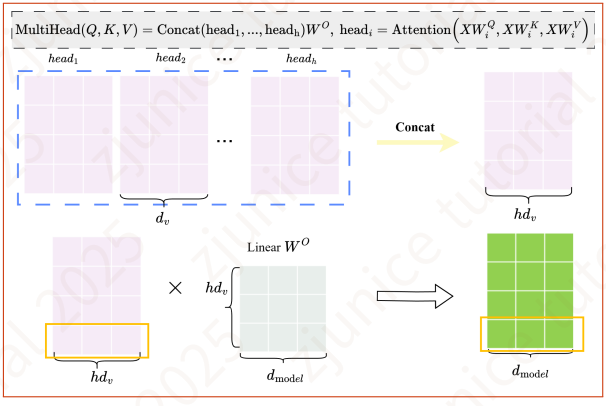


- $\text{softmax}(\frac{QK^T}{\sqrt{d}})V$: Compute attention scores between tokens in the input sequence
- Masked: Prevent information leakage from future tokens during training

Masked Self-Attention



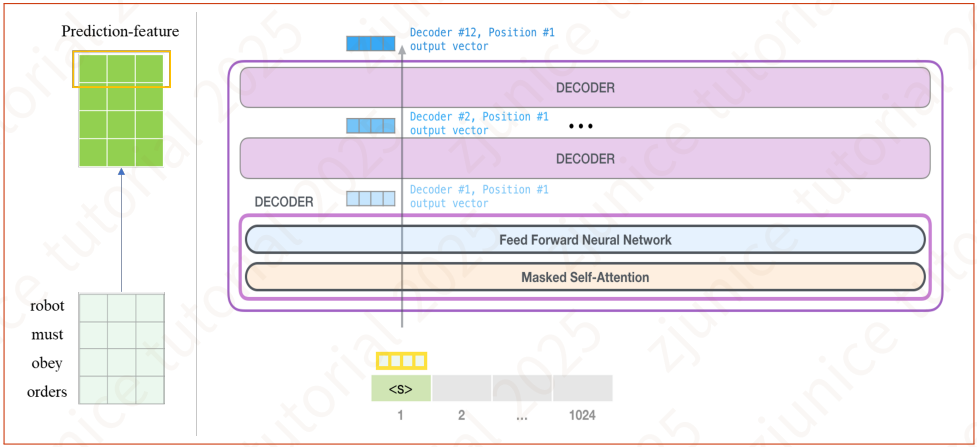
Multi-Head Attention



- $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$; $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$
- $d_{\text{model}} = \text{embedding\_size}$, $d_k = d_v = d_{\text{model}}/h$
- Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

Overview of Architecture

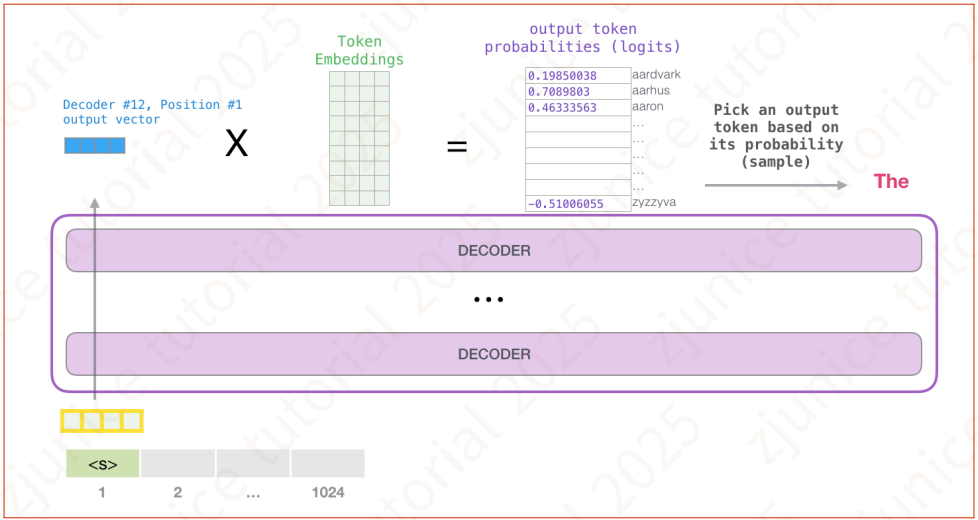
Multiple decoder blocks stacked



- The process is identical in each block
- Each block has its own weights in both self-attention and the NN sublayers

Model Output

Convert the output vector to token



Training and Inference

- Training Dataset Example: English → French

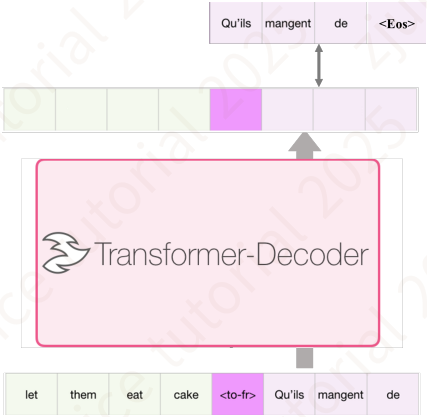


Figure: Training phase

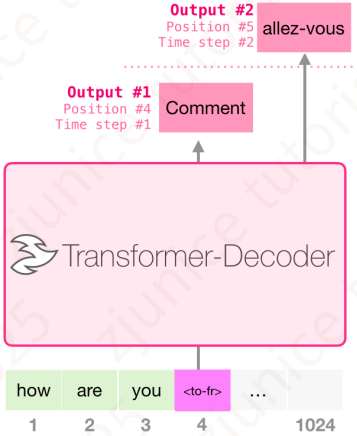


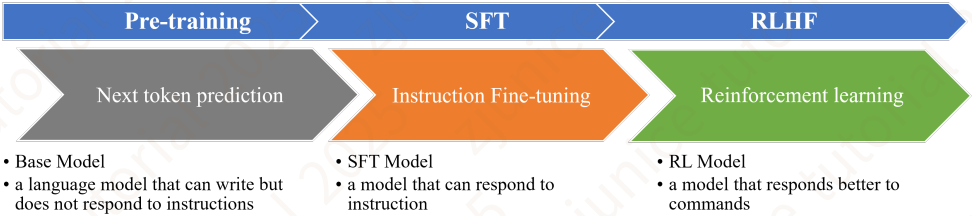
Figure: Inference phase

- Parallel training and autoregressive inference

Training Large Language Models

Overview of LLM Training

GPT Series: Generative **Pre-trained** Transformer

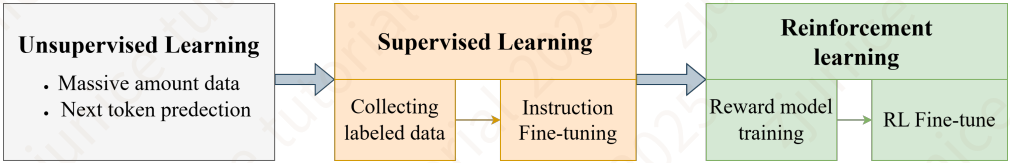


- **Pre-training:** Inculcate knowledge. Training the model to predict the next word using a massive amount of web data. This step results in a base model.
- **SFT:** a supervised fine-tuning step that makes the model more useful in following instructions and answering questions. This step results in an instruction tuned model or a supervised fine-tuning / SFT model.
- **RLHF:** fine-tuning with human feedback is a promising direction for aligning language models with human intent.

Overview of LLM Training

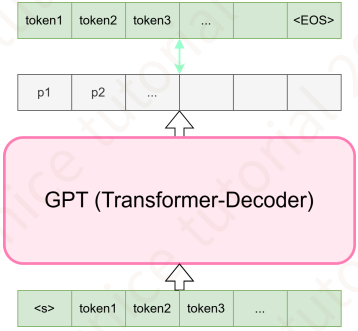
GPT Series: Generative **Pre-trained** Transformer

- Technical route for training LLMs



Pre-training of LLMs[3]

- Labeled data for learning these specific tasks is scarce
- Leverage linguistic information from unlabeled data
- The effective objective in text representation learning – Next token prediction

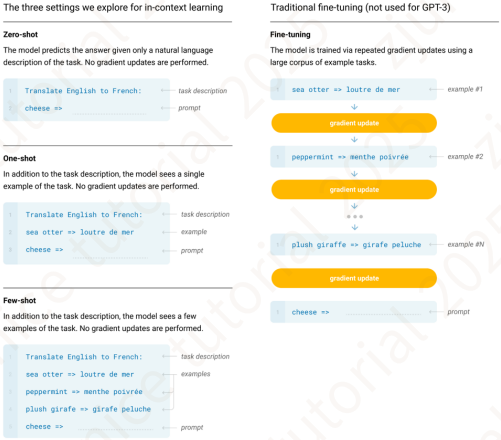


Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

- $h_0 = UW_e + W_p, h_l = \text{Decoder}(h_{l-1}) \forall i \in [1, n], P(u) = \text{softmax}(h_n W_e^T)$
- $U = (u_{-k}, \ldots, u_{-1})$ is context vector; W_e and W_p are token and position embedding matrices.
- $L_1(\mathcal{U}) = \sum_i \log P(u_i \mid u_{i-k}, \ldots, u_{i-1}; \Theta)$

[3] A. Radford et al., "Improving language understanding by generative pre-training,", 2018 .

Pre-training of LLMs[4][5]



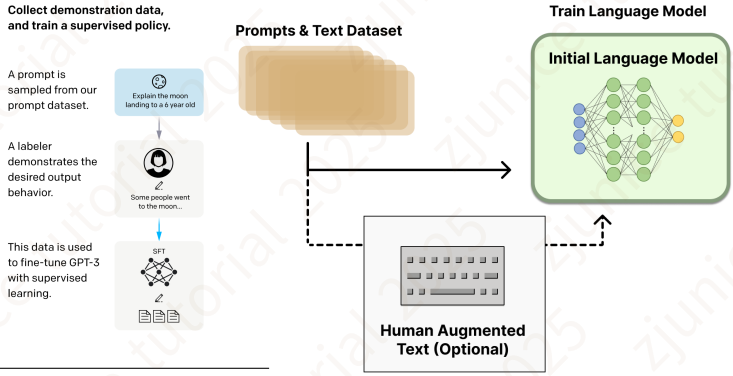
- A single task system $p(output|input)$
- A general system $p(output|input, task)$
- LLM will begin to learn to infer and perform the tasks demonstrated in natural language sequences in order to better predict them, regardless of their method of procurement
- LLM will be performing unsupervised multitask learning

[4] A. Radford et al., "Language models are unsupervised multitask learners,", 2019 .

[5] T. B. Brown et al., *Language models are few-shot learners*, 2020. DOI: 10.48550/arXiv.2005.14165 arXiv: 2005.14165 [cs].

Supervised Fine-tune[6]

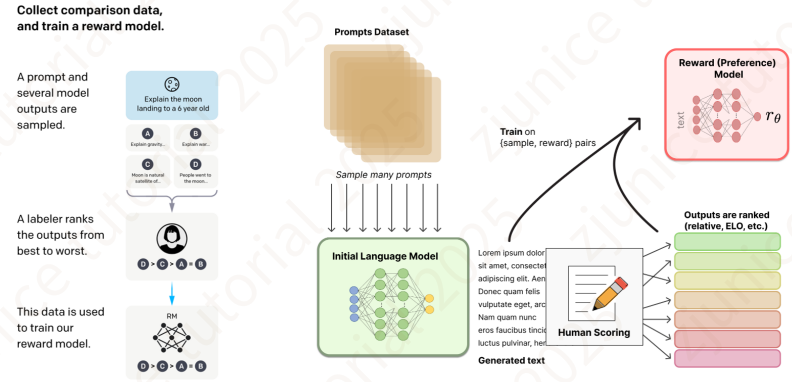
- Making language models bigger does not inherently make them better at following a user's intent
- Needed an initial source of instruction-like prompts to bootstrap the process



[6] L. Ouyang et al., *Training language models to follow instructions with human feedback*, 2022. DOI: 10.48550/arXiv.2203.02155 arXiv: 2203.02155 [cs].

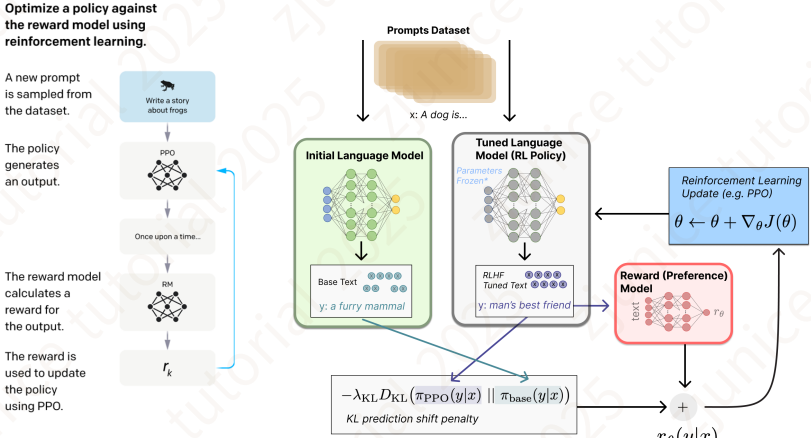
Reward Model Training

- RL enables the agent to adapt to the **environment** make optimal decisions, relying only on feedback and **rewards** from interactions
- Need a reward model can predict the human-preferred output



RL Fine-tune[7]

- Continued enhancement of the language model without labeled data
- Environment: the language model itself

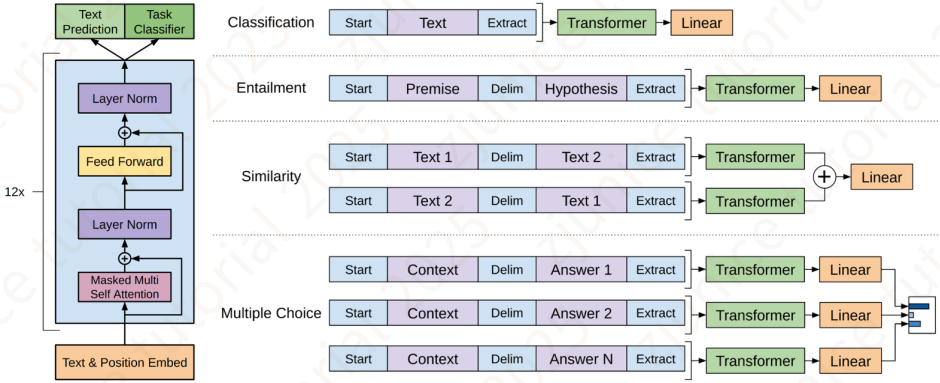


[7] More details can be found in tutorial *"Evolution and Application of Reinforcement Learning (RL) in Large Language Models (LLMs)"*

Weight-based Evolution

Linear Head[8]

The most effective way to transfer these learned representations to the target task.

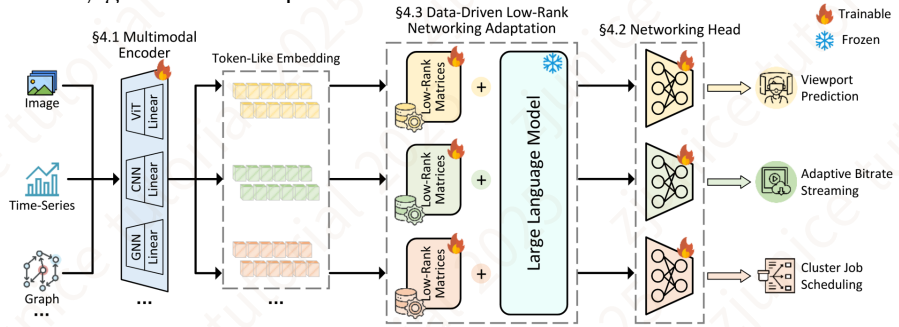


- For a labeled dataset $\mathcal{C} = \{(x^1, \cdots | y), \cdots\}$, adding a linear head W_y
- $L_1(\mathcal{U}) = \sum_i \log P(u_i \mid u_{i-k}, \ldots, u_{i-1}; \Theta)$, $P(y \mid x^1, \ldots, x^m; \Theta) = \text{softmax}(h_l^m W_y)$
- $L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y \mid x^1, \ldots, x^m; \Theta)$, $L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda L_1(\mathcal{C})$

[8] A. Radford et al., "Improving language understanding by generative pre-training,", 2018 .

Fine-tuning with an Added Linear Head

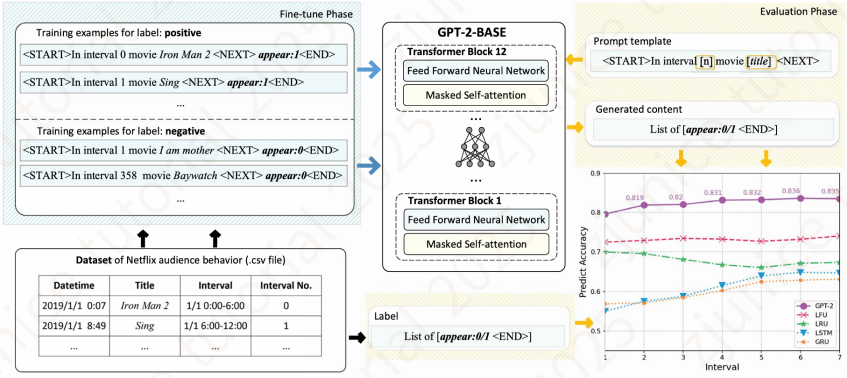
- NetLLM[9]: networking-related use cases - viewport prediction, adaptive bitrate streaming and cluster job scheduling.
- Empowering the LLM to effectively process multimodal data in networking and efficiently generate task-specific answers.



[9] D. Wu et al., "NetLLM: Adapting large language models for networking," in *Proceedings of the ACM SIGCOMM 2024 Conference*, Sydney NSW Australia: ACM, 2024, pp. 661–678, ISBN: 979-8-4007-0614-1. DOI: 10.1145/3651890.3672268

Natural language for specialized domains[10]

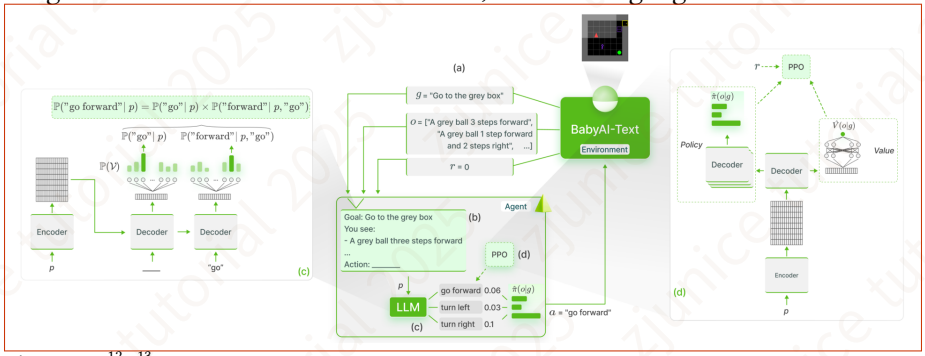
- NetGPT: Synergizing appropriate LLMs at the edge and the cloud based on their computing capacity. Balancing low latency, resource efficiency, and customized outputs.
- Enhancing prompts with location-specific and user-specific context, while cloud models deliver high-quality responses.



[10] Y. Chen et al., "NetGPT: An ai-native network architecture for provisioning beyond personalized generative services," *IEEE Network*, vol. 38, no. 6, pp. 404–413, 2024. DOI: 10.1109/MNET.2024.3376419

Natural Language Actions

- LLM as a policy that is updated as the agent interacts with the environment
- Using an interactive textual environment; Natural language as actions[11]



See also in papers[12] ,[13]

[11] T. Carta et al., "Grounding large language models in interactive environments with online reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause et al., Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 3676–3713.

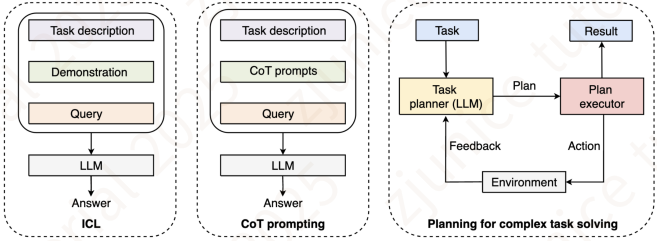
[12] C. Tan et al., *Tool-aided evolutionary LLM for generative policy toward efficient resource management in wireless federated learning*, 2025. arXiv: 2505.11570 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2505.11570

[13] R. Tan et al., *LLM4MAC: An LLM-driven reinforcement learning framework for mac protocol emergence*, 2025. arXiv: 2503.08123 [cs.NI]. [Online]. Available: https://arxiv.org/abs/2503.08123

Text-based Evolution

Prompt Engineering

Prompt engineering: users design various inputs for LLMs to generate desired outputs.
No requirements for extra training, producing output instantly based on user inputs.



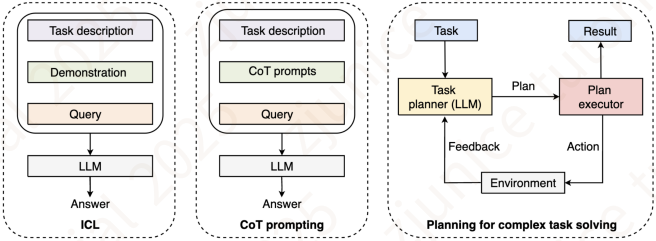
- **In-context learning (ICL)**[14]: Let the LLM recognize and perform new tasks by leveraging contextual information – demonstrations, instructions, etc.
- **Chain-of-thought (CoT) prompting**: Providing reasoning steps, such as "Let's think step by step" Contributing to the generation of reasoning LLMs[15]

[14] Z. Shi et al., "Why larger language models do in-context learning differently?" In *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[15] DeepSeek-AI et al., *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*, 2025. arXiv: 2501.12948 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2501.12948

Methods to Workflows & Agentic

Complex scenarios like mathematical reasoning and multi-hop question answering

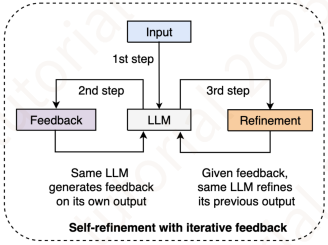


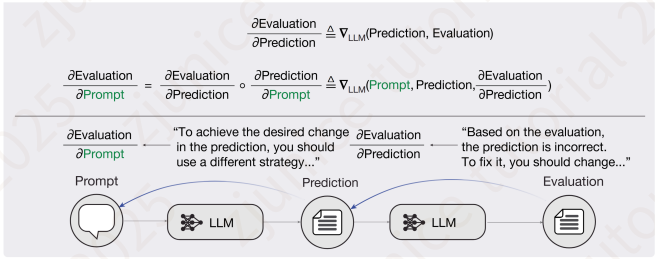
- **Planning for complex task solving**[16]: the task planner, the plan executor, and the environment. Forming workflows to solve problems
- **Agentic system**: LLMs operating as autonomous agents that can make decisions, use external tools, and execute tasks to achieve a goal, rather than just generating text responses.

[16] S. Yao et al., "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems*, A. Oh et al., Eds., vol. 36, Curran Associates, Inc., 2023, pp. 11 809–11 822.

Text Gradients and Back Propagation

Improving their outputs through iterative feedback and refinement





- feedback is actionable, containing concrete steps to further improve the initial outputs. *TextGrad*[17], *Optimizer*[18]
- Prediction = LLM(Prompt + Question)
- Evaluation = LLM(Evaluation Instruction + Prediction)

[17] M. Yuksekgonul et al., "Optimizing generative ai by backpropagating language model feedback," *Nature*, vol. 639, no. 8055, pp. 609–616, Mar. 20, 2025, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-025-08661-4 Accessed: Aug. 7, 2025.

[18] C. Yang et al., "Large language models as optimizers," in *The Twelfth International Conference on Learning Representations*, 2023.

Conclusion

Conclusion

I have talked about

- Causal Tramsformer
  - Self attention and the architecture of the causal transformer
  - The training and inference process
- Large Language Models
  - The training process of the LLM (GPT series)
  - From pre-training to supervised fine-tuning and finally RLHF
- Evolution of the LLM
  - Parameter-based training and applications, including adding linear heads, natural language fine-tuning, natural language reinforcement learning
  - Text-based evolution, including prompt engineering, agentic systems and self-feedback reinforcement of LLMs

Thank you

Networked Intelligence for Comprehensive Efficiency (NICE) Lab
College of Information Science and Electronic Engineering
Zhejiang University
https://nice.rongpeng.info