Quick notes about Vicuna, ChatDoctor, and thoughts on high-quality Chat AI

Vicuna

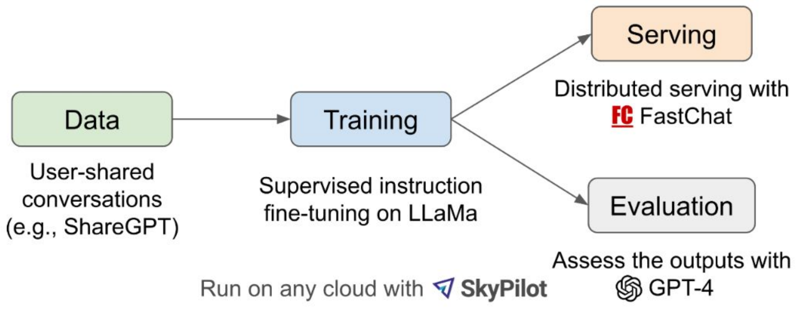
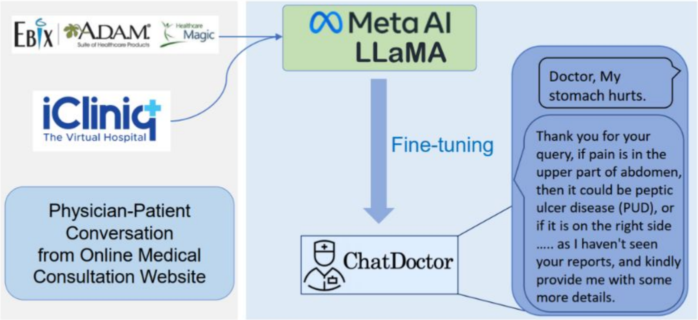


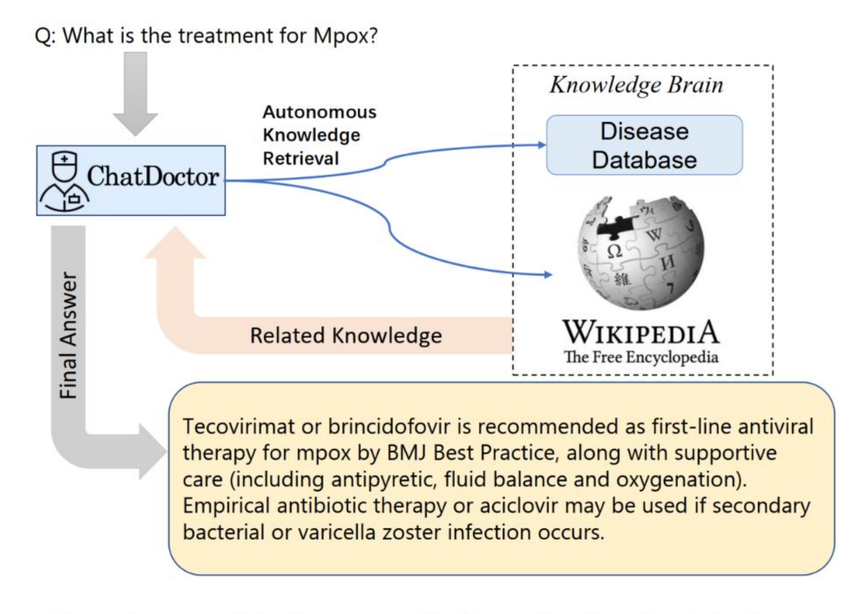
Figure 2. Workflow Overview

Model Name	LLaMA	Alpaca	Vicuna
Dataset	Publicly available datasets (1T token)	Self-instruct from davinci-003 API (52K samples)	User-shared conversations (70K samples)
Training code	N/A	Available	Available
Evaluation metrics	Academic benchmark	Author evaluation	GPT-4 assessment
Training cost (7B)	82K GPU-hours	$500 (data) + $100 (training)	$140 (training)
Training cost (13B)	135K GPU-hours	N/A	$300 (training)

ChatDoctor

1. We designed a framework for fine-tuning large language models in the medical domain.
2. We collected and open-sourced a dataset with 100k patient-physician conversations for fine-tuning the large language model. The dataset contains extensive medical expertise for the medical application of LLMs.
3. Based on the external knowledge brain, we proposed an autonomous ChatDoctor model with online analysis ability of novel expertise.





Some personal thoughts on high-quality Chat AI

- Distill from “oracle AI” (e.g. Alpaca) seems popular to bootstrap
- DeepSpeed or similar technique to further push the limit of hardware
- Using AI to critique itself (e.g. constitutional AI) is a powerful idea
    - Self-instruct in Alpaca is just a first step
- We might need high quality user data for better quality (Vicuna vs Alpaca)
    - Sometime free, e.g. ShareGPT.com by Vicuna
    - But sometimes, at the cost of more labor cost
- SFT (Supervise finetune) vs RLHF (reinforcement learning from human feedback)
    - RLHF could be powerful to make “good” models to be “great”!
    - But it is expensive and hard to train train RLHF pipelines (even with LoRA)
- The under-estimated multi-turn conversation for a smart chat AI
    - Often single turn data is used for finetune
    - The context length (e.g. 512 tokens) is a bottleneck

Why RLHF matters according to John Schulman?

[lack reference, a chinese summary of his talk to Berkeley in April 2023, but I could not find it now]

- SFT is too sensitive to different variations of same meaning, but Reward model in RLHF is not sensitive, aligns with humans
- SFT only provides the positive signal (d*o what I told you do*), RL provides the negative signal too (*learn from the sample with higher reward, and walk away form the sample with low reward*)
- Training data in SFT may bring in new knowledge that is not in pretrained model, so SFT will tend more to answer question that it does not know, while RL will encourage model to say “I don’t know”.

Some recommended readings on training/inference efficiency

- Data/Model/Tensor parallelism intro by HuggingFace
- The Annotated Transformer by Harvard NLP
- Multi-query Attention by Google
- FlashAttention by Stanford
- Efficiently Scaling Transformer Inference by Google