Nome: Fernando Buligon Antunes **Data:**09/04/2025

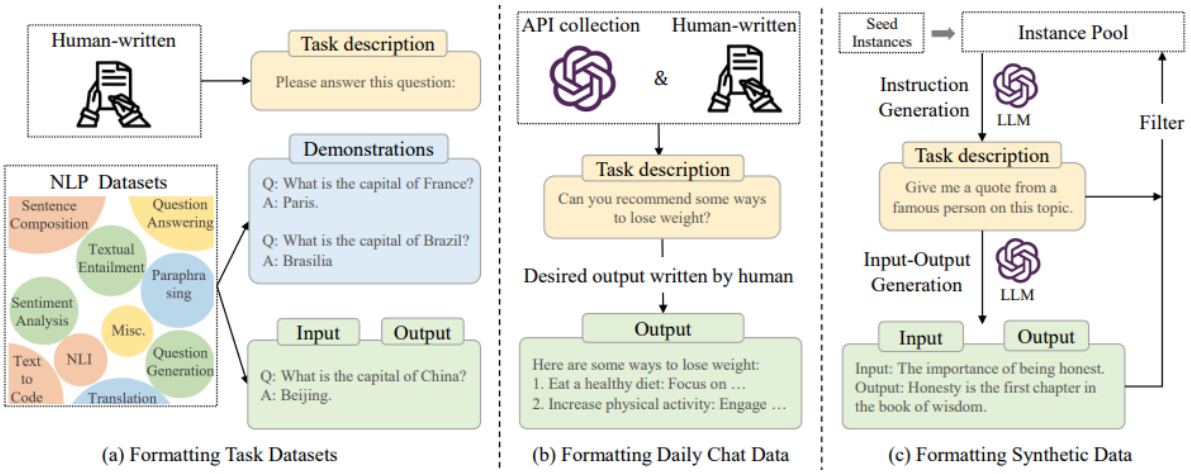
Large Language Models (Tópico 5)

Este tópico aborda as duas principais maneiras para fazer com um modelo LLM pré treinado generalizado seja capaz de se adaptar para que consiga se especializar em algum ponto dependendo do objetivo do modelo.

Instruction tuning, primeiro temos que escolher ou montar uma instância de instruções formatadas para depois aplicá las na LLM para que ela consiga aprender de uma maneira supervisionada, essa técnica foi amplamente utilizada em grandes LLMs, como por exemplo, GPT-4.

Depois são discutidos três diferentes métodos para construir instâncias formatadas:

- **Task Datasets:** Antes do instruction tuning ser proposto, alguns estudos anteriores fizeram a coleta de várias instâncias de algumas tarefas tradicionais de NLP, como tradução, resumo, classificação, e etc. Reunindo todos esses dados, acabou sendo criado uma base de dados generalizada para treinamento supervisionado.
- **Daily Chat Data:** Apesar da maioria das instâncias serem formatadas com instruções, elas são retiradas principalmente de repositórios públicos de NLP, então acontece de faltar diversidade de instruções ou não se enquadrar direito com as necessidades humanas. Então foi pensado em fazer uso das perguntas feitas por pessoas reais da API da OpenAI como descrição das tarefas, resolvendo o problema anterior, também foram contratadas pessoas para descrever as respostas esperadas.
- **Synthetic Data:** Uma das soluções para diminuir o trabalho manual, economizar tempo e dinheiro, são os dados sintéticos, então essa técnica foi aplicada em algumas instâncias em LLMs, então eram geradas as instâncias, depois tinham que passar por um filtro de qualidade, e depois davam o resultado, o único problema é que fazendo dessa forma, as instâncias geradas podem sofrer com a falta de diversidade e de complexidade.



Na imagem acima, é possível visualizar as três técnicas.

As instâncias afetam diretamente os resultados finais do modelo, então é importante fazer algumas coisas para que esse impacto seja positivo, como por exemplo, aumentar o número de tarefas, melhorar a diversidade, fazer instruções construídas de maneiras mais eficazes e formatar de maneira intuitiva. Fazendo isso, vai melhorar o desempenho dos modelos, aumentando a capacidade deles em tarefas generalizadas ou específicas.

Alignment Tuning, LLMs mostraram habilidades incríveis em tarefas de NLP gerais, mas algumas vezes o resultado não é o esperado, podendo trazer informações inadequadas, como informações pessoais, criminosas, falsas ou até mesmo criadas. Para evitar essas situações, são feitos ajustes por humanos com alguns critérios como:

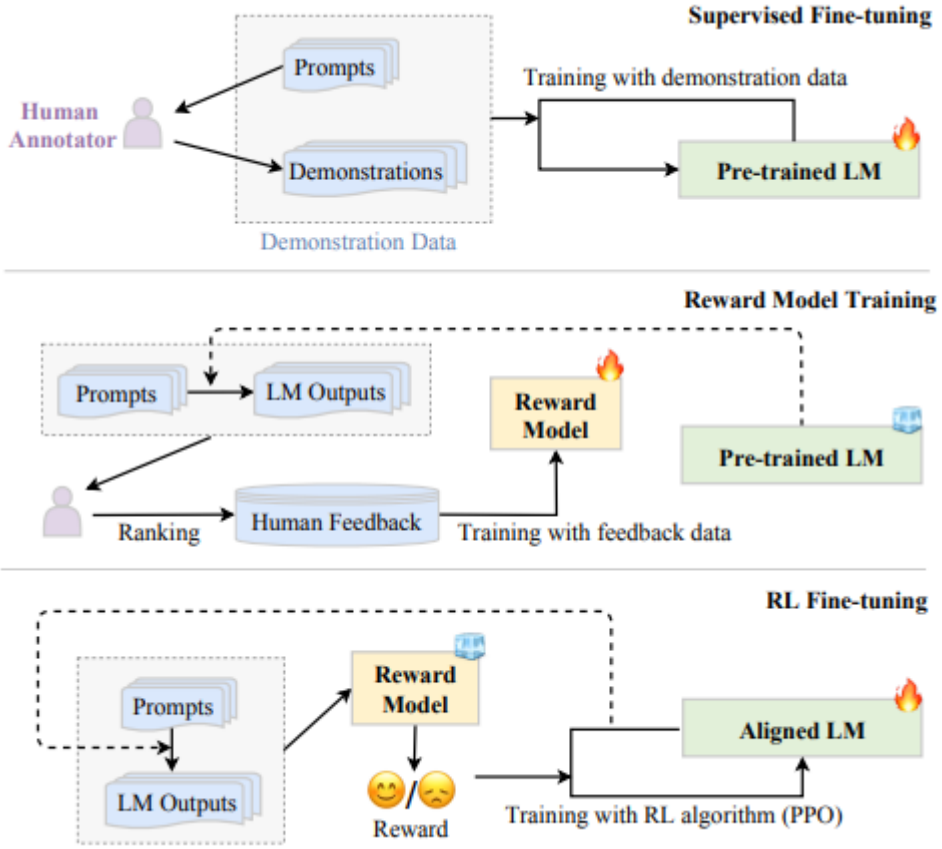
- **Helpfulness**: o modelo deve demonstrar uma tentativa clara de ajuda para o usuário, resolvendo o que foi pedido da maneira mais eficiente o possível;
- **Honesty**: o modelo deve sempre responder com dados consistentes, ao invés de responder com desinformações ou dados proibidos;
- **Harmlessness**: o modelo não pode tratar os usuários de maneira ofensiva e também deve ser capaz de detectar malícias nos prompts, respondendo de maneira educada.

Feedback humano é altamente valioso para uma LLM, por isso são criados times para essa área específica, em que humanos capacitados são contratados para fazer anotações, gerando os feedbacks, existem três maneiras de fazer essa coleta:

- **Ranking-based approach**: inicialmente as pessoas que faziam as anotações analisavam as respostas, selecionavam a melhor, fazendo com que as outras fossem descartadas, mas o problema nisso é que havia um desperdício de respostas, depois foi montando um ranking, em que cada resposta recebe uma pontuação, fazendo com que haja um aproveitamento melhor e que o modelo seja capaz de priorizar outputs mais confiáveis;

- **Question-based approach**: os anotadores respondiam perguntas elaboradas por pesquisadores, permitindo feedbacks mais detalhados com base nos critérios de alinhamento e restrições adicionais;
- **Rule-based approach**: aplicam regras para avaliar se as respostas geradas pelo modelo se enquadram nos critérios de utilidade, corretude e segurança. São gerados dois tipos de feedbacks, o primeiro é a resposta preferida em comparação com as outras respostas geradas, e o segundo é violação de regra, retornando uma pontuação e o grau de inadequação.

RLHF (Reinforcement Learning from Human Feedback), técnica introduzida para alinhar as LLMs com os valores humanos, fazendo uso de algoritmos de aprendizado reforçado para adaptar as LLMs para o feedback humano aprendendo sobre um modelo de recompensas. Então é composto basicamente por três componentes chaves, um modelo LM pré treinado, um modelo de recompensa aprendendo com o feedback humano e um algoritmo de RL.



Na imagem acima é possível ver o fluxo de trabalho.

Apesar do RLHF mostrar ser capaz de atingir ótimos resultados em alinhar os comportamentos das LLMs com base nos princípios humanos, essa forma de trabalho acaba possuindo algumas limitações, por essa causa, foi introduzido o non-RL alignment, ou em português, alinhamento sem aprendizado reforçado.

O alinhamento sem RL propõe ajustar diretamente as LLMs com aprendizado supervisionado sobre uma base de dados de alinhamento com dados de altíssima qualidade. A ideia principal é que essa base já possua respostas seguras e regras de segurança embutidas, fazendo com que o modelo aprenda a maneira correta de responder por ali. Por isso é de extrema importância a montagem e organização dessa base de dados, o comportamento do modelo varia completamente dependendo do dados.

Um dos grandes problemas é em relação ao uso de memória, devido ao grande número de parâmetros, acaba requisitando um poder de processamento maior.

Para resolver esse problema, foram desenvolvidas algumas técnicas:

- **Adapter Tuning**: incorpora pequenos módulos de redes neurais nos modelos transformers, aí durante o fine tuning esses módulos chamados de “adapters” são otimizados dependendo da tarefa designada, enquanto os originais são congelados;
- **Prefix Tuning**: adiciona uma sequência de prefixos, formando vetores contínuos treináveis para cada camada de transformer, otimizados por uma função MLP;
- **Prompt Tuning**: adiciona vetores de prompts treináveis na camada de entrada;
- **LoRA (Low-Rank Adaptation)**: congela as matrizes originais e treina matrizes de baixa dimensão.