Transcrição

Olá, hoje nossa análise entra em um território fundamental para quem usa inteligência artificial. Como é que a gente consegue extrair o máximo dos grandes modelos de linguagem, sabe? Os LLMs, tipo GPT-4, Claude? Exato. Porque não é só jogar a pergunta lá, né? Tem técnica por trás.
A gente deu uma olhada em guias práticos, artigos, comparando tecnologias, discussões de especialistas de várias fontes, de O, Data Science Academy, PM3, DNX Brasil, até o pessoal no Reddit comentando. Sim, sim, muita coisa boa. E o nosso objetivo aqui é meio que desvendar três abordagens que sempre aparecem:

1. Engenharia de Prompt
2. Fine Tuning
3. Retrieval Augmented Generation (RAG)

O que é cada uma afinal? E quando usar? Pois é, muitas vezes o desafio não está nem na capacidade do modelo em si, mas em como a gente pede as coisas ou adapta o uso dele, né? Essas técnicas são, olha, essenciais para quem quer otimizar processos, automatizar tarefas ou simplesmente ter respostas melhores, mais úteis. Para tudo, né? Marketing, desenvolvimento, análise de dados, até atendimento. Exato.
Então, vamos entender o que cada uma dessas abordagens realmente oferece com base no que a gente viu nas suas fontes todas. Beleza. Começando pela talvez mais direta, Engenharia de Prompt. Pelo que eu vi, é basicamente a arte de criar o pedido, o prompt, de um jeito eficaz. Isso. E a regra de ouro que aparece em vários guias é clareza. Tem que ser claro. Como se estivesse explicando para uma pessoa. Uma estrutura útil que apareceu bastante é a PROMT:

- Definir a persona
- O roteiro
- O objetivo
- Modelo de saída
- Dar panorama, contexto, né?
-

E tá pronto para testar, para iterar. É uma boa base. E a engenharia de prompt vai além disso, né? As fontes que a gente analisou detalham umas técnicas bem interessantes. Tem o Zero Shot, que é pedir direto, sem dar exemplo nenhum. Ah, o mais básico. Isso. Aí tem o Fill Shot, que você dá tipo uns dois ou três exemplos para guiar o modelo. E uma que achei bem legal, o Chain of Thought, o COT. Ah, essa eu vi mencionada. Faz o modelo pensar alto. Exatamente. Ele explica o passo a passo do raciocínio. Alguns artigos técnicos comparativos

mostraram que isso melhora bastante em tarefas de lógica, matemática, coisas mais complexas.
E tem outras ainda, tipo:

- Decomposição, quebrar um problema grande em partes menores
- Ensembling, que é usar vários prompts e combinar as respostas
- Até autocrítica, onde o modelo meio que avalia o que ele mesmo gerou. Nossa, quanta coisa só para fazer a pergunta certa. Pois é. E o pessoal nas discussões menciona muito ferramentas como LinkedIn, Prompt Layer. Ah, LinkedIn aparece bastante. Ajuda a conectar tudo, né? Isso. É um framework para encadear chamadas, integrar com outras APIs. Facilita criar aplicações mais robustas. E o Prompt Layer ajuda a organizar e testar os prompts. Legal. A vantagem clara, então, é ser mais barato e rápido de implementar. Sim, custo baixo, acesso rápido. A desvantagem? Bom, você depende do que o modelo já aprendeu no treino original, né? E às vezes a resposta pode não ser tão consistente. Certo. Faz sentido.

Agora, subindo um degrau na complexidade. Fine tuning. Isso soa mais pesado. Tipo, mandar o modelo de volta para a escola. É uma boa analogia. É quase isso. Aqui a gente ajusta os parâmetros internos do modelo, os pesos, usando um conjunto de dados específico para uma tarefa.

A gente viu exemplos, acho que foi num fórum do Reddit, de equipes que criaram chatbots de suporte com a voz e o tom certinho da marca. Uma coisa que só o fine tuning profundo permite com essa precisão toda. Ah, entendi. Para customizar mesmo? Exato. O resultado é alta customização, performance excelente em tarefas bem de nicho, mas... Tem um mas, claro. Sim.

O custo é bem maior. Exige poder computacional, que não é barato, tempo, dados de treinamento de altíssima qualidade, e isso é crucial. E, claro, gente que entenda de machine learning para fazer direito.

E tem aquele risco, né? Li algo sobre isso. O modelo ficar tão especializado que meio que esquece o resto. Fica menos geral. Sim, você tocou num ponto importante. É o que chamam às vezes de catastrophic forgetting. O modelo fica ótimo naquilo, mas pode perder um pouco da versatilidade.

Por isso, o fine tuningé mais recomendado para tarefas, assim, bem definidas, que vão durar bastante tempo e ter um volume grande de uso. Aí a especialização compensa o investimento e essa possível perda. para atualizar, novo treino. Entendido. É um investimento maior para casos específicos.

E aí chegamos no RAG, geração aumentada por recuperação. Esse parece um meio termo interessante. Como é que ele funciona na prática? O RAG é bem engenhoso, eu diria. Ele junta o poder de gerar texto do LLM com uma busca tipo em tempo real numa base de conhecimento externa.
Espera, como assim? Quando chega a pergunta, o sistema RAG primeiro vai lá e busca documentos ou dados que sejam relevantes para aquela pergunta. Pode ser na documentação interna da empresa, artigos de notícias recentes, base de produtos.

Ah, tá. E só depois ele manda para o LLM a pergunta original junto com essa informação que ele recuperou. E o LLM usa isso para responder? Exatamente. Um guia prático da Data Science Academy que a gente viu destacava justamente isso. Como o RAG é bom para lidar com informação que muda toda hora.

Notícias, documentação técnica que atualiza sempre. Isso reduz muito aquelas respostas erradas porque o modelo está desatualizado ou as tais alucinações, né? E o legal é que o modelo original fica intacto.

Puxa, isso é muito útil. Mas essa busca externa tem que ser boa, né? Se ela falhar ou trouxer lixo, a resposta do LLM vai ser ruim também, certo? Com certeza. Esse é o calcanhar de Aquiles do RAG, digamos assim. A qualidade da resposta final depende totalmente da qualidade do que foi recuperado.

Busca ruim, resposta ruim. É batata.
Além disso, é mais complexo de montar do que só usar um prompt, né? Tem que integrar a busca com o LLM e ainda tem o limite de quanta informação cabe na janela de contexto do modelo.

Ah, tem isso também? Tem. Ferramentas como o RAGAS, que apareceram em algumas análises técnicas, ajudam a medir o quão bem essa dupla, busca e geração, está funcionando junta.

Ok, então, tentando simplificar a coisa toda. Engenharia de prompt é ajustar a pergunta que a gente faz. Isso. Fine tuning é ajustar o modelo em si lá dentro. Perfeito. E RAG é dar informação extra para o modelo, mas só na hora da resposta. Exatamente essa a ideia.
A escolha vai depender muito do problema que você quer resolver. Precisa de agilidade, baixo custo, flexibilidade, provavelmente em engenharia de prompt. Certo. Precisa de uma expertise muito profunda e consistente numa tarefa específica e tem recursos para investir. Aí, talvez, seja fine tuning.

Agora, precisa que as respostas usem dados super recentes ou informações muito específicas que o modelo não tem como saber, aí o RAG brilha.
E elas podem trabalhar juntas, imagina? Podem e devem. Vimos discussões sobre isso, um modelo que já passou por fine tuning pode muito bem usar RAG para buscar dados atuais e, claro, sempre se beneficia de um bom prompt. Não são excludentes.

Claro que, como algumas discussões de especialistas apontaram, ainda temos desafios gerais, né? Viés nos dados, o risco de gerar conteúdo problemático e aquela dificuldade de entender como o modelo chega na resposta, a tal da caixa preta. Fica bem claro que saber interagir e saber adaptar esses modelos de IA está virando uma habilidade nossa cada vez mais valiosa. Conhecer essas abordagens abre um leque enorme para usar as ferramentas de um jeito mais estratégico, mais poderoso, né? E responsável também.
Exatamente. E isso até me faz pensar, sabe? À medida que a gente fica melhor nessas técnicas, fazer prompts melhores, alimentar com dados específicos via fine tuning, dar contexto externo com RAG, será que a própria noção de inteligência do modelo não muda um pouco de lugar?

Hum, como assim? Será que a verdadeira habilidade está menos no modelo isolado, naquela coisa meio mágica, e mais na nossa capacidade de guiar, de complementar, de potencializar ele com as ferramentas e os dados certos? É algo para a gente ficar pensando, né?