Tech Challenge 3 - Fine-Tuning de de um modelo LLM

Leandro Cordeiro David - RM356103

Vídeo Descritivo do Trabalho

O vídeo apresentando a solução desenvolvida está disponível no youtube no link:
https://www.youtube.com/watch?v=GHSYdv84v10

Código Fonte

O código fonte do python notebook desenvolvido está disponível no github em:
https://github.com/leandrodvd/fiap-postech-ia4devs/tree/main/techchallenge3_finetuning

Introdução

O presente trabalho faz parte do Tech Challenge 3 da pós-graduação em Inteligência Artificial. O objetivo principal é realizar o fine-tuning de um modelo de linguagem (LLM) utilizando o dataset "The AmazonTitles-1.3MM", que contém informações sobre títulos de produtos e suas descrições extraídas da Amazon. A tarefa é treinar o modelo para responder perguntas sobre produtos com base nas descrições presentes no dataset, abordando o desafio de integrar conhecimento específico e privado ao modelo, que inicialmente não tem acesso a essas informações.

Seleção e Preparação do Dataset

O dataset utilizado, "The AmazonTitles-1.3MM", contém consultas textuais de usuários e as descrições de produtos correspondentes. O arquivo principal utilizado foi o `trn.json`, no qual as colunas de interesse são `title` (título do produto) e `content` (descrição do produto). Para otimizar o processo de treinamento, foi criada uma versão reduzida do dataset, chamada `trn_100k.json`, contendo apenas as primeiras 100.000 linhas do arquivo original.

Após a leitura dos dados, foi realizada uma análise inicial para identificar e remover linhas com conteúdo vazio na coluna `content`, essencial para garantir que o treinamento fosse realizado apenas com informações válidas. A limpeza do dataset foi feita removendo strings vazias, resultando em um conjunto de dados finalizado e sem linhas nulas.

Em seguida, para o treinamento do modelo, foi utilizado um subconjunto de 500 linhas, selecionadas para minimizar o custo e tempo de execução, uma vez que cada interação com a API do OpenAI gera custos de operação.

Geração de Prompts

Para treinar o modelo, foram geradas perguntas com base nos títulos dos produtos presentes no dataset. O objetivo era criar prompts variáveis que simulassem perguntas feitas por usuários reais. Foram utilizadas cinco variações de perguntas, como:

- "Please describe the product [TITLE] from Amazon."
- "What can you tell me about the product [TITLE] from Amazon?"
- "Tell me about [TITLE] from Amazon."
- "What is the product [TITLE] from Amazon."
- "Give me a description of [TITLE] from Amazon."

Essas perguntas foram geradas aleatoriamente para cada produto do dataset, criando uma nova coluna `question` no DataFrame contendo as perguntas correspondentes a cada título.

Chamada ao Foundation Model

Antes de realizar o fine-tuning, foi feita uma chamada ao modelo de LLM da OpenAI, GPT-4o-mini, sem qualquer ajuste, para estabelecer uma base de comparação. A função `chat.completions.create` do sdk python da OpenAI foi utilizada. O resultado obtido foi uma resposta padrão do modelo pré-treinado, que, em sua maioria, retornava respostas genéricas ou mensagens afirmando não ter acesso a informações específicas sobre os produtos da Amazon.

Essa etapa inicial permitiu verificar que o modelo, antes do treinamento, não possuía conhecimento sobre os produtos contidos no dataset, reforçando a necessidade do fine-tuning.

Processo de Fine-Tuning

Para o fine-tuning, foi necessário estruturar o dataset no formato exigido pela API da OpenAI. Cada linha do dataset foi transformada em um par de mensagem e resposta no seguinte formato:

- **User Prompt (Pergunta do Usuário):** A pergunta gerada usando o título do produto.
- **Expected Response (Resposta Esperada):** A resposta contendo as informações da coluna `content`, onde o modelo é instruído a responder "This is what I know about [Título do Produto] from Amazon:" seguido da descrição do produto.

Após a preparação dos prompts, o arquivo de treinamento foi salvo no formato JSON (prompts.jsonl) e enviado para a API da OpenAI, utilizando a função `openai.File.create`. A função `fine_tuning.jobs.create` foi utilizada para iniciar um job de treinamento, seguida

da função `fine_tuning.jobs.retrieve` para acompanhar o status do job de fine tuning e obter o identificador do modelo treinado ao final do processo. O treinamento foi executado no modelo GPT-4o-mini mais recente, e foi monitorado até sua conclusão.

Execução do Fine-Tuning

O processo de treinamento durou cerca de 20 minutos. Após a finalização, o modelo ajustado foi identificado por um novo ID gerado pela OpenAI. O nome do modelo treinado foi retornado pela API, e este foi utilizado para realizar novos testes, comparando as respostas antes e após o fine-tuning.

Testes com o Modelo Treinado

Para avaliar o impacto do fine-tuning, as mesmas 500 perguntas utilizadas anteriormente foram submetidas ao modelo ajustado. As respostas retornadas apresentaram uma diferença significativa: o modelo agora era capaz de fornecer respostas baseadas no conteúdo do dataset, especialmente para produtos cujas informações estavam disponíveis no arquivo de treino.

Em casos onde o modelo foi treinado corretamente, ele respondeu conforme o padrão treinado, utilizando o formato: "This is what I know about [Título do Produto] from Amazon...". Entretanto, houve alguns casos onde o modelo ainda retornou respostas incompletas ou informações públicas, indicando que há espaço para melhorias no treinamento.

Considerações Finais

O fine-tuning do modelo GPT-4o-mini utilizando o dataset "The AmazonTitles-1.3MM" foi concluído com sucesso, demonstrando que o modelo treinado passou a incorporar informações específicas dos produtos presentes no dataset. Embora as respostas não tenham sido perfeitas em todos os casos, o modelo mostrou uma clara melhora ao seguir o formato e as informações fornecidas no processo de treinamento.

Há espaço para melhorias, como a expansão do conjunto de perguntas ou ajustes finos no processo de geração de respostas, mas o trabalho cumpriu seu objetivo principal de exercitar a técnica de fine-tuning em um modelo de linguagem.