

Fine-tuning com LLaMa-v2

Aerty Santos; Eduardo Oliveira

Abstract—Neste artigo, são analisados os avanços do LLaMa 2 no Processamento de Linguagem Natural. O LLaMa 2, um modelo open-source revolucionário treinado com até 70 bilhões de parâmetros, impactando pesquisadores e desenvolvedores em todo o mundo. São discutidas as inovações do modelo, que incluem técnicas como engenharia imediata, fine-tuning (ajuste fino) e geração aumentada de recuperação e incorporação(RAG).

O estudo concentrou-se no ajuste fino do LLaMa 2 por meio da utilização da Wikipedia em Português, visando aprimorar sua capacidade de compreensão e geração de texto neste idioma. Os resultados foram comparados antes e depois da utilização desta técnica, assim como com utilizando a estrutura RAG, empregando o supercomputador Atena. Esse estudo capacita profissionais de Processamento de Linguagem Natural a aplicar estrategicamente essas técnicas, impulsionando avanços significativos na compreensão e na geração de linguagem natural.

Index Terms—GPT, LLaMa, LLM, PNL, Fine-tuning.

I. INTRODUÇÃO

A constante evolução no campo do Processamento de Linguagem Natural (PNL) tem sido marcada por grandes avanços na criação de Modelos de Linguagem Grande (LLMs). Nesse contexto, em julho de 2023, a Meta AI se destaca com a introdução do LLaMa 2, um modelo de GPT, disponibilizado gratuitamente, uma conquista revolucionária que elevou o patamar da comunidade de PNL. Com uma variedade impressionante de modelos pré-treinados e ajustados, que variam de 7 bilhões a surpreendentes 70 bilhões de parâmetros, o LLaMa 2 não apenas amplia o legado do seu predecessor, o LLaMa 1, mas também oferece aprimoramentos notáveis que têm impactado profundamente pesquisadores, desenvolvedores e entusiastas da inteligência artificial em todo o mundo[1]. Este artigo explora as inovações, melhorias e potenciais impactos do LLaMa 2 na vanguarda do PNL, destacando sua versatilidade, adaptabilidade e contribuições para o desenvolvimento contínuo do entendimento e geração de linguagem natural.

Com a disponibilidade do modelo, diversos setores passaram a fazer uso, mas mesmo os 70 bilhões de parâmetros de treinamento, não faz dele um especialista em cada área, por diversas vezes não respondendo da maneira apropriada. Neste contexto, o ajuste fino emerge como uma técnica fundamental na otimização de modelos pré-treinados para tarefas específicas. A engenharia imediata, por sua vez, focaliza na adaptação ágil e personalização de modelos para cenários distintos. A geração aumentada de recuperação (RAG) representa um avanço significativo, combinando habilmente técnicas de geração de texto com capacidades de busca e recuperação de informações. Por fim, a incorporação simplifica a integração de conhecimento específico em modelos de linguagem, permitindo uma compreensão mais profunda e contextualizada. Ao desmistificar esses conceitos, estamos capacitando os profissionais da área a explorar e aplicar essas técnicas de forma mais informada e estratégica.

Este artigo investiga o processo de refinamento (fine-tuning) do modelo de linguagem GPT (Generative Pre-trained Transformer) LLaMa2, utilizando a Wikipedia em Português. A pesquisa foi conduzida utilizando a capacidade computacional do supercomputador Atena, que atualmente possui 3 unidades da GPU NVIDIA A100 de 80GB, e somando mais de 500GB de memória RAM. O que permitiu a comparação dos resultados sem alterações no modelo, com os resultados das duas técnicas: após o fine-tuning, bem como utilizando o resultado outra estrutura de recuperação de respostas, conhecida como RAG. O objetivo central é ampliar significativamente a capacidade de compreensão e geração de texto do modelo na língua portuguesa.

O artigo foi dividido em algumas seções, sendo a 2, uma breve revisão bibliográfica, a seção 3 foca na metodologia adotada, por fim, a seção 4 reunirá os resultados e conclusões obtidas.

II. REVISÃO BIBLIOGRÁFICA

O fine-tuning do modelo LLaMa 2 foi conduzido com o intuito de personalizar e adaptar o modelo a um contexto específico dentro do campo do Processamento de Linguagem Natural, aperfeiçoando seu conhecimento na língua portuguesa. Os aprimoramentos introduzidos, como um treinamento expandido com mais tokens e uma extensão de contexto mais ampla, foram direcionados para fortalecer a compreensão e a geração de linguagem em cenários particulares, garantindo que o modelo pudesse capturar nuances e detalhes específicos desse contexto em particular.

Foram realizados testes com os modelos LLaMa 2 de tamanhos 7B, 13B e 70B. Os testes foram realizados primeiro em CPU e depois em GPU. Para os testes, optou-se pelo LLaMa 13b devido ao seu equilíbrio entre tamanho e capacidade, aliado à sua maior precisão nas tarefas de tradução de idiomas. Enquanto o LLaMa 7b se destaca por sua velocidade e tamanho reduzido, pode não atingir a precisão necessária para traduzir idiomas com estruturas gramaticais complexas, como o português. Por outro lado, o LLaMa 70b, sendo o maior e mais poderoso dos modelos, oferece desempenho superior em tarefas complexas, mas requer hardware mais robusto e pode ser mais lento. Logo, o LLaMa 13b se revela como a opção mais indicada para os testes, proporcionando um equilíbrio entre tamanho, capacidade e precisão.

A. Fine-tuning

O ajuste fino de instruções é uma prática frequente empregada para adaptar um LLM básico a um cenário de utilização específico. Uma analogia que pode ajudar a entender o fine-tuning é imaginar que há um robô que aprende a reconhecer diferentes tipos de frutas, como maçãs, bananas e laranjas.

Inicialmente, o robô aprende com muitos exemplos gerais de frutas, mas pode cometer alguns erros, como confundir uma maçã com uma laranja. Fine-tuning seria como ajustar o robô para se tornar melhor em reconhecer uma fruta específica. Por exemplo, se o robô confunde maçãs com laranjas, você pode mostrar mais exemplos de maçãs para que ele se torne mais preciso. Isso é como ajustar o robô para afinar suas habilidades e se tornar especialista em reconhecer maçãs corretamente. Gerando um novo modelo, agora personalizado e especializado em detectar maçãs. Para que o fine-tuning aconteça é necessário expor o modelo pré-treinado a informação que deve seguir um padrão, o qual possui:

- **Instrução:** Elas ajudam a direcionar o modelo para se adaptar a uma tarefa específica, indicando o comportamento desejado. Essas instruções são cruciais para personalizar o modelo para atender a requisitos específicos.
- **Entrada:** Elas incluem exemplos da tarefa ou domínio específicos que você deseja que o modelo aprenda.
- **Resposta:** As saídas esperadas são as respostas que você deseja que o modelo produza com base nas entradas fornecidas durante o treinamento. São os "rótulos" associados às entradas.

Ao ser exposto a diversos exemplos de perguntas e respostas, o modelo aprende, logo, neste artigo foram utilizados uma instrução e diversos pares de entrada-saída provenientes da Wikipédia Brasil para aprimorar o modelo a gerar a saída correspondente à entrada fornecida como um prompt, como exemplificado a seguir [2]:

Instrução: Analise a pergunta abaixo e forneça...

Entrada: Qual a capital do Brasil?

Resposta: Brasília

B. RAG

De forma geral os grandes modelos de linguagem pré-treinados têm demonstrado a capacidade de armazenar conhecimento factual em seus parâmetros, alcançando resultados de ponta quando refinados para tarefas de processamento de linguagem natural. Contudo, sua habilidade de acessar e manipular o conhecimento de maneira precisa ainda é limitada, resultando em um desempenho inferior em tarefas intensivas em conhecimento quando comparados a arquiteturas específicas para a tarefa. Além disso, a atribuição de proveniência às suas decisões e a atualização do conhecimento sobre o mundo permanecem como problemas de pesquisa em aberto [2].

O RAG foi projetado para aumentar as capacidades do LLM, recuperando informações relevantes de fontes de conhecimento antes de gerar uma resposta e é excelente na incorporação de conhecimento externo. Ou seja, combina os pontos fortes das abordagens baseadas em recuperação e em generativas.

Antes de desenvolver um sistema RAG, é importante se familiarizar com as ferramentas que tornam esse processo possível. Cada ferramenta desempenha um papel específico, garantindo o funcionamento eficiente e eficaz do sistema RAG.

No cerne do sistema está o LLM, o modelo central de IA responsável por gerar respostas textuais semelhantes às

humanas. A *Vector Store* é onde a magia acontece: um sistema de armazenamento dedicado que abriga incorporações e seus dados textuais correspondentes, garantindo uma recuperação rápida e eficiente. O *Vector Store Retriever* age como o mecanismo de busca do sistema, recuperando documentos relevantes por meio da comparação de similaridades vetoriais, assegurando que as informações mais pertinentes estejam sempre acessíveis.

O *Embedder* tem a função de converter informações textuais em representações vetoriais antes de armazenar ou recuperar dados, tornando-os compreensíveis para o sistema. Cada interação começa com a consulta ou declaração do usuário, capturada pelo *Prompt*, que prepara o terreno para os processos de recuperação e geração.

Com uma grande quantidade de dados para processar, o *Document Loader* é essencial, importando e lendo documentos, preparando-os para segmentação e incorporação. O *Document Chunker* fragmenta os documentos em partes menores e mais digeríveis, tornando os dados mais gerenciáveis e eficientes para recuperação.

Por fim, a Ferramenta de Entrada do Usuário captura a consulta ou declaração fornecida pelo usuário final, iniciando todo o processo RAG.

Através do Langchain, um framework conhecido por simplificar a criação eficiente de aplicativos baseados em Modelos de Linguagem (LLMs) e sistemas conversacionais, é viabilizado o carregamento do XML da Wikipedia em português. Compreender a estrutura do Langchain é fundamental para adotar uma abordagem unificada na criação e implementação padronizada de LLMs em diferentes aplicativos. Destaca-se a colaboração do Langchain com o Hugging Face - uma plataforma no GitHub que disponibiliza mais de 120 mil modelos - proporcionando um potencial significativo para o desenvolvimento de LLMs adaptáveis a uma ampla variedade de casos de uso, ampliando sua aplicabilidade e eficácia.

O RAG é uma ferramenta versátil que pode ser usada em uma ampla variedade de cenários para beneficiar a sociedade. Por exemplo, pode ser usado para fornecer informações médicas confiáveis, respondendo a perguntas abertas sobre esse tópico. Também pode ser usado para ajudar as pessoas a serem mais eficazes em seus empregos, fornecendo-lhes insights e ferramentas para melhorar sua produtividade.

C. Fine-tuning vs. RAG

Os Modelos de Linguagem de Ajuste Fino são conhecidos por sua adaptabilidade em uma ampla gama de tarefas em Processamento de Linguagem Natural. Esses modelos são especialmente eficazes em atividades como classificação de texto, análise de sentimento, geração de texto, entre outras, focadas na compreensão e produção de texto a partir de entradas variadas.

Por outro lado, os modelos de Geração Aumentada de Recuperação se destacam em cenários que exigem acesso a fontes externas de conhecimento. Eles são particularmente relevantes para responder a perguntas em domínios abertos, resumir documentos extensos ou mesmo para chatbots que fornecem informações provenientes de bases de conhecimento.

D. Wikipédia Brasil

A Wikipédia foi lançada em janeiro de 2001 por Larry Sanger e Jimmy Wales. Sua finalidade era criar uma enciclopédia online que reunisse o conhecimento humano de forma colaborativa, permitindo que qualquer internauta participasse da construção do projeto. Diferentemente das enciclopédias tradicionais, a Wikipédia inovou ao ser redigida em colaboração por qualquer pessoa disposta a contribuir, em vez de um grupo de especialistas, normalmente professores. A ideia era produzir uma enciclopédia que fosse abrangente e aprofundada, refletindo a diversidade de conhecimento disponível na comunidade online. Em dezembro de 2023, a Wikipédia lusófona ocupava a décima-oitava posição em número de artigos, contando neste momento com 1 114 303 artigos [4].

III. METODOLOGIA

Foram feitas pesquisas exploratórias acerca do tema, e definidos algumas etapas que envolviam: obtenção dos dados e pré-processamento, processamento e comparações dos resultados obtidos.

A. Obtenção dos dados e pré-processamento

Como a atividade envolvia treinamento do modelo com a base do subdomínio português da Wikipédia, foi necessário obtê-la. No próprio site da organização é disponibilizado vários repositórios, foi utilizado a base disponibilizada no diretório brasileiro em <https://wikipedia.c3sl.ufpr.br/ptwiki/> a base do dia 20/09/2023.

Como visto anteriormente, para realizar o fine-tuning, é necessário submeter o modelo a perguntas e respostas, mas dada a diversidade de artigos presentes na base, restringimos o treinamento para o tema "História do Brasil". Para o treinamento foram realizados 200 perguntas e suas respectivas respostas, e por falta de um profissional da área, as obtivemos utilizando o ChatGPT da empresa OpenAI, para tal submetemos trechos dos textos e solicitamos ao modelo que gerasse perguntas e respostas.

A desenvolvedora do modelo disponibiliza a biblioteca, em python, chamada HF (Hugging Face), e necessita que os dados estejam formatados em JSONL, contendo um objeto JSON por linha, cada um representando um único exemplo. Após obtidos o arquivo JSONL, as perguntas e respostas foram divididas em dois conjuntos, sendo treino e teste, utilizando o algoritmo desenvolvido particiona.py.

Durante o refinamento do modelo, foram utilizadas bibliotecas específicas da Hugging Face, incluindo:

- Transformers
- Peft
- Trl

O PEFT (Partial Efficient Fine-Tuning) é uma ferramenta que permite ajustar eficientemente modelos de linguagem, preservando parâmetros essenciais e evitando ajustes completos. Seus benefícios incluem a redução de custos computacionais e de armazenamento, prevenção do esquecimento catastrófico e melhor desempenho em cenários de dados limitados [3].

Dentro do PEFT, existem métodos distintos, sendo o LoRA (Layer-wise Relevance Adaptive) um dos mais utilizados. Ele seleciona subconjuntos de parâmetros com base na importância, enquanto a variante QLoRA utiliza quantização de 4 bits para reduzir ainda mais a demanda de memória [3].

O QLoRA introduz várias inovações projetadas para reduzir o uso de memória sem sacrificar o desempenho:

- 1) **NormalFloat de 4 bits:** um tipo de dados de quantização teoricamente ótimo para dados normalmente distribuídos, que oferece resultados empíricos melhores do que Inteiros de 4 bits e Floats de 4 bits.
- 2) **Dupla Quantização:** um método que quantiza as constantes de quantização, economizando em média cerca de 0,37 bits por parâmetro (aproximadamente 3 GB para um modelo de 65 bilhões de parâmetros).
- 3) **Otimizadores Paginados:** utilizando a memória unificada da NVIDIA para evitar picos de memória de checkpoint de gradientes que ocorrem ao processar um mini lote com um comprimento de sequência longo.

O QLoRA inclui um tipo de armazenamento de dados de baixa precisão, tipicamente de 4 bits, e um tipo de dados de computação geralmente designado como BFloat16. Na prática, ao utilizar um tensor de peso QLoRA, ocorre a desquantização do tensor para BFloat16, seguida pela execução de uma multiplicação de matriz em 16 bits.

Essas contribuições são combinadas em uma abordagem LoRA melhor ajustada, que inclui adaptadores em cada camada da rede e, portanto, evita quase todos os comprometimentos de precisão observados em trabalhos anteriores.

Paralelamente, a biblioteca TRL (Transformers Reinforcement Learning) foi empregada para treinar modelos de linguagem através do aprendizado por reforço. Sua API de Supervised Fine-Tuning (SFT) facilita a personalização e treinamento de modelos com conjuntos de dados customizados.

Detalhes de um processo de ajuste fino: O código usado configura um ambiente de treinamento para um modelo usando a biblioteca Transformers. O *Trainer* é inicializado com parâmetros essenciais, como o modelo a ser treinado, conjuntos de dados de treinamento e avaliação já tokenizados, e argumentos de treinamento específicos. Estes incluem a definição do diretório de saída para armazenar os resultados, a estratégia de salvamento do modelo, a frequência de salvamento e avaliação durante o treinamento, juntamente com configurações relacionadas à otimização, como o uso de ponto flutuante de 16 bits (bf16), o otimizador AdaFactor e o tipo de agendamento de taxa de aprendizado (cosseno), com uma taxa de aprendizado de $1e-5$. Além disso, há a opção para avaliar o modelo no final do treinamento e registrar logs em um diretório específico. O código também inclui a opção de usar o *Weights & Biases* para o acompanhamento do treinamento, se desejado, e define um nome para o run atual baseado na data e hora de execução. Essa configuração detalhada permite um controle preciso do processo de treinamento de modelos de linguagem utilizando a biblioteca Transformers.

Os dados de treinamento para Modelos de Linguagem de Ajuste Fino são específicos da tarefa em questão, geralmente constituídos por exemplos rotulados correspondentes à tarefa

almejada. No entanto, esses conjuntos de dados não incorporam mecanismos explícitos de recuperação.

Já os modelos de Geração Aumentada de Recuperação são treinados para operações combinadas de recuperação e geração, normalmente utilizando uma mistura de dados supervisionados (para geração de conteúdo) e dados que demonstram como recuperar e utilizar informações externas.

IV. RESULTADOS

Um dos parâmetros ajustáveis são a temperatura e quantidade máxima de tokens. Temperatura é um parâmetro que varia entre 0-1, e impacta na criatividade das respostas geradas. Temperaturas altas (por exemplo, 0,8) levam a respostas mais criativas, mas potencialmente menos coerentes, por outro lado, temperaturas mais baixas (por exemplo, 0,2) geram respostas mais determinísticas e focadas, mas potencialmente mais previsíveis. Este parâmetro deve ser modificado conforme a personalidade que deseja do modelo. Estabelecer a quantidade máxima de tokens permite controlar o tamanho das respostas e garante que elas sejam adequadas, além de controlar os recursos computacionais, proporcionando uma experiência mais eficiente, com entrega de informações relevantes de forma clara e sucinta. O limite da LLaMa2 é de 4096 tokens. Nos testes, os melhores resultados foram alcançados quando 0,02 de temperatura e o 400 como máximo de tokens gerados. O impacto de ajustar o número máximo de tokens depende da aplicação específica. Em alguns casos, respostas mais longas podem ser benéficas, enquanto em outros, respostas curtas e concisas são preferíveis. A seguir estão exemplos de algumas perguntas que foram feitas ao modelo nas 3 condições: sem ajuste, com RAG e com fine-tuning.

1) *Qual a capital do Brasil?*:

LLaMa2-13b: A capital do Brasil é Brasília, localizada no Distrito Federal.

LLaMa2-13b RAG: The capital of Brazil is Brasília.

LLaMa2-13b fine-tuning: A capital do Brasil é Brasília, localizada no Distrito Federal, no centro-oeste do país. A cidade foi planejada e construída a partir da década de 1950 como uma nova capital para substituir Rio de Janeiro, que era a antiga capital colonial do Brasil. A transferência da capital federal para Brasília ocorreu em 21 de abril de 1960.

2) *Quem foi Tiradentes e qual foi sua importância histórica?*:

LLaMa2-13b: Tiradentes (1743-1792) foi um líder político e militar brasileiro, considerado o "pai da independência do Brasil". Ele foi um dos principais organizadores da Inconfidência Mineira, uma conspiração que visava a independência do Brasil de Portugal.

LLaMa2-13b RAG: Tiradentes é considerado um dos principais líderes da independência do Brasil. Ele foi um advogado, militar e político que lutou pela causa da independência do país. Ele foi capturado e executado pelas autoridades portuguesas em 1792, mas seu legado como um defensor da liberdade e da democracia é ainda hoje lembrado e celebrado no Brasil.

LLaMa2-13b fine-tuning: Tiradentes foi um líder político, militar e escritor brasileiro, considerado o "pai

da independência do Brasil". Ele foi um dos principais líderes da Inconfidência Mineira, movimento que buscava a independência do Brasil da Coroa Portuguesa.

3) *Qual era o nome do país antes de ser chamado de Brasil?*:

LLaMa2-13b : A Ilha de Vera Cruz foi o nome dado pelos portugueses à região que hoje em dia é conhecida como o Brasil, quando foi descoberta em 1500 por Pedro Álvares Cabral. O nome "Brasil" só foi adotado oficialmente em 1822, após a independência do país.

LLaMa2-13b RAG : Não é possível responder à pergunta com base nas informações fornecidas, pois o texto não menciona qualquer país ou território antes da formação do Brasil. O texto se concentra em descrições da vida no Nordeste do Brasil na época da independência e na Guerra do Paraguai.

LLaMa2-13b fine-tuning : O nome do país antes de ser chamado de Brasil era "Ilha de Vera Cruz" ou "Terra de Vera Cruz". Fonte: "História do Brasil", de Lilia Moritz Schwarcz.

A Ilha de Vera Cruz foi descoberta em 1500 por Pedro Álvares Cabral, que a batizou em homenagem à Virgem Santa Maria, padroeira de Portugal. A partir de então, o nome foi usado para se referir ao território que hoje é o Brasil. A partir do século XVII, o nome "Brasil" passou a ser usado com mais frequência, mas o nome "Ilha de Vera Cruz" continuou a ser usado oficialmente até o século XIX.

4) *Quando foi proclamada a independência do Brasil?*:

LLaMa2-13b : 7 de setembro de 1822.

LLaMa2-13b RAG : A independência do Brasil foi proclamada em 1822.

LLaMa2-13b fine-tuning : A independência do Brasil foi proclamada em 7 de setembro de 1822. A declaração de independência foi feita por Dom Pedro I, que foi aclamado como imperador do Brasil. A independência do Brasil foi reconhecida oficialmente em 12 de outubro de 1822, quando o Tratado de Rio de Janeiro foi assinado entre o Brasil e Portugal.

5) *Quando foi assinada a primeira Constituição brasileira?*:

LLaMa2-13b : A primeira Constituição brasileira foi assinada em 21 de março de 1824.

LLaMa2-13b RAG : Não sou capaz de responder à sua pergunta, pois a constituição não existe. A Constituição Federal brasileira é uma lei que estabelece as normas fundamentais do país e foi promulgada em 1988. Antes disso, o Brasil era governado por um regime militar, e a última constituição anterior à atual foi promulgada em 1967.

LLaMa2-13b fine-tuning : A primeira Constituição brasileira foi assinada em 24 de fevereiro de 1824. Ela foi promulgada por Dom Pedro I, que havia sido proclamado imperador em 1822. A Constituição estabeleceu a monarquia constitucional no Brasil e estabeleceu os princípios da liberdade, da igualdade e da democracia.

6) *Quem foi a primeira presidente mulher do Brasil?*:

LLaMa2-13b : Dilma Rousseff.

LLaMa2-13b RAG : Não é possível responder à pergunta com base nas informações fornecidas, pois Fernando Collor de Mello é um homem e não uma mulher. A primeira presidente mulher do Brasil foi Dilma Rousseff, que governou de 2011 a 2016.

LLaMa2-13b fine-tuning : Dilma Rousseff foi a primeira presidente mulher do Brasil, eleita em 2010 e reeleita em 2014. Ela foi a 36ª presidente do Brasil e governou até 2016, quando foi afastada do cargo por impeachment.

7) *Quem foi o primeiro imperador do Brasil?:*

LLaMa2-13b : Dom Pedro I.

LLaMa2-13b RAG : The first emperor of Brazil was Pedro I (1822-1831).

LLaMa2-13b fine-tuning : Dom Pedro I (1822-1831) foi o primeiro imperador do Brasil. Ele foi coroado em 1 de dezembro de 1822, após a proclamação da independência do Brasil em 7 de setembro de 1822. Dom Pedro I foi o único imperador do Brasil, governando por cerca de nove anos antes de abdicar em favor de seu filho, Dom Pedro II.

8) *Qual foi o período do governo de Getúlio Vargas?:*

LLaMa2-13b : O período do governo de Getúlio Vargas foi de 1930 a 1945, e novamente de 1951 a 1954. Ele governou o Brasil como ditador durante o regime conhecido como Estado Novo, e posteriormente como presidente eleito após a redemocratização do país.

LLaMa2-13b RAG : O período do governo de Getúlio Vargas foi de 1930 a 1945, e novamente de 1951 a 1954.

LLaMa2-13b fine-tuning : O período do governo de Getúlio Vargas foi de 1930 a 1945, com interrupções. Ele foi eleito presidente em 1930 e governou até 1945, quando foi deposto. Durante seu governo, houve muitas mudanças políticas, econômicas e sociais no Brasil. Ele implementou políticas de contenção da inflação, estabilizou a moeda e implementou programas de desenvolvimento econômico. Ele também estabeleceu o Estado Novo, um regime autoritário que durou de 1937 a 1945.

9) *Quando foi abolida a escravidão no Brasil?:*

LLaMa2-13b : A escravidão foi abolida no Brasil em 13 de maio de 1888, através da Lei Áurea.

LLaMa2-13b RAG : A escravidão no Brasil foi abolida em 13 de maio de 1888, conforme a Lei Áurea (Lei nº 3.353).

LLaMa2-13b fine-tuning : A escravidão foi abolida no Brasil em 13 de maio de 1888, por meio da Lei Áurea, assinada por D. Pedro II. Essa lei aboliu a escravidão em todo o território brasileiro, libertando cerca de 20 milhões de escravos. No entanto, a escravidão continuou a existir de facto em alguns estados, especialmente no Nordeste, até o final do século XIX.

10) *Quem foi o presidente durante a Revolução de 1930?:*

LLaMa2-13b : Getúlio Vargas.

LLaMa2-13b RAG : Não posso responder à sua pergunta, pois não tenho conhecimento sobre a Revolução de 1930.

LLaMa2-13b fine-tuning : Getúlio Vargas foi o presidente durante a Revolução de 1930. Ele foi eleito em 1929 e tomou posse em 1º de março de 1930. A Revolução de 1930 ocorreu em outubro daquele ano, quando Getúlio Vargas foi deposto e substituído pelo marechal Pedro de Toledo.

V. CONCLUSÕES

A análise das respostas dos três modelos às perguntas sobre a história do Brasil revela sua capacidade de oferecer respostas precisas e abrangentes, mesmo diante de questões abertas, desafiadoras ou ambíguas.

É importante notar que esses modelos podem incorrer em erros, seja pela falta de das informações pertinentes não estavam presentes no momento do treinamento, ou seja quando as perguntas são mal formuladas. Não são raras as situações em que o mesmo modelo, em determinadas ocasiões respondem perfeitamente e sem nenhum ajuste ele responde de forma incorreta, o que reforça a importância de quão elaborado deve ser o seu prompt, para uma resposta adequada.

O experimento destacou que o LLaMa2-13b puro geralmente apresenta respostas concisas, enquanto o LLaMa2-13b RAG oferece respostas mais detalhadas, e o LLaMa2-13b fine-tuning fornece as respostas mais completas. Tomando como exemplo a pergunta "Quando foi abolida a escravidão no Brasil?", é evidente que:

Todas as respostas dos modelos LLaMa são precisas e abrangentes. A resposta correta indica que a escravidão foi abolida em 13 de maio de 1888, por meio da Lei Áurea.

O LLaMa2-13b oferece uma resposta concisa, porém completa, ao fornecer a data e o mecanismo da abolição da escravidão.

Já o LLaMa2-13b RAG fornece uma resposta mais detalhada, abordando informações sobre o número de escravos libertados e a persistência da escravidão em alguns estados, visto que estava presente neste material, no entanto, caso fosse perguntado algo que não constava, o algoritmo foi instruído a não responder.

Por fim, o LLaMa2-13b fine-tuning apresenta a resposta mais completa, incluindo detalhes sobre o número de escravos libertados, a persistência da escravidão em determinados estados e o fato de a Lei Áurea ter sido assinada por D. Pedro II. As técnicas de fine-tuning e RAG não são dissociativas, e é comum a sua utilização combinada.

REFERENCES

- [1] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandez, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rishi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey

- Edunov, Thomas Scialom. *LLaMa 2: Open Foundation and Fine-Tuned Chat Models*. 2023. *arXiv preprint* arXiv:2307.09288, *cs.CL*.
- [2] Sewon Min, Mike Lewis, Luke Zettlemoyer, Hannaneh Hajishirzi. *MetalCL: Learning to Learn In Context*. 2022. *arXiv preprint* arXiv:2110.15943, *cs.CL*.
- [3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. *arXiv preprint* arXiv:2305.14314, *cs.LG*.
- [4] Wikipédia, *Wikipédia em português* — *Wikipédia, a enciclopédia livre*, 2023, https://pt.wikipedia.org/w/index.php?title=Wikip%C3%A9dia_em_portugu%C3%AAs&oldid=66593297, [Online; acessado em 14 de setembro de 2023].