Aprimorando Geração Aumentada por Recuperação via Ajuste Fino Sequencial de Modelos de Linguagem Pequenos

Ronaldinho Vega Centeno Olivera[1],
Frances A. Santos[1], **Julio Cesar dos Reis**[1], **Allan M. de Souza**[1]

[1]Instituto de Computação - Universidade Estadual de Campinas (UNICAMP)
CEP : 13083-970 – Campinas – SP – Brazil

ronaldinho@lrc.ic.unicamp.br
{frances.santos, jreis, allanms}@ic.unicamp.br

Abstract. *Language Models (LMs) excel in general knowledge but often face challenges in specialized domains, where complexity and constant evolution pose additional obstacles. This study enhances the performance of Retrieval-Augmented Generation (RAG) systems for the Question Answering (QA) task through sequential fine-tuning of the RAG components, employing Small Language Models (SLMs). Our approach adjusts both the embedding model and the generative model using minimal computational resources and improves overall effectiveness compared to vanilla RAG. The proposed methodology, scalable and cost-effective, enables the practical application of RAG systems across different domains and tasks.*

Resumo. *Modelos de linguagem (Language Models, LMs) se destacam em conhecimento geral, mas frequentemente enfrentam dificuldades em domínios especializados, nos quais a complexidade e a constante evolução representam desafios adicionais. Este estudo visa aprimorar a efetividade de sistemas de Geração Aumentada por Recuperação (Retrieval-Augmented Generation, RAG) para a tarefa de Perguntas e Respostas (Question Answering, QA) por meio do ajuste sequencial dos componentes do RAG, utilizando modelos de linguagem pequenos (Small Language Models, SLMs). Nossa abordagem ajusta tanto o modelo de embedding quanto o modelo generativo utilizando poucos recursos computacionais e melhora a efetividade geral em relação ao vanilla RAG. A metodologia proposta, escalável e econômica, viabiliza a aplicação prática de sistemas RAG em diferentes domínios e tarefas.*

1. Introdução

Os Grandes Modelos de Linguagem (*Large Language Models*, LLMs) revolucionaram o Processamento de Linguagem Natural (*Natural Language Processing*, NLP), em especial nas tarefas de perguntas e respostas (*Question Answering*, QA), sumarização de texto e agentes conversacionais [Fan et al. 2024]. Seu treinamento de propósito geral em conjuntos de dados amplos e variados permite que tais modelos tenham conhecimento abrangente e boas capacidades linguísticas.

Embora os LLMs ofereçam oportunidades transformadoras, enfrentam limitações notáveis em contextos específicos de domínio como telecomunicações, finanças, saúde e direito, em que a complexidade, a terminologia especializada representam desafios

[Zhou et al. 2024, Siriwardhana et al. 2023] prejudicando sua capacidade de recuperar e gerar respostas com precisão.

Para enfrentar esses desafios, duas metodologias principais emergiram: ajuste fino (*Fine-Tuning*, FT) e geração aumentada por recuperação (*Retrieval-Augmented Generation*, RAG). O *ajuste fino* envolve treinamento adicional em conjuntos de dados específicos de domínio para adaptar os LLMs a tarefas especializadas [Fan et al. 2024]. Embora eficaz, essa abordagem pode ser computacionalmente custosa e menos adaptável em áreas cujo conteúdo muda frequentemente. Por outro lado, RAG combina o conhecimento paramétrico de um modelo generativo com o conhecimento não paramétrico, extraído de bases de dados externas. As informações recuperadas são utilizadas como contexto para aprimorar o processo de geração, permitindo a integração dinâmica do conhecimento específico do domínio [Lewis et al. 2020]. Isso torna o RAG particularmente adequado para domínios técnicos, permitindo que os modelos forneçam respostas contextualmente relevantes e precisas [Gao et al. 2024].

Outro desafio inerente à utilização de LLMs é a alta demanda computacional, tanto para treinamento quanto para inferência, dificultando sua implementação em ambientes com recursos limitados. Seu consumo de energia gera implicações ambientais e financeiras notáveis. Por exemplo, o treinamento do GPT-3 gerou aproximadamente 502 toneladas de emissões de CO_2, o que equivale a $27,77$ vezes a média da pegada de carbono anual de uma pessoa nos Estados Unidos [Maslej et al. 2023].

Nesse contexto, Modelos de Linguagem de Pequena Escala (*Small Language Models*, SLMs), como o *bge-small-en-v1.5* [Xiao et al. 2024] e o *phi-2* [Microsoft Research 2023], oferecem uma alternativa prática e ambientalmente consciente. Esses modelos proporcionam eficiência computacional, menor consumo de energia e impacto ambiental reduzido, mantendo um desempenho competitivo. Isso os torna adequados para domínios que exigem inferência em tempo real e baixo consumo de recursos. Técnicas como o *Parameter-Efficient Fine-Tuning* (PEFT), complementam os SLMs ao permitir que apenas um pequeno número de parâmetros adicionais seja ajustado ou que um subconjunto dos parâmetros pré-treinados seja atualizado. Isso preserva o conhecimento capturado pelo modelo pré-treinado enquanto o adapta a tarefas específicas, reduzindo o consumo de recursos computacionais [Han et al. 2024]. Por sua adaptabilidade e eficiência, os SLMs aliados às técnicas PEFT tornam-se soluções particularmente adequadas para a integração da Inteligência Artificial em domínios específicos, permitindo lidar com requisitos técnicos e contextos especializados.

Neste artigo, propomos uma abordagem que utiliza uma estratégia de *ajuste fino* sequencial [Fan et al. 2024], denominada **RAG-SFT**, para otimizar o desempenho do RAG nos domínios de telecomunicações, saúde e de conhecimento geral, utilizando SMLs. Primeiro, ajustamos o modelo de *embeddings bge-small-en-v1.5* em dados específicos de domínio para recuperar informações relevantes de documentos do domínio. O contexto recuperado é então usado para ajustar o modelo *phi-2* para geração de respostas textuais, aprimorando sua capacidade de gerar respostas mais exatas e adequadas ao contexto. Para alcançar isso de maneira eficiente, utilizamos a técnica PEFT de *Low-Rank Adaptation* (LoRA) [Hu et al. 2022], que reduz os custos computacionais enquanto mantém a alta qualidade de desempenho. Utilizamos bases de dados vetoriais indexadas usando o *Facebook AI Similarity Search* (FAISS) [Johnson et al. 2021], permitindo

recuperação eficiente e precisa.

Nossos resultados experimentais demonstram que a abordagem proposta melhorou significativamente tanto a recuperação de informações quanto a geração de respostas. Especificamente, o ajuste fino do modelo *bge-small-en-v1.5* contribuiu para um desempenho superior na recuperação de informação, enquanto o ajuste fino do modelo *phi-2* resultou em melhorias na qualidade da geração de respostas. Ao combinar esses dois componentes por meio de um ajuste fino sequencial, observamos uma sinergia que potencializou ainda mais o desempenho geral do pipeline RAG.

Ao preencher a lacuna entre LLMs de propósito geral e o conhecimento específico de domínio, esta investigação avança o uso de AI no processamento de conteúdo técnico especializado, oferecendo uma solução escalável e econômica. Para promover a reprodutibilidade e fomentar pesquisas futuras, disponibilizamos publicamente nosso código[1].

O restante desse artigo está organizado da seguinte forma: a Seção 2 explica os conceitos fundamentais, incluindo RAG e ajuste fino. A Seção 3 revisa estudos que aplicam RAG com em diversos domínios. A Seção 4 detalha a abordagem proposta. A Seção 5 apresenta os resultados experimentais enquanto a Seção 6 discute as principais descobertas. Por fim, a Seção 7 sintetiza as conclusões e direções futuras.

2. Fundamentação Teórica

Esta seção aborda os principais conceitos técnicos que sustentam o desenvolvimento deste trabalho, com foco em temas como RAG, ajuste fino e o uso de técnicas de PEFT para adaptar modelos de linguagem a contextos específicos.

RAG é uma abordagem que aprimora Modelos de Linguagem (*Language models*, LMs) integrando a recuperação de conhecimento externo ao processo generativo [Lewis et al. 2020]. Sistemas RAG recuperam informações relevantes de bases de conhecimento externas com base em uma consulta, aumentam o contexto com essas informações recuperadas e, em seguida, geram respostas utilizando um modelo de linguagem. Esse *framework* reduz alucinações, melhora a factualidade e permite que as saídas do modelo sejam mais precisas. A principal tarefa do RAG é QA [Gao et al. 2024], embora também seja aplicado em tarefas como extração de informação, geração de diálogo e busca de código, demonstrando sua versatilidade em aplicações intensivas em conhecimento.

O ajuste fino é essencial para adaptar modelos a domínios especializados com jargões ou estruturas específicas, como as áreas da saúde, jurídica ou de telecomunicações, onde modelos pré-treinados frequentemente apresentam limitações. Para modelos de *embedding*, o ajuste fino aprimora as representações vetoriais para melhorar a precisão da recuperação em contextos específicos de domínio, utilizando técnicas como aprendizado contrastivo para alinhar consultas e documentos [Gao et al. 2024]. Para os modelos generativos, o ajuste fino permite a incorporação de conhecimento especializado, adaptação a formatos de dados específicos e geração de respostas ajustadas ao estilo desejado.

Uma técnica de PEFT para ajuste fino é LoRA, que reduz significativamente o número de parâmetros treináveis ao introduzir matrizes de decomposição de baixo *rank*

[1] `https://github.com/DinhoVCO/RAG_SFT`

nas camadas do Transformer, mantendo os pesos do modelo pré-treinado congelados [Hu et al. 2022]. Isso torna o ajuste mais eficiente em termos computacionais, permitindo a adaptação a novas tarefas sem modificar toda a estrutura do modelo. Um único modelo pré-treinado pode ser reutilizado para diversas tarefas por meio da troca das matrizes injetadas, o que reduz o armazenamento necessário e o custo de alternância entre tarefas. Como apenas essas matrizes de baixo rank são otimizadas, LoRA diminui o custo do treinamento.

3. Trabalhos Relacionados

A Geração Aumentada por Recuperação (RAG) representa uma estratégia fundamental para superar as limitações de LMs em domínios especializados, onde a precisão e a atualização do conhecimento são cruciais [Lewis et al. 2020]. No entanto, a otimização de sistemas RAG apresenta desafios significativos, e grande parte da literatura existente concentra-se em melhorias parciais do pipeline, focando ou no componente de recuperação ou no de geração, mas raramente em ambos de forma conjunta.

Estudos iniciais no campo validaram a eficácia fundamental do RAG no domínio das telecomunicações. [Maatouk et al. 2023], por exemplo, demonstraram que o desempenho de LLMs melhora significativamente com a simples adição de contexto externo, uma implementação básica de RAG. Contudo, o estudo se limita à avaliação e, embora identifique a necessidade de modelos especializados, não propõe metodologias para otimizar os componentes do RAG a fim de alcançar tal especialização. Expandindo essa premissa, [Piovesan et al. 2024] mostraram que um SLM como o phi-2, quando aumentado com RAG, pode se tornar competitivo com modelos muito maiores, como o GPT-3.5. A principal limitação dessa abordagem, no entanto, reside em sua implementação, que não aplica nenhum tipo de ajuste fino: tanto o modelo de embedding (bge-base-en-v1.5) quanto o modelo generativo (phi-2) são utilizados em suas versões pré-treinadas. Essa falha em especializar os componentes, especialmente o de recuperação, pode levar à seleção de documentos de menor relevância, restringindo a qualidade do contexto e resultando em um desempenho geral subótimo do sistema [Gao et al. 2024].

Dando um passo adiante na especialização de componentes, outros pesquisadores se concentraram no ajuste fino do modelo generativo. [Rosenthal et al. 2025], por exemplo, desenvolveram um benchmark para avaliar sistemas RAG na geração de respostas longas e coesas. Para isso, optaram por uma abordagem de full fine-tuning, realizando o treinamento completo de todos os 783 milhões de parâmetros do modelo FLAN-T5-Large, ajustando-o com exemplos que incluíam a passagem de referência (gold passage) como contexto.

Essa estratégia, embora eficaz, pode ser computacionalmente custosa, mesmo para modelos de menor escala. Os resultados mostraram que este modelo ajustado aprendeu a gerar respostas concisas, com um comprimento médio próximo ao das respostas de referência, e alcançou um alto desempenho na identificação de perguntas sem resposta. No entanto, esta abordagem de ajuste fino se concentrou exclusivamente no componente gerador, operando sob a premissa de uma recuperação de passagens ideal, o que não reflete os cenários realistas onde a recuperação de informação é, em si, um desafio.

Em contraste com essas abordagens parciais, nossa proposta aborda as limitações identificadas por meio de uma otimização integral do pipeline. Propomos um ajuste fino

sequencial que otimiza, em primeiro lugar, o modelo de embedding para melhorar a precisão da recuperação de informações no domínio específico e, em segundo lugar, o modelo generativo, para que ele aprenda a utilizar de forma mais eficaz esse contexto de alta qualidade. Acreditamos que esta otimização sinérgica é crucial, pois a qualidade da geração depende diretamente da relevância do material recuperado. Dessa forma, nossa metodologia busca preencher a lacuna existente na literatura, oferecendo uma solução mais robusta e completa para a aplicação de sistemas RAG em contextos especializados.

4. Metodologia

Nossa abordagem segue uma metodologia sequencial, em que cada componente aprimorado se baseia no anterior, garantindo uma integração efetiva e resultados otimizados em cada etapa.

O processo inicia com o ajuste fino do modelo de recuperação para cada conjunto de dados para melhorar a precisão na recuperação de documentos relevantes. Em seguida, os documentos são segmentados usando a estratégia de *Divisão Recursiva por Tokens*. Esses segmentos são então codificados e armazenados em uma base de dados vetorial com FAISS. Finalmente, o modelo phi-2 é ajustado usando LoRA para cada conjunto de dados, aproveitando os segmentos recuperados como contexto para aprimorar sua capacidade de gerar respostas precisas e específicas para o domínio. A Figura 1 apresenta o fluxo completo de nossa metodologia.

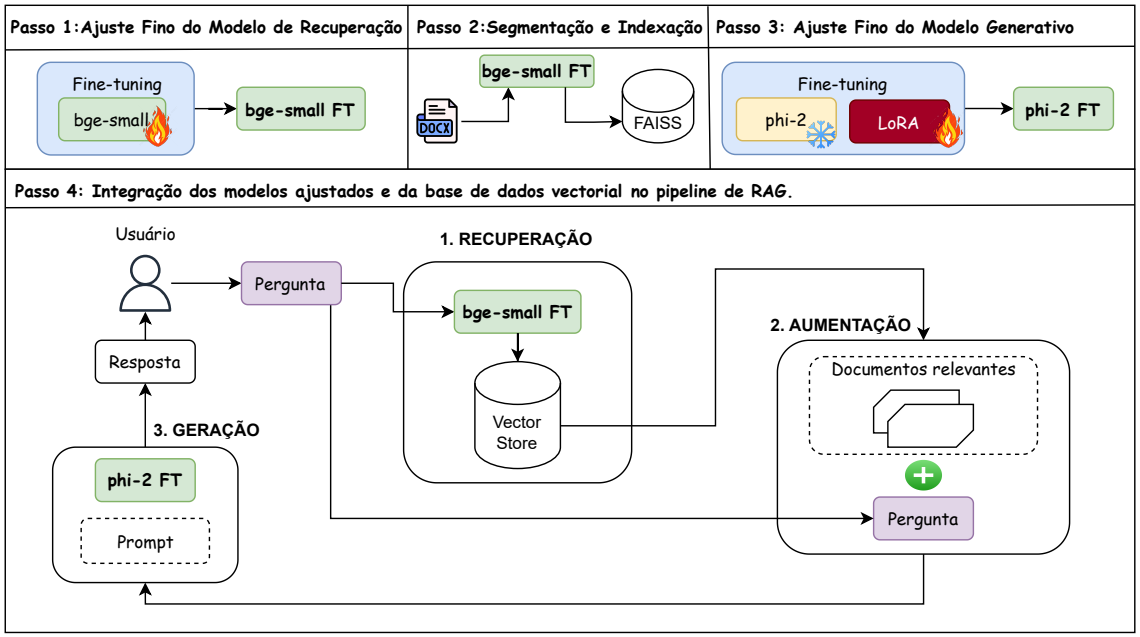

Figura 1. Fluxo da metodologia proposta (RAG-SFT). O processo inicia com o ajuste fino do modelo de embeddings (bge-small). Em seguida, os documentos são segmentados e indexados em uma base de dados vetorial. Posteriormente, o modelo generativo (phi-2) é ajustado com LoRA. Por fim, os dois modelos ajustados e a base de dados vetorial são integrados para compor o pipeline RAG final.

4.1. Conjunto de Dados

Utilizamos quatro conjuntos de dados de perguntas e respostas: dois de domínio geral e dois de domínio específico. Os conjuntos de domínio geral incluem o CLAP-NQ

[Rosenthal et al. 2025], um subconjunto do *Natural Questions* [Kwiatkowski et al. 2019] que incorpora perguntas sem resposta, e o *BoolQ* [Clark et al. 2019], focado em compreensão de leitura com perguntas do tipo sim/não.

Quanto aos conjuntos de domínio específico, utilizamos o *TeleQnA* [Maatouk et al. 2023], um conjunto de perguntas de múltipla escolha na área de telecomunicações, e o *COVID-QA* [Möller et al. 2020], que contém perguntas extraídas de artigos científicos relacionados à COVID-19. Para o *TeleQnA*, criamos um subconjunto filtrado contendo apenas perguntas relacionadas ao padrão 3GPP (3GPP-QA-MultipleChoice[2]) e um corpus composto por 554 documentos no formato *.docx* dos padrões da 3GPP Release 18 [3rd Generation Partnership Project (3GPP) 2023]. A Tabela 1 resume as principais características de cada um dos conjuntos de dados utilizados.

Tabela 1. Resumo dos conjuntos de dados utilizados com número de exemplos por divisão e total de passagens geradas.

Dataset	Treinamento	Validação	Teste	Total	Passagens Geradas
TeleQnA-3GPP	724	181	905	1810	378571
COVID-QA	1292	323	404	2019	8397
BoolQ	7541	1886	3270	12.756	13091
CLAP-NQ	2996	749	600	4345	261999

4.2. Ajuste Fino do Modelo de *Embeddings*

Utilizamos o modelo bge-small-en-v1.5 [Xiao et al. 2023] para a geração de *embeddings*, pois a família de modelos BGE em inglês alcançou um desempenho de ponta no *benchmark* MTEB [Muennighoff et al. 2023]. O processo de ajuste fino do modelo de *embeddings* bge-small-en-v1.5 empregou aprendizado contrastivo com a função de perda *Multiple Negatives Ranking* [Reimers and Gurevych 2019, Karpukhin et al. 2020], aproveitando *in-batch negatives*. O objetivo foi maximizar a similaridade entre a consulta e seu respectivo trecho positivo, minimizando ao mesmo tempo a similaridade com todos os outros trechos do lote. A função de perda é definida como:

$$\mathcal{L} = -\frac{1}{|B|}\sum_{i=1}^{|B|}\log\frac{\exp\left(\text{sim}(q_i, p_i)\right)}{\sum_{j=1}^{|B|}\exp\left(\text{sim}(q_i, p_j)\right)} \tag{1}$$

onde B representa o tamanho do lote (*batch size*), q_i é o *embedding* da consulta i, p_i é o *embedding* do trecho positivo correspondente a q_i, p_j é o *embedding* de qualquer outro trecho j no lote, e $\text{sim}(\cdot, \cdot)$ denota uma função de similaridade, especificamente a similaridade do cosseno.

O modelo foi ajustado separadamente para cada *dataset* com um tamanho de lote (*batch size*) de 128 e treinado por 10 épocas.

A Figura 2 apresenta a primeira etapa do treinamento sequencial em que ajustamos o modelo em cada domínio.

[2]https://huggingface.co/datasets/DinoStackAI/3GPP-QA-MultipleChoice

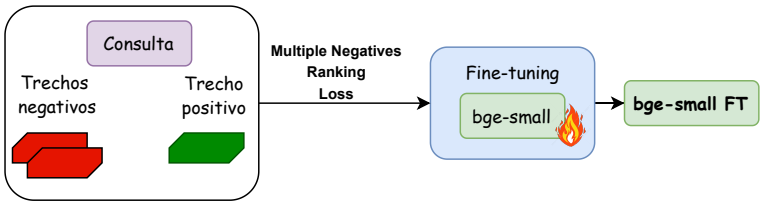

Figura 2. Ajuste Fino do Modelo de *Embeddings*.

4.3. Segmentação e Indexação

Para todos os experimentos, adotamos a estratégia de segmentação Recursiva por Tokens [LangChain 2023], uma abordagem hierárquica que divide os textos inicialmente em parágrafos ou sentenças, recorrendo a unidades menores quando necessário. Essa técnica tem como objetivo de preservar a coerência semântica dos trechos, ao mesmo tempo em que garante a compatibilidade com o limite de 512 *tokens* imposto pelo modelo de *embeddings* utilizado.

Para a geração das passagens, utilizamos, por padrão, um tamanho máximo de segmento (*chunk size*) de 150 *tokens*, pois esse valor fornece o melhor equilíbrio entre preservação do contexto e precisão na recuperação para o modelo phi-2 [Gichamba et al. 2024], com uma sobreposição (*overlap*) de 20 *tokens* entre segmentos consecutivos. O corpus de cada conjunto de dados foi processado utilizando a estratégia de segmentação. A Tabela 1 apresenta a quantidade de passagens geradas para cada *dataset* após esse processamento.

Posteriormente, cada uma dessas passagens foi vetorizada utilizando o modelo de embeddings ajustado (bge-small FT) e armazenada em um banco de dados vetorial, utilizando o índice **FlatIP** do FAISS para uma busca eficiente por similaridade, conforme ilustrado na Figura 3.

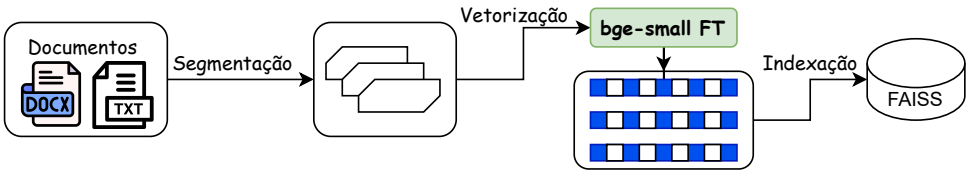

Figura 3. Fluxo de segmentação e indexação de documentos. O processo inicia com a divisão dos documentos em segmentos de texto, que são então vetorizados utilizando o modelo de *embeddings* bge-small previamente ajustado. Finalmente, os vetores são indexados em uma base de dados vetorial com FAISS.

4.4. Ajuste Fino do Modelo Generativo

Para o ajuste fino, utilizou-se o modelo de linguagem base Phi-2. Trata-se de um modelo com 2,7 bilhões de parâmetros que se destaca por suas capacidades de raciocínio e compreensão de linguagem, oferecendo desempenho de ponta entre os SLMs, mesmo sem ter sido alinhado com Aprendizagem por Reforço a partir de Feedback Humano (Reinforcement Learning from Human Feedback, RLHF) ou por ajuste de instruções [Microsoft Research 2023]. O processo de ajuste fino do modelo Phi-2 foi projetado para aprimorar sua capacidade de: *(i)* responder às perguntas no formato correto; *(ii)* infe-

rir respostas a partir dos documentos recuperados; e *(iii)* melhorar sua compreensão do domínio.

Para alcançar esses objetivos, utilizamos a técnica LoRA [Hu et al. 2022], que permite um ajuste fino eficiente ao modificar apenas um pequeno subconjunto dos parâmetros do modelo. No nosso caso, foram treinados $26,2$ milhões de parâmetros, o que representa 0,94% do total de $2,78$ bilhões de parâmetros do modelo original, evitando assim o alto custo computacional do treinamento completo.

A configuração utilizada para o ajuste fino incluiu um *rank* (r) de 32, um valor de LoRA alpha de 32 e os módulos alvo Wqkv, fc1 e fc2. Uma taxa de *dropout* de $0,05$ foi aplicada para melhorar a generalização. O tipo de tarefa foi definido como "CAUSAL_LM" para se alinhar ao objetivo de modelagem de linguagem. O ajuste foi conduzido individualmente para cada *dataset*, com um *batch size* de 10 ao longo de 5 épocas. A Figura 4 apresenta o pipeline de ajuste fine do modelo.

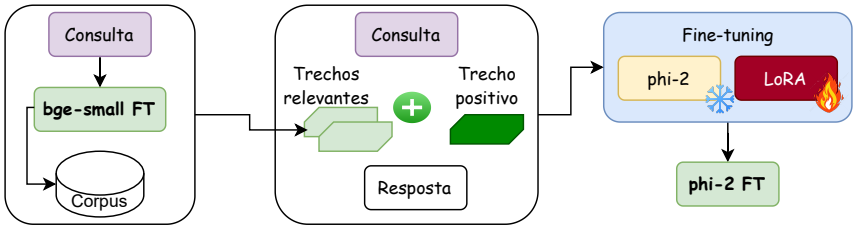

Figura 4. Ajuste Fino do Modelo phi-2.

Durante o treinamento, até as quatro principais passagens recuperadas são incluídas, garantindo que a passagem ouro esteja presente entre elas, inserida em uma posição escolhida aleatoriamente. O *prompt* a seguir foi utilizado para o ajuste fino do modelo phi-2:

Instruct: `{instruct}`
Context:
Document 1: `{passagem 1}`
Document 2: `{passagem ouro}`(posição aleatória)
Document 3: `{passagem 2}`
Document 4: `{passagem 3}`
Question:
`{question}`
Output:
`{answer}`

4.5. Procedimentos Experimentais

Os experimentos foram conduzidos com o objetivo de avaliar o impacto do ajuste fino de modelos de recuperação e geração em um pipeline RAG aplicado a quatro conjuntos de dados com diferentes características: TeleQnA, COVID-QA, BoolQ e CLAP-NQ. As avaliações foram organizadas em três etapas principais: *(i)* avaliação de modelos recuperadores, *(ii)* avaliação de modelos generativos, e *(iii)* integração e avaliação do pipeline RAG completo.

Avaliação dos Modelos Recuperadores. A performance dos modelos de *embeddings* foi avaliada utilizando duas métricas padrão em tarefas de recuperação de infor-

mação: *normalized Discounted Cumulative Gain* (nDCG@10) e *Mean Reciprocal Rank* (MRR@10). Os experimentos compararam versões dos modelos bge-small, bge-base e bge-large, bem como a versão ajustada bge-small FT (veja Seção 5.1 para os resultados).

Avaliação dos Modelos Generativos. As métricas utilizadas para avaliação variaram conforme as características de cada conjunto de dados: ROUGE-L e acurácia no TeleQnA; ROUGE-L foi usada para COVID-QA; acurácia no *dataset* BoolQ; ROUGE-L e acurácia para perguntas sem resposta (*unanswerable*) no CLAP-NQ. As versões base e ajustada do modelo generativo (phi-2 e phi-2 FT, respectivamente) foram avaliadas isoladamente e em combinação com diferentes modelos recuperadores (veja Seção 5.2 para os resultados).

Avaliação do Pipeline RAG e Estudos de Ablação. Para compor o pipeline RAG, os modelos recuperadores e generativos ajustados foram integrados, formando combinações como bge-small FT + phi-2 FT. Foram conduzidos estudos de ablação com o objetivo de quantificar o impacto individual do ajuste fino em cada componente (recuperador e generativo). O desempenho final foi avaliado em todos os *datasets*, utilizando as métricas apropriadas mencionadas anteriormente (veja Seção 5.3 para os resultados).

Infraestrutura e Configurações Computacionais. Todos os experimentos foram executados em uma estação equipada com GPU NVIDIA Quadro RTX 6000 (24 GB VRAM). Para otimizar o uso da GPU, os parâmetros de treinamento foram ajustados de acordo com as características de cada modelo. O modelo bge-small-en-v1.5 foi treinado por 10 épocas com um *batch size* de 128, enquanto o modelo phi-2 foi treinado por 5 épocas utilizando um *batch size* de 10, respeitando as limitações de memória e desempenho (veja Seção 5.4 para os resultados).

5. Resultados Experimentais

5.1. Resultados sobre o Modelo Recuperador

Para cada *dataset* avaliado, o modelo de *embeddings* bge-small-en-v1.5 foi ajustado. A avaliação foi conduzida utilizando métricas especificamente projetadas para tarefas de recuperação de informação, incluindo *normalized Discounted Cumulative Gain* (nDCG) e *Mean Reciprocal Rank* (MRR). Os resultados estão resumidos na Tabela 2.

Tabela 2. Resultados de nDCG@10 e MRR@10 no conjunto de teste para os quatro *datasets* usando diferentes modelos. FT indica os modelos com ajuste fino.

Retriever	TeleQnA		COVID-QA		BoolQ		CLAP-NQ	
	nDCG@10	MRR@10	nDCG@10	MRR@10	nDCG@10	MRR@10	nDCG@10	MRR@10
bge-small	0.94	0.92	0.72	0.68	0.84	0.80	0.88	0.86
bge-base	0.95	0.94	0.74	0.70	0.86	0.83	0.90	0.89
bge-large	0.95	0.94	0.75	0.71	0.86	0.83	0.89	0.87
bge-small FT	**0.98**	**0.97**	**0.86**	**0.83**	**0.87**	**0.84**	**0.92**	**0.91**

Nossos modelos ajustados, **bge-small FT**, apresentaram uma melhoria significativa de desempenho não apenas em relação ao bge-small e bge-base, mas também superaram o modelo bge-large, demonstrando que nossa abordagem de ajuste fino é altamente eficaz na melhoria da recuperação de informações, mesmo quando comparada a modelos de maior porte.

5.2. Resultados sobre o Modelo Generativo

O processo de ajuste fino do modelo Phi-2 foi realizado separadamente para cada conjunto de dados. A Tabela 3 apresenta uma comparação entre diferentes configurações.

Os resultados demonstraram que o processo de ajuste fino do modelo phi-2 (**phi-2 FT**) melhora o desempenho do modelo relação ao modelo base (**phi-2**) nos conjuntos de dados BoolQ, TeleQnA e COVID-QA.

5.3. Resultados do Pipeline RAG

Para avaliar a efetividade geral da nossa abordagem RAG, integramos nossos modelos ajustados e utilizamos as melhores configurações identificadas em cada etapa. O sistema RAG inclui o modelo de *embeddings* ajustado **bge-small FT**, a base de dados vectorial e o modelo gerativo ajustado **phi-2 FT**. Estudos de ablação foram conduzidos para analisar a contribuição de cada componente.

Tabela 3. Resultados para os quatro *datasets* usando diferentes modelos. FT indica os modelos com ajuste fino. As métricas de avaliação utilizadas foram: TeleQnA (Rouge-L, Accuracy), COVID-QA (Rouge-L), BoolQ (Accuracy) e CLAP-NQ (Rouge-L, Accuracy para perguntas sem resposta).

Retriever	Generator	TeleQnA		COVID-QA	BoolQ	CLAP-NQ	
		Rouge-L	Acc	Rouge-L	Acc	Rouge-L	Acc(un)
-	GPT 3.5	0.66	53.92	0.11	68.53	0.16	50.66
-	phi-2	0.31	29.61	0.08	63.85	0.19	-
-	phi-2 FT	0.68	56.13	0.14	63.73	-	-
bge-small	phi-2	0.51	69.50	0.20	71.46	0.27	-
bge-small FT	phi-2	0.53	71.16	0.21	73.11	**0.28**	-
bge-small	phi-2 FT	0.81	74.25	0.38	73.24	0.19	77.66
bge-small FT	**phi-2 FT**	**0.82**	**75.69**	**0.41**	**73.42**	0.17	**81.33**

A Tabela 3 apresenta uma comparação entre RAG basico (bge-small + phi-2) e nossa abordagem de RAG com ajuste sequencial (bge-small FT + phi-2 FT). Os resultados evidenciam melhorias consistentes em todos os conjuntos de dados após o ajuste fino, tanto do componente recuperador quanto do gerador.

No dataset TeleQnA, comparando a abordagem RAG básica (bge-small + phi-2) com a nossa proposta (bge-small FT + phi-2 FT), a métrica Rouge-L aumentou de 0.51 para 0.82, enquanto a acurácia subiu de 69.50% para 75.69%, indicando uma melhoria significativa na capacidade de encontrar a resposta certa. No COVID-QA, o Rouge-L passou de 0.20 para 0.41, mostrando que o modelo com ajuste fino é muito mais eficaz na geração de respostas em contextos complexos e específicos como questões relacionadas à COVID-19. Para o *dataset* BoolQ, a acurácia aumentou de 71.46% para 73.42%, o que demonstra uma leve, porém consistente, melhoria na compreensão de perguntas booleanas. O modelo generativo treinado no conjunto de dados CLAP-NQ apresenta *overfit*, o que o leva a gerar uma maior quantidade de respostas classificadas como *unanswerable*. Esse comportamento cria uma relação inversa e um impacto direto entre as métricas de acurácia e ROUGE-L: enquanto a acurácia para identificar perguntas sem resposta é alta devido a essa especialização, a pontuação ROUGE-L é prejudicada porque o modelo também classifica erroneamente perguntas que de fato têm resposta, reduzindo a qualidade

geral da geração. Esse comportamento sugere a necessidade de incorporar métricas adicionais à acurácia e ao ROUGE-L para poder avaliar de forma mais precisa a qualidade das respostas geradas para esse tipo de pergunta.

De forma geral, observa-se que o desempenho nos domínios específicos (TeleQnA e COVID-QA) foi superior em comparação com os *datasets* de domínio aberto (BoolQ e CLAP-NQ), o que indica que o ajuste fino é particularmente efetivo quando aplicado a contextos especializados. Embora o ajuste fino do modelo generativo utilizando passagens contribua para a melhoria do seu conhecimento, seu verdadeiro potencial se revela quando é combinado com o componente recuperador.

5.4. Análise sobre os Recursos Computacionais e Estatísticas de Treinamento

O uso da GPU durante os treinamentos foi intensivo, com picos sustentados entre 80% e 100% de utilização. O ajuste fino do modelo bge-small teve duração de aproximadamente 400 segundos por *dataset*. Já o treinamento do modelo phi-2 foi mais demorado, com o experimento mais longo totalizando 100 minutos. As Figuras 5 e 6 apresentam, respectivamente, as estatísticas completas de uso da GPU durante o processo de ajuste fino dos modelos phi-2 e bge-small-en-v1.5.

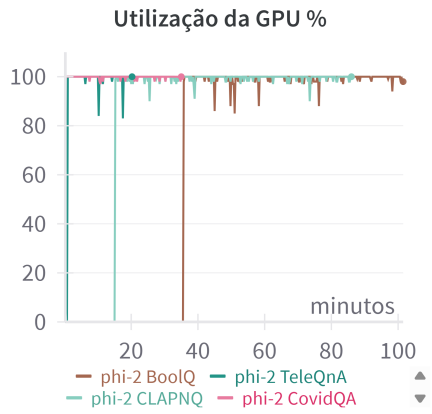

Figura 5. Uso da GPU para o ajuste fino do modelo phi-2.

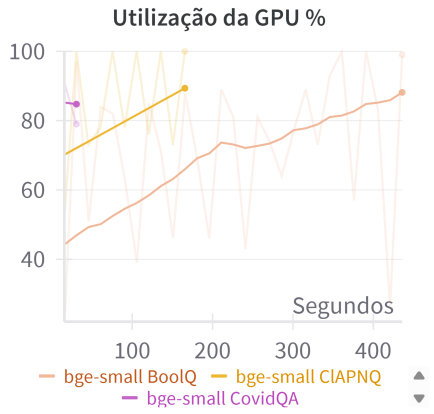

Figura 6. Uso da GPU para o ajuste fino do modelo bge-small.

6. Discussão

Os resultados obtidos neste estudo demonstraram de que a aplicação de ajuste fino no componente gerador do *framework* RAG pode trazer ganhos significativos de efetividade em tarefas de QA orientadas a domínios específicos. A abordagem de RAG-FT superou as demais abordagens em todas as métricas analisadas em todos os conjunto de dados utilizados. Tais ganhos reforçam a hipótese de que a adaptação supervisionada do modelo generativo aos padrões linguísticos e semânticos do domínio em questão contribui para a geração de respostas mais precisas e contextualmente apropriadas.

O impacto desses achados é relevante para o avanço das aplicações de LLMs em contextos especializados, como domínios jurídicos, científicos e técnicos. Ao permitir que modelos de geração condicionados a recuperação possam ser ajustados com dados específicos, a abordagem RAG-FT amplia o potencial de uso prático de sistemas de QA

em instituições públicas, organizações de pesquisa e ambientes corporativos, onde a precisão semântica é crítica.

Adicionalmente, nossa proposta oferece um caminho viável para contornar os custos computacionais elevados do ajuste fino completo de modelos de linguagem, ao focar apenas no componente gerador e manter a arquitetura de recuperação modular e reutilizável.

Apesar dos resultados promissores, algumas limitações devem ser consideradas. Em primeiro lugar, o processo de ajuste fino requer um volume representativo de exemplos anotados no domínio-alvo, o que pode não estar prontamente disponível em todos os contextos. Em segundo lugar, o estudo concentrou-se em tarefas de QA com formato de resposta curta e supervisionada; a generalização da abordagem para tarefas abertas, conversacionais ou com múltiplos turnos ainda precisa ser investigada. Embora os resultados em métricas objetivas tenham sido superiores, uma avaliação qualitativa mais aprofundada com especialistas do domínio poderia enriquecer a análise da qualidade semântica das respostas geradas.

Trabalhos futuros visam explorar otimizações adicionais para o sistema, com destaque para a incorporação de técnicas avançadas no contexto de RAG, como o reranqueamento das passagens recuperadas. Planejamos avaliar a efetividade da abordagem proposta em novos domínios. Outro caminho promissor de investigação envolve a realização de ajustes finos utilizando dados puramente sintéticos, com o objetivo de examinar a viabilidade e os impactos dessa estratégia em cenários com limitação de dados anotados manualmente.

7. Conclusão

Esta investigação desenvolveu e avaliou uma proposta de ajuste fino do modelo Retrieval-Augmented Generation (RAG), adaptada para tarefas de *Question Answering* (QA) em múltiplos domínios. A metodologia proposta baseia-se no ajuste sequencial do modelo de embeddings (bge-small-en-v1.5) e do modelo generativo (phi-2), com o objetivo de otimizar tanto a etapa de recuperação quanto a geração de respostas. Os resultados obtidos demonstraram melhorias significativas na precisão da recuperação de documentos e na qualidade das respostas geradas. A proposta superou todas as demais abordagens avaliadas, evidenciando um avanço expressivo no desempenho geral do sistema. Cada componente ajustado contribuiu de maneira substancial para os ganhos observados, destacando os benefícios da sinergia decorrente do treinamento sequencial. Esses achados destacam o potencial de sistemas baseados em RAG para aplicações em domínios especializados, nos quais a integração de conhecimento específico e a adaptação de modelos pré-treinados são essenciais para reduzir a lacuna entre soluções de inteligência artificial genéricas e abordagens orientadas a contextos específicos.

Agradecimentos

O presente estudo recebeu apoio financeiro da Petróleo Brasileiro S.A. (Petrobras) por meio do projeto de pesquisa e desenvolvimento (P&D) “Aplicação de Large Language Models (LLMs) para o monitoramento online de processos industriais”, em parceria com a Universidade Estadual de Campinas.

Referências

[3rd Generation Partnership Project (3GPP) 2023] 3rd Generation Partnership Project (3GPP) (2023). 3gpp specifications - release 18. https://portal.3gpp.org/#/55934-releases.

[Clark et al. 2019] Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

[Fan et al. 2024] Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. (2024). A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

[Gao et al. 2024] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey.

[Gichamba et al. 2024] Gichamba, A., Idris, T. K., Ebiyau, B., Nyberg, E., and Mitamura, T. (2024). Colbert retrieval and ensemble response scoring for language model question answering. Accepted at the 2024 IEEE Global Communications (GLOBECOM) Workshops.

[Han et al. 2024] Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

[Hu et al. 2022] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

[Johnson et al. 2021] Johnson, J., Douze, M., and Jégou, H. (2021). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

[Karpukhin et al. 2020] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

[Kwiatkowski et al. 2019] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

[LangChain 2023] LangChain (2023). How to split by token in langchain. https://python.langchain.com/docs/how_to/split_by_token/. Accessed: 2025-01-01.

[Lewis et al. 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

[Maatouk et al. 2023] Maatouk, A., Ayed, F., Piovesan, N., Domenico, A. D., Debbah, M., and Luo, Z.-Q. (2023). Teleqna: A benchmark dataset to assess large language models telecommunications knowledge.

[Maslej et al. 2023] Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Ngo, H., Niebles, J. C., Parli, V., Shoham, Y., Wald, R., Clark, J., and Perrault, R. (2023). Artificial intelligence index report 2023.

[Microsoft Research 2023] Microsoft Research (2023). Phi-2: The surprising power of small language models. `https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/`. Accessed: 2025-01-01.

[Möller et al. 2020] Möller, T., Reina, A., Jayakumar, R., and Pietsch, M. (2020). COVID-QA: A question answering dataset for COVID-19. In Verspoor, K., Cohen, K. B., Dredze, M., Ferrara, E., May, J., Munro, R., Paris, C., and Wallace, B., editors, *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

[Muennighoff et al. 2023] Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). MTEB: Massive text embedding benchmark. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

[Piovesan et al. 2024] Piovesan, N., De Domenicoo, A., and Ayed, F. (2024). Telecom language models: Must they be large? In *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6.

[Reimers and Gurevych 2019] Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

[Rosenthal et al. 2025] Rosenthal, S., Sil, A., Florian, R., and Roukos, S. (2025). CLAPnq: Cohesive long-form answers from passages in natural questions for RAG systems. *Transactions of the Association for Computational Linguistics*, 13:53–72.

[Siriwardhana et al. 2023] Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., and Nanayakkara, S. (2023). Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

[Xiao et al. 2023] Xiao, S., Liu, Z., Zhang, P., and Muennighoff, N. (2023). C-pack: Packaged resources to advance general chinese embedding.

[Xiao et al. 2024] Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., and Nie, J.-Y. (2024). C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

[Zhou et al. 2024] Zhou, H., Hu, C., Yuan, Y., Cui, Y., Jin, Y., Chen, C., Wu, H., Yuan, D., Jiang, L., Wu, D., Liu, X., Zhang, C., Wang, X., and Liu, J. (2024). Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities.