Generating Personalized Feedback from Professional Profile Tests Using RAG: A Case Study with MAPER

Erik Jhones F. Nascimento
Federal University of Ceara (UFC)
Fortaleza, Brazil
erikjhonesf@gmail.com

Jessyca A. Bessa
Federal Institute of Ceara (IFCE)
Maranguape, Brazil
jessyca@ nuven.ifce.edu.br

Maryane C. Lima
Federal Institute of Ceara (IFCE)
Maracanaú, Brazil
maryane.castro993@gmail.com

Igor Rafael S. Valente
Federal Institute of Ceara (IFCE)
Maracanaú, Brazil
igor@ifce.edu.br

Saulo M. Maia
Federal University of Ceara (UFC)
Fortaleza, Brazil
saulo@ufc.alu.br

Maria Lúcia Rodrigues Correa
Getulio Vargas Foundation (FGV)
Belo Horizonte, Brazil
contato@marialuciarodrigues.com

***Abstract*—Professional profile mapping tests play a crucial role in identifying talents and guiding employee development. However, the growing demand for these evaluations has resulted in increased workload and saturation for the professionals conducting them. Large Language Models (LLMs), when properly integrated, offer promising support for automating report generation and diagnostic feedback. In this study, we explore the use of Retrieval-Augmented Generation (RAG) combined with a structured database of psychological feedback to automatically generate personalized reports for the MAPER test. Our method retrieves example feedback based on specific combinations of competencies and scores, then prompts an LLM to generate fluent and semantically rich output. Experimental results demonstrate that our approach achieves up to 77% semantic similarity compared to expert-generated feedback, indicating strong potential for real-world applications.**

***Index Terms*—retrieval augmented generation, large language model, professional profile mapping test**

I. Introduction

Professional profile mapping tests are essential tools in people management and professional development. They help identify talents, guide individual growth, and improve organizational efficiency.

These tests are administered by psychologists specialized in professional assessment and generally require the evaluated individual to answer a series of questions or perform cognitive exercises. Although each result is unique to the individual being assessed, there is a tendency for psychologists to follow consistent assessment patterns and deliver results based on predefined models. These patterns may arise from the test's inherent structure or from the psychologist's accumulated experience in administering it.

In parallel, Large Language Models (LLMs) [1; 2] have emerged as standard tools for tasks involving text generation, document translation, diagnostic assistance, academic text explanation, and assessment correction. For example, although some degree of human supervision is still necessary for critical tasks, these models are already responsible for generating approximately 14% of United Nations press releases[3].

This raises a natural question: *Can LLMs be used to generate feedback from professional profile tests in a way that remains individual and personalized?* In principle, there are many parametric methods designed to efficiently input new knowledge into LLMs and enable them to work with new tasks — e.g., Low-Rank Adaptation (LoRa) [4; 5; 6], Quantized Low-Rank Adaptation (QLoRa) [7], Odds Ratio Preference Optimization (ORPO) [8]. These approaches aim to fine-tune LLMs using minimal computational resources while preserving accuracy.A drawback of these methods is the lack of rapid and low-cost parameter updates, coupled with the need for considerable technical expertise for proper implementation.

As an alternative to parametric fine-tuning methods, the Facebook AI Research (FAIR) team created the Retrieval Augmented Generation (RAG) method [9; 10]. RAG is a powerful technique that enhances language models by integrating them with external knowledge bases. RAG addresses a key limitation of models: models rely on fixed training datasets, which can lead to outdated or incomplete information. RAG takes advantage of the LLM's prior knowledge and keeps its parameters frozen. All the knowledge needed to answer a query is passed directly via prompt in an approach known as hard-prompt. The LLM is responsible for combining its prior knowledge with the knowledge provided to answer the query.

In this paper, we propose to generate new feedback from professional profile tests using RAG. Our approach was focused on the MAPER test. Among the tests available on the market, MAPER stands out for being focused on the Brazilian reality, taking into account cultural aspects and the specific demands of the local job market.

In summary, our **contributions** are:

- Development and structuring of a comprehensive database

containing feedback samples from professional profile tests.
- Implementation of a retrieval mechanism tailored to match competency-score combinations.
- Design of an optimized prompt to guide LLM responses with natural and objective language.
- Definition of an end-to-end methodology for transforming test scores into personalized feedback.
- Validation through experiments, showing semantic similarity scores of up to 77% compared to expert feedback.

The remainder of this article is divided as follows: in section II we provide a background about LLMs, fine-tuning approachs and RAG, on section III we show some works that had used RAG to solve real applications, section IV we show how to mount our database, section V we show our RAG methodology, section VI we present our experiments, and section VII we show our conclusions.

II. Background

In this section, we define some key concepts for a full understanding of this work.

Large Language Models (LLMs). Typically refer to language models [1; 2] based on the transformer architecture [11]. Some of the most well-known architectures for developing and training LLMs include *Generative Pre-trained Transformers* (GPT), *Language Model for Dialogue Applications* (LaMDA), *Pathways Language Model* (PaLM), and AnthropicLM (used by Claude). LLMs such as ChatGPT, Gemini, LLaMA, Claude, and O1 Mini typically have hundreds of billions of parameters and are trained on massive textual datasets. Their main advantages include a strong ability to understand natural language and solve complex problems via text generation.

Fine-tuning. With the rapid growth in the number of parameters in state-of-the-art LLMs, fine-tuning for each downstream task has become prohibitively expensive in terms of time and resources. During standard fine-tuning, a parameter matrix ΔW is created with the same dimensions as the original parameter matrix W of the model. After the *backpropagation* step, W is updated to $W = W + \Delta W$. The goal of parameter-efficient fine-tuning (PEFT) [12] is to adapt models to new tasks by updating only a small number of (possibly new) parameters. Dominant PEFT approaches include Low-Rank Adaptation (LoRA) [4; 5; 6], soft-prompt methods [13], partial refinement and masking strategies [14; 15], and adapter-based techniques [16; 17]. Like traditional fine-tuning methods, PEFT approaches also present some drawbacks, such as the need for large volumes of annotated data and the risk of overfitting.

Retrieval-Augmented Generation (RAG). [9; 18; 19] enhances language models by augmenting prompts with relevant, factual, and updated information. Sources for RAG may include web searches or private databases. The RAG paradigm is currently categorized into three stages: Naive RAG, Advanced RAG, and Modular RAG. Naive RAG represents the earliest methodology, following a traditional process of indexing, retrieval, and generation, commonly referred to as the ”Retrieve-Read” framework. Advanced RAG improves upon this by enhancing retrieval quality through pre- and post-retrieval strategies. It refines indexing using sliding windows, fine-grained segmentation, and metadata, while optimizing the retrieval process. Modular RAG increases adaptability by refining components through similarity search techniques and retriever fine-tuning. It introduces restructured modules and pipelines to address specific challenges, supporting both sequential processing and end-to-end training. Although distinct, Modular RAG builds upon the earlier stages, marking a progression in the paradigm.

Professional Mapping Profile Test (MAPER). The MAPER test is based on the inventory developed by Dr. Max Martin Kostick in 1966, known as PAPI (*Personality and Preference Inventory*) [20]. In Brazil, MAPER was developed and adapted by psychologist Maria Lúcia Rodrigues to better reflect the Brazilian context, incorporating cultural aspects and specific demands of the local job market. MAPER uses an ipsative scale [20], which requires respondents to choose the statements that ”most resemble me” and those that ”least resemble me” from a set of equally desirable items. This format is considered more resistant to manipulation, as it forces the respondent to choose between two distinct behavioral traits.

The test evaluates 20 competencies: **C1** Planning Ability, **C2** Organizational Ability, **C3** Coaching Leadership, **C4** Motivational Leadership, **C5** Communication Style, **C6** Decision Making, **C7** Delegation Ability, **C8** Time Management, **C9** Workload, **C10** Creative Potential, **C11** Ability to Deal with Unforeseen Events, **C12** Change Management, **C13** Relationship with Superiors, **C14** Conflict Management, **C15** Emotional Control, **C16** Trust Relationships, **C17** Group Relationships, **C18** Personal Image, **C19** Vital Tone, and **C20** Need for Achievement. Each competency is scored from 0 to 10, and the score is classified as ideal, below ideal, above ideal, subideal (1 point below ideal), or superideal (1 point above ideal).

III. Related Works

While there are no previous studies specifically focused on professional profile mapping, several works have explored the application of Retrieval-Augmented Generation (RAG) in other domains such as medicine, ophthalmology, business, and education. A. A. Trindade [21] applied RAG in conjunction with computer vision techniques to classify patient histories related to cardiovascular risk. Passinato et al. [22] developed a chatbot to answer questions about eye health, integrating RAG with a textual database on ophthalmology. Superbi et al. [23] used RAG to enhance the accuracy of GPT in answering mathematics questions from the *Exame Nacional do Ensino Médio* (ENEM), a national high school exam in Brazil. Souza et al. [24] proposed a novel approach for retrieving and generating responses based on structured table data using RAG. Medeiros et al. [25] applied RAG to enable a large language model to answer domain-specific questions about legal contracts.

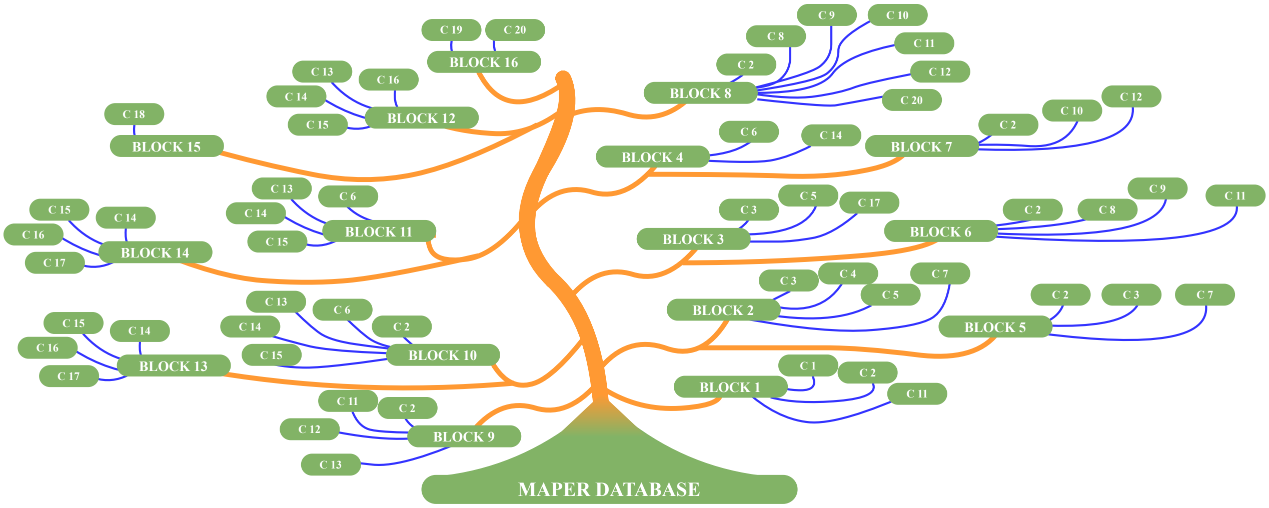

Fig. 1: Our database have 16 blocks (jsons), where each block contains feedbacks for each combination of competences associated with them. e.g. in block 1, for each combination of competences 1, 2 and 11, there are several associated feedbacks.

IV. Database Construction

We used 680 audio files recorded by a psychologist responsible for administering the MAPER test. These audio files contain real test feedback, including the test introduction, detailed comments on each competency (including their interrelations when applicable), a section explaining the client's overall scores, and a closing statement.

Initially, we transcribed the audio files using OpenAI's Whisper tool [26]. All transcripts were reviewed to identify recurring patterns and correct potential transcription errors. We found that the feedback related to the 20 competencies could be organized into **16 competency blocks**, each focused on specific competencies and their interrelationships. This segmentation enabled the creation of a standardized feedback database, which can be used to train and refine machine learning models for generating more precise and consistent feedback. In Figure 1, we illustrate how each block is associated with its respective competencies.

This structure improves consistency and allows for better alignment between related competencies, leading to more insightful and context-aware performance analysis.

To facilitate rapid retrieval of personalized feedback, we stored feedback examples for each competency block in **16 JSON files**. These files contain different combinations of competency pairs and score classifications (e.g., `"c1_below_c2_above"` for Competencies 1 and 2). Each JSON file includes a set of pre-written feedback templates tailored to specific combinations of competencies and their score classes.

Below is an example of a real entry from Block 1:

"c1_abaixo_c2_abaixo": "Planejamento, competência 1. Aqui está indicando que você vai direto para a ação. Você faz acontecer depressa demais e às vezes pode gerar retrabalho. E como a organização, competência 2, você não é muito organizado, você pode se perder por falta de detalhes. Você pode se perder por falta de detalhes e gerar muito retrabalho para você mesmo."

V. Feedback Generation Using RAG

In this section, we detail the key components involved in the generation of feedback using a RAG-based approach applied to the MAPER professional profile mapping test.

To illustrate, suppose we want to generate automatic feedback for a person named **Hermione Granger**. Assume Hermione completed the MAPER test and received the following scores: $C_1 = 7, C_2 = 3, C_3 = 3, C_4 = 3, C_5 = 5, C_6 = 5, C_7 = 6, C_8 = 5, C_9 = 6, C_{10} = 4, C_{11} = 7, C_{12} = 7, C_{13} = 3, C_{14} = 5, C_{15} = 4, C_{16} = 4, C_{17} = 7, C_{18} = 6, C_{19} = 3, C_{20} = 7$.

The first step, from the block **Generation Functions** in Figure 2, is responsible for classifying the scores according to the five categories introduced in Section II and assembling the corresponding competency/class combinations. For example, Hermione's profile may be represented as: `C1_superideal_C2_subideal`, `C3_below_C4_below`, etc.

The next step retrieves the relevant feedback entries (chunks) from the database based on the competency/class combinations. As described in Section IV, each JSON file uses a key-value structure that allows efficient retrieval without the need for similarity search algorithms. This approach simplifies the process significantly. But what happens if a competency/class combination is not found in the database? This issue does not occur because all possible combinations were pre-mapped, as illustrated in Figure 1.

The retrieved feedback chunks are then concatenated with a predefined prompt to form the input query for the LLM. We use the Zero-shot prompting technique [], which provides

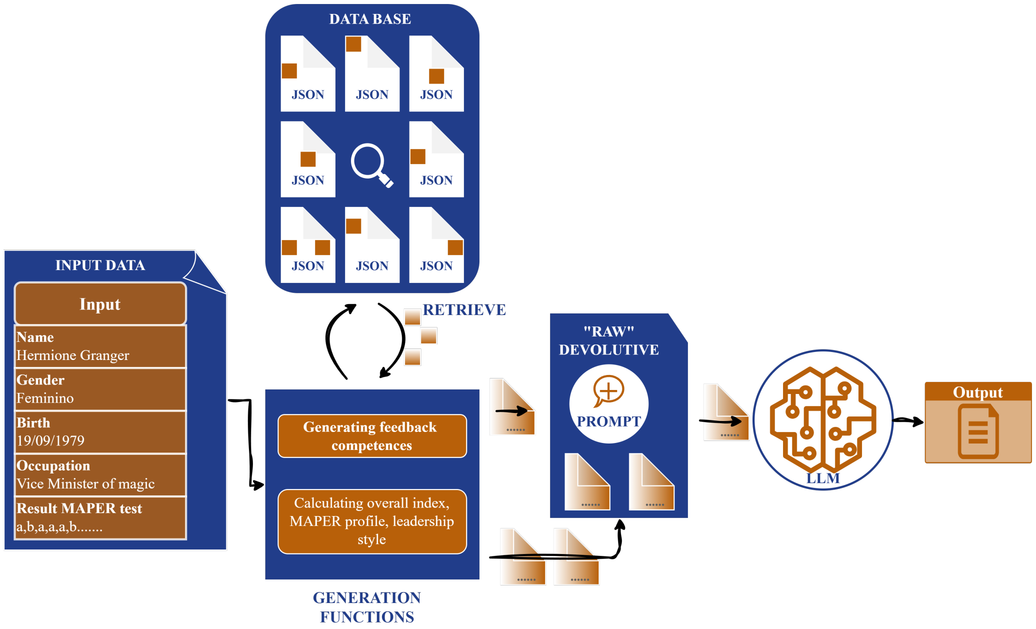

Fig. 2: Flowchart of our approach. The input goes through a step of classifying the grades and assembling the class/competency combinations. Then the chunks are retrieved from our database. The returned chunks plus the prompt enter the LLM, which finally returns the new feedback.

the model with only a task description and requests the desired output. To determine the optimal prompt, we conducted prompt engineering experiments with several candidates and selected the one that consistently produced semantically accurate responses that aligned with the psychologist's standard feedback style.

The final optimized prompt is shown below:

> *Trabalhe com o Português do Brasil. Você está comentando os resultados de um teste (não comente isso no retorno) com um usuário do gênero x em um áudio. Não se apresente e nem comente explicitamente meus dados de gênero, ano de nascimento e geração.*
> *A seguir estão suas falas divididas por diferentes momentos do áudio. Una esses trechos em um único texto contínuo, utilizando conectivos para garantir a fluidez e a coerência. Remova todos os cumprimentos e despedidas. Reescreva, alterando algumas palavras e ajustando levemente a estrutura para manter o estilo natural e conversacional. Responda apenas com os fatos objetivos, sem fazer suposições ou interpretações sobre o estado emocional ou mental do usuário. Apenas apresente informações baseadas nos dados fornecidos. Além disso, insira comentários personalizados com base nos seguintes dados do usuário (mas não exagere, para que não haja repetições): sexo xxx e ano de nascimento yyyy. É uma conversa, então, refira-se ao usuário por "você"; nunca use a terceira pessoa. Nunca diga o que você é ou o que acha, nem se refira a "eu sou..." e nunca diga como o usuário se sente ou deveria se sentir. Retorne sem nenhuma informação adicional ou explicações. Elimine repetições e troque por sinônimos, especialmente conectivos. Não adicione adjetivos como: interessante, legal, bom, ruim, ótimo, excelente, maravilhoso, etc.*

We provide this prompt to the LLM, which then returns the generated feedback accordingly.

VI. Experiments

In this section, we detail the experiments conducted for the generation of new feedback using RAG in combination with three different LLMs. We also present the results obtained and provide corresponding analysis.

Baselines. Since running LLMs locally requires high computational resources (e.g., VRAM, GPU, RAM), we used the GroqCloud API service. This API offers several text-to-text models free of charge, with a limit on requests per second. Among the models available via Groq, we evaluated three: `llama3-70b-8192`, `llama3-8b-8192`, and `mistral-saba-24b`. We accessed the Groq API using the LangChain Python library [27].

Metrics. To evaluate our generation method, we adopted a quantitative approach based on semantic similarity metrics. ROUGE-1 measures the overlap of unigrams (individual words) between the generated and reference texts, while ROUGE-2 evaluates bigrams (pairs of consecutive words), offering a more refined assessment of textual fluidity [28]. ROUGE-L, in contrast, is based on the Longest Common Subsequence (LCS), capturing structural similarities by identifying the longest shared sequence of words.

BERTScore [29] uses embeddings generated by transformer-based models, such as BERT, to perform semantic comparisons between texts. This enables the evaluation of similarity even when different lexical choices are made to convey the same meaning, thereby addressing the limitations of purely token-based metrics.

For each LLM, we generated 40 different feedback samples using our RAG methodology. We then computed the precision, recall, and F1-score using both ROUGE and BERTScore metrics.

A. Results and Discussion

As shown in Table I, `llama3-70b-8192` consistently outperformed the other evaluated models across all metrics, particularly in the F1-score and BERTScore evaluations. It achieved up to 6% higher F1-score compared to the `mistral-saba-24b` model, and reached a BERTScore of approximately 77%, indicating a high degree of semantic similarity between the generated and reference feedback.

This performance suggests that Meta's `llama3-70b-8192` model may have been pre-trained on datasets with greater coverage of psychological or behavioral assessment language—potentially including examples related to the PAPI framework, which underlies the MAPER test. This hypothesis is further supported by the observation that even the smaller `llama3-8b-8192` model performed competitively in BERTScore, showing that the LLaMA architecture is particularly well-suited for this domain.

In contrast, while the `mistral-saba-24b` model demonstrated reasonably strong performance in ROUGE metrics, it lagged in capturing semantic nuances as indicated by its lower BERTScore values. This suggests that although Mistral can replicate lexical structures, it may struggle to reproduce deeper contextual and psychological coherence in feedback generation tasks.

We opted not to include generation time as a reported metric, since all models were executed in a cloud environment (GroqCloud), where latency can vary depending on external factors. Furthermore, in a real-world setting, the generation time for individual psychological feedback is typically non-critical, as accuracy and personalization are prioritized over speed.

In summary, the `llama3-70b-8192` model demonstrated the best overall performance for the task of generating personalized feedback using our RAG pipeline. Its ability to produce coherent, context-aware, and semantically aligned responses makes it a highly suitable candidate for production-level deployment in professional profile evaluation systems.

VII. Conclusion

This is the first work on the application of Retrieval Augmented Generation (RAG) to generate feedback for the professional profile mapping test.

TABLE I: Comparison of the generative capacity of different LLMs using RAG with our database. Here we report the results for the metrics ROUGE 1, 2 and L, and BertScore..

	llm	**precision**	**recall**	**f1-score**
rouge1	llama70b	**63.7** ± 3.5	**68.6** ± 3.9	**66.1** ± 2.3
	llama8b	42.1 ± 6.4	42.6 ± 5.1	42.3 ± 5.6
	mistral	57.4 ± 6.3	48.9 ± 5.2	53.1 ± 4.7
rouge2	llama70b	32.5 ± 2.8	**35.4** ± 3.1	**33.7** ± 2.8
	llama8b	18.7 ± 7.4	21.3 ± 6.2	19.8 ± 5.2
	mistral	**35.8** ± 3.1	29.5 ± 2.8	31.3 ± 2.4
rougel	llama70b	**40.3** ± 3.4	**43.4** ± 1.6	**41.8** ± 2.6
	llama8b	25.9 ± 4.8	24.1 ± 3.7	24.4 ± 4.2
	mistral	39.4 ± 2.9	41.8 ± 1.9	39.5 ± 2.7
bertSc	llama70b	**77.1** ± 1.5	**74.9** ± 1.6	**76.6** ± 1.6
	llama8b	60.3 ± 1.8	62.6 ± 1.4	61.3 ± 1.5
	mistral	70.7 ± 1.9	70.3 ± 2.5	70.4 ± 2.2

Our approach consists of first processing several real audios, recorded by a psychologist, containing feedbacks from 20 competences. With this, we created a database divided into 16 blocks, where each block contains examples of feedbacks associated with each combination of competences. And finally, we applied the RAG method to generate new feedbacks based on our database.

Our experiments demonstrate that feedback generated through our RAG-based methodology, particularly using `llama3-70b-8192`, exhibits up to 77% semantic similarity with expert-generated feedback. These findings validate the effectiveness of combining structured domain-specific data with advanced LLMs for producing personalized, context-aware responses in professional assessment settings.

VIII. Acknowledegment

This research is supported by grants from the NUVEN (computer vision and engineering laboratory), the innovation hub of the Federal Institute of Ceará, CNPq (Brazilian council for scientific and technological development) and CAPES (coordination for the improvement of higher education personnel). We are thankful for generous support by EMBRAPII (Brazilian company for industrial research and innovation) for the financial support.

References

[1] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[3] W. Liang, Y. Zhang, M. Codreanu, J. Wang, H. Cao, and J. Zou, "The widespread adoption of large language model-assisted writing across society," *arXiv preprint arXiv:2502.09747*, 2025.

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[5] A. Edalati, M. Tahaei, I. Kobyzev, V. P. Nia, J. J. Clark, and M. Rezagholizadeh, "Krona: Parameter efficient tuning with kronecker adapter," *arXiv preprint arXiv:2212.10650*, 2022.

[6] M. Valipour, M. Rezagholizadeh, I. Kobyzev, and A. Ghodsi, "Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation," *arXiv preprint arXiv:2210.07558*, 2022.

[7] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[8] J. Hong, N. Lee, and J. Thorne, "Orpo: Monolithic preference optimization without reference model," *arXiv preprint arXiv:2403.07691*, 2024.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[10] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2312.10997

[11] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[12] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment," *arXiv preprint arXiv:2312.12148*, 2023.

[13] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[14] M. Zhao, T. Lin, F. Mi, M. Jaggi, and H. Schütze, "Masking as an efficient alternative to finetuning for pretrained language models," *arXiv preprint arXiv:2004.12406*, 2020.

[15] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.

[16] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[17] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J.-W. Low, L. Bing, and L. Si, "On the effectiveness of adapter-based tuning for pretrained language model adaptation," *arXiv preprint arXiv:2106.03164*, 2021.

[18] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," *arXiv preprint arXiv:1911.00172*, 2019.

[19] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

[20] M. Kostick, "Perception and preference inventory (papi)," *Personality and Individual Differences*, vol. 8, no. 4, pp. 459–470, 1987, disponível em: URL, A.

[21] A. A. Trindade, "Classificação de eletrocardiograma com integração de modelos de linguagem grande e técnicas de geração aumentada de recuperação para suporte à decisão médica," in *Escola Regional de Informática de Mato Grosso (ERI-MT)*. SBC, 2024, pp. 7–12. [Online]. Available: https://doi.org/10.5753/eri-mt.2024.245867

[22] E. B. Passinato, W. S. Rios, and A. R. Galvão Filho, "Integração de modelos de linguagem e rag na criação de chatbots oftalmológicos," in *Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*. SBC, 2024, pp. 354–365. [Online]. Available: https://doi.org/10.5753/sbcas.2024.2228

[23] J. Superbi, H. Pinto, E. Santos, L. Lattari, and B. Castro, "Enhancing large language model performance on enem math questions using retrieval-augmented generation," in *Brazilian e-Science Workshop (BreSci)*. SBC, 2024, pp. 56–63. [Online]. Available: https://doi.org/10.5753/bresci.2024.243977

[24] E. A. de Souza, P. F. da Silva, D. Gomes, V. Batista, E. Batista, and M. Pacheco, "Tablerag: A novel approach for augmenting llms with information from retrieved tables," in *Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)*. SBC, 2024, pp. 182–191. [Online]. Available: https://doi.org/10.5753/stil.2024.245371

[25] A. S. de Medeiros, C. Cavalcante, J. Nepomuceno, L. Lago, N. Ruberg, and S. Lifschitz, "Contrato360: uma aplicação de perguntas e respostas usando modelos de linguagem, documentos e bancos de dados," in *Simpósio Brasileiro de Banco de Dados (SBBD)*. SBC, 2024, pp. 155–166. [Online]. Available: https://doi.org/10.5753/sbbd.2024.240871

[26] T. Amorese, C. Greco, M. Cuciniello, R. Milo, O. Sheveleva, and N. Glackin, "Automatic speech recognition (asr) with whisper: Testing performances in different languages." in *S3C@ CHItaly*, 2023, pp. 1–8.

[27] R. Jay, "Introduction to langchain and llms," in *Generative AI Apps with LangChain and Python: A Project-Based Approach to Building Real-World LLM Apps*. Springer, 2024, pp. 1–38.

[28] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[29] I. J. Unanue, J. Parnell, and M. Piccardi, "Berttune: Fine-tuning neural machine translation with bertscore," *arXiv preprint arXiv:2106.02208*, 2021.