Leveraging Large Language Models for Cosmetics Product Recommendations: A Comparative Study with Dorabruschi

Tommaso Salvatori
salva358@mit.edu
MIT Sloan School of Management

Abstract

Recommender systems are crucial for personalized product suggestions, but traditional methods face limitations with sparse data, especially for small companies. This paper explores the potential of large language models (LLMs) for building a cosmetics routine recommendation system, using Dorabruschi, a small cosmetics company, as a case study. The study evaluates three LLM-based models: custom GPT-4, GPT-3.5-Turbo with Retrieval Augmented Generation (RAG), and fine-tuned GPT-3.5-Turbo. Through a qualitative assessment involving Dorabruschi specialists, the paper analyzes the performance and suitability of each approach. The results highlight the potential of LLMs in addressing cold-start problems and providing comprehensive beauty routines.

1. Introduction

Recommender systems have become essential for businesses to provide personalized product suggestions to their customers. However, traditional approaches like collaborative filtering face limitations, particularly for small companies dealing with sparse data due to limited historical interactions [5]. Large language models (LLMs) provide an intriguing alternative for building effective recommender systems.

Firstly, LLMs, such as BERT or GPT and PaLM series, hold a vast base of encoded world knowledge that can be leveraged to provide external information beyond the target domain's data [22]. Secondly, their language generation capabilities can allow customers to engage in conversational exchanges - similar to those with a store assistant - and receive justifications for their recommendations in natural language [8], enhancing interactivity and explainability. Perhaps most importantly, LLMs can alleviate cold-start scenarios through remarkable zero and few-shot recommendation capabilities [19].

However, there are still numerous challenges in using LLMs for recommender systems. Their stochastic nature can lead to hallucinations or unpredictable outputs, as seen in 2022 when Air Canada's chatbot promised an inexistent discount to its passenger, for which the company was held liable [23]. Additionally, the performance of these models is highly sensitive to the design of its input prompt, and incorporating product information and user preferences into these can be problematic. Finally, LLMs can exhibit popularity and fairness biases, favoring items that appear more frequently in their training corpus and exhibiting unfairness to social attributes like gender and race.

This research explores the potential of LLMs for building a cosmetics routine recommendation system. It will conduct the study in the context of Dorabruschi, a small cosmetics company in Florence, Italy. Dorabruschi recently relaunched its brand, and its new customers and product line have introduced a cold-start problem with little historical customer-product data. To address this, this paper will evaluate three LLM-based models: i) custom GPT-4, ii) GPT-3.5-Turbo with Retrieval Augmented Generation (RAG), and iii) fine-tuned GPT-3.5-Turbo. It will conduct a qualitative assessment involving Dorabruschi specialists to analyze the performance and suitability of each of the approaches. In doing so, it will seek to provide insights and guidelines for implementing a deployable, user-facing chatbot for Dorabruschi's product recommendations.

Suggesting a comprehensive beauty routine is non-trivial; the chatbot must first select relevant products and then ensure they work well together to form a cohesive skincare regimen. This paper contributes to the field by demonstrating the effectiveness of LLM-based recommendation systems in such a complex scenario, differentiating it from existing work focused on simpler recommendation tasks. All code used in this research is publicly available on GitHub [18].

The objectives of this research are twofold: (i) to investigate the effectiveness of LLMs in making product recommendations, and (ii) to identify the most effective LLM approach for this task within the context of Dorabruschi's specific needs. To achieve these goals, this paper is structured

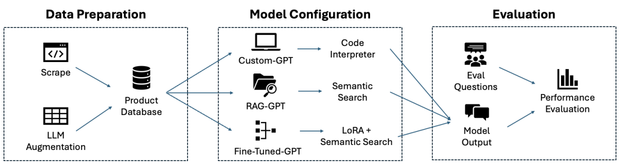

Figure 1. **Overview.** Product data is first collected by scraping the Dorabruschi website and augmented with the use of a helper LLM. Then, the three models are configured, each leveraging different techniques. Finally, the responses of the models are evaluated on a set of 25 questions created by product specialists at the company.

as follows: the "Related Work" section reviews prior studies on LLMs and their applications in recommender systems, the "Data" section details the preparation of the dataset, the "Methodology" section outlines the specific configurations of the three LLM-based models under investigation and the method used to evaluate them, the "Results" section presents the findings from the qualitative assessments conducted by Dorabruschi specialists, the "Discussion" section interprets these findings and explores their implications, and the "Conclusion" section synthesizes these findings to offer insights and actionable guidelines.

2. Related Work

Traditional recommender systems typically employ collaborative filtering (CF) and content-based methods to predict user preferences based on historical interactions and additional item metadata [10, 11]. However, the advent of LLMs like BERT and GPT has introduced new paradigms for generating recommendations.

LLMs have already been effectively utilized for sequential recommendation tasks. For example, BERT4Rec leverages bidirectional attention to model user behavior sequences, enabling more accurate predictions of future interactions [20]. Similarly, transformer-based models have been adapted to generate both recommendations and explanations, enhancing the transparency and interpretability of the recommendation process [13].

The primary advantage of using LLMs in recommender systems lies in their ability to process and generate natural language, which allows for the creation of interactive and explainable recommendations. This capability is particularly beneficial in conversational recommender systems, where LLMs can dynamically adapt to user queries and provide personalized suggestions in real-time [8, 9].

Recent studies have also explored the use of retrieval-augmented generation (RAG) to combine LLMs with retrieval mechanisms, ensuring that the recommendations are based on relevant and up-to-date information. This approach addresses the limitations of LLMs, such as hallucinations and context window restrictions, by retrieving pertinent data to inform the generation process [12, 24].

Moreover, fine-tuning LLMs on domain-specific datasets has been shown to improve their performance in recommendation tasks. By incorporating domain-specific knowledge, fine-tuned models can better understand the nuances of user preferences and provide more accurate and relevant recommendations [24].

Overall, the integration of LLMs into recommender systems represents a promising direction for enhancing recommendation quality and user satisfaction. While challenges such as computational costs and the risk of hallucinations remain, ongoing research continues to explore effective strategies to mitigate these issues and fully leverage the capabilities of LLMs in recommendation contexts.

3. Data

To ensure the three approaches produce useful outputs, it is necessary to compile an accurate and exhaustive catalog of Dorabruschi products. To do so, a Python script was developed that used the Beautiful Soup HTML parser to systematically navigate through the company website and scrape all relevant product information. The data extracted from 60 product pages included the product title, description, usage instructions, ingredients, skin type, price, and quantity.

The first challenge encountered was that the product catalog extracted from the Dorabruschi website exhibited inconsistencies, particularly with the phrasing of feature values across different products. This variability poses a challenge for automated systems that rely on uniformity for optimal performance. In particular, it would have been detrimental to the custom GPT-4 model, which uses a built-in code interpreter, as will be discussed later. To address this, ChatGPT (GPT-4) was instructed to standardize the features by transforming each product attribute into a set of predefined categorical values based on the most similar match, ensuring uniformity. For instance, 'skin type' descriptors such as 'for dry to very dry skin' or 'suitable for all types of skin including sensitive' were standardized to fixed categories like 'Dry' and 'All Skin Types'.

In addition, since a large part of the LLMs' task consists of mapping customer queries to relevant products, additional features were introduced to enhance the utility of the dataset for each task. ChatGPT was once again used to populate new columns such as 'intended skin concern' and 'skin benefits' based on the contextual interpretation of existing data columns like 'description' and 'ingredients'. For example, a product containing salicylic acid in the ingredients was labeled as 'Acne' for 'intended skin concern' and as 'Purifying' for 'skin benefit'.

This two-staged approach of standardization and expansion served to streamline the dataset into a more structured format and enrich it with meaningful attributes that directly reflect customer language (see Figure 2), thereby enhancing the relevance and accuracy of LLM product recommendations. Finally, the product catalog was verified by a Dorabruschi specialist to ensure information correctness prior to building each model. These data preparation steps are an example of the general approach of using 'helper LLMs' to aid in various aspects of a machine learning pipeline [15].

title	description	usage_instructions	properties	ingredients	product_type	benefits	intended_concerns	skin_type
ACE 10% multivitamin concentrate	This rapidly absorbed concentrated treatment e...	Apply a few drops of concentrate in the mornin...	Anti-wrinkle, Antioxidant, Illuminating	Aqua [water], Glycerin, Tocopheryl acetate, Pr...	Serum	Wrinkle	Aging, Dullness, Wrinkles	All Skin Types
Revitalizing multivitamin cream	Cream with a velvety and light texture, design...	Apply in the morning and/or in the evening to ...	Anti-wrinkle, Antioxidant, Illuminating	Aqua [water], Glycerin, Cetyl alcohol, Capryli...	Moisturizer	Wrinkle	Aging, Dullness, Wrinkles	All Skin Types
Smoothing renewing cream	Cream with a velvety and light texture, it is ...	Apply in the evening to perfectly cleansed ski...	Anti-wrinkle, Renewing, Illuminating	Aqua [water], Peg-6 stearate, Glycolic acid, C...	Moisturizer	Wrinkle	Aging, Dullness, Wrinkles	All Skin Types

Figure 2. **Data sample**. Sample of the final product dataset, consisting of 60 products and 13 features.

4. Methodology

With the product database prepared and standardized, the next phase involved implementing the LLMs to solve the recommendation task. Given the comprehensive nature of the product catalog and the large variety of possible customer queries, prompt engineering methods such as zero-shot or few-shot learning were deemed inadequate. These methods risk model hallucination – recommending plausible-sounding products that don't actually exist – due to their inability to access and process the extensive product details during the generation process.

Moreover, directly embedding the entire product catalog into the model's prompt was impractical (though this may not be a limitation for much longer as the availability of models with larger context windows increases). The catalog exceeds 20,000 tokens, far surpassing the 16,385 token context window limit of GPT-3.5-Turbo. To address these challenges, three distinct LLM-based approaches were selected, in increasing order of technical complexity:

1. **Custom GPT:** This initial, no-code approach leverages OpenAI's GPTs, which are custom versions of ChatGPT that combine instructions and extra knowledge [1].

2. **RAG-GPT (Retrieval Augmented Generation with GPT-3.5-Turbo):** This method generates embeddings for each product and uses a semantic search mechanism to retrieve and rank products that are most relevant to the user's queries.

3. **Fine-tuned GPT:** This approach involves a GPT-3.5-Turbo model that has been fine-tuned with over 100 tailored prompt-completion pairs. This model is also supplemented with a semantic search component to further mitigate risks of hallucination.

The choice of GPT-3.5-Turbo as the baseline for the second and third approaches was influenced by its cost-effectiveness – it costs approximately 1/20th the price of GPT-4 Turbo [3] – along with its speed superiority and the absence of complex reasoning requirements for this task.

4.1. Custom GPT

GPTs are custom versions of ChatGPT tailored towards a specific task. Creating one, like the 'Dorabruschi Beauty Routine Advisor', involves configuring the model with precise instructions and supplementary knowledge, making it suitable for straightforward applications without requiring additional coding or complex integration.

For this application, the GPT was meticulously configured with instructions to recommend beauty routines using Dorabruschi's product line, tailored to individual user skincare needs. Recognizing the model's sensitivity to prompt structure, extensive trial and error led to the development of six detailed instruction categories: user information collection, routine construction, routine explanation, handling of out-of-scope requests, communication norms, and recommendation limits.

To interact dynamically with the product database, the GPT leverages a code interpreter, which is capable of both writing and executing code based on written instructions. Therefore, a chain-of-thought reasoning approach was developed, breaking the recommendation process into intermediate steps for better accuracy in product extraction. Initially, the GPT gathers essential user data such as skin type, concerns, and target body area. It then uses this information to methodically filter through the product catalog, identifying items matching the user's profile. In this effect, the GPT uses a retrieval-like mechanism, similar to the RAG approach. Here, rigorous prompting was necessary to avoid the frequent compilation errors that arose when it would search for inexistent column values or incorrect column names.

Once suitable products are identified, they are presented to the user in a curated beauty routine, with the GPT explaining the rationale behind each recommendation, including usage instructions and pricing details. Additional safeguards were put in place to prevent the model from generating fictitious product suggestions outside the catalog's scope and conversation starters were designed to steer user interactions (see Figure 3).

The complete prompt configuration, refined through continuous iterations to enhance reliability and reduce errors, can be found in Appendix A. Overall, this approach proved to be a straightforward, no-code solution suitable for non-technical users. However, as will be discussed in subsequent sections, it is plagued by inconsistencies and occasional compilation errors. Another concern is that GPTs are currently not externally deployable and that the user must have ChatGPT Plus to access one.

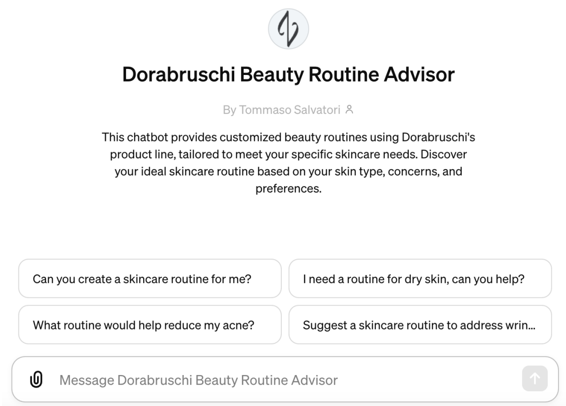

Figure 3. **User interface.** The user interface of the Dorabruschi Beauty Routine Advisor GPT. The user is prompted with conversation starters to reveal their skincare needs and concerns so that the model can produce a customized beauty routine tailored to those requirements.

4.2. RAG-GPT

Retrieval augmented generation (RAG) is a hybrid architectural approach that enhances the capabilities of language models by integrating external knowledge sources into their responses, making them more accurate and contextually relevant. A RAG model consists of two primary components: a retrieval system that identifies relevant data likely to contain the necessary information for answering a query and a generative model that synthesizes these inputs to produce coherent and contextually appropriate responses [12]. In this project, RAG addresses several key challenges:

1. **Context Window Limitation:** It mitigates the limitations posed by the model's context window due to the product data exceeding the token limit of GPT-3.5-Turbo. RAG cleverly incorporates as much relevant product information as possible directly into the prompt, ordered by relevance to the customer's query. This helps prevent the types of compilation errors encountered when using code interpreter with the custom GPT approach, which was due to filtering with incorrect or imprecise keywords.

2. **Hallucinations:** By instructing the model to utilize only the product information presented in the prompt, the risk of generating inaccurate or fictitious content is reduced.

3. **Dynamic Knowledge Integration:** Unlike static approaches like fine-tuning that require retraining to update their knowledge base, RAG dynamically pulls the most relevant data at runtime. This is particularly beneficial for incorporating new products into the Dorabruschi catalog without the need for constant model updates.

To implement RAG, each product's details within the database were concatenated into a comprehensive string labeled 'all product info'. The 'text-embedding-ada-002' model from OpenAI was used to create embedding vectors for each of the 60 product entries. A function was developed to compute an embedding for the customer's query and to retrieve the top-k product embeddings with the highest cosine similarity, as visualized in Figure 4. Cosine similarity, a measure of the angle between two vectors in a multidimensional space, effectively captures the semantic similarity between the query and product descriptions.

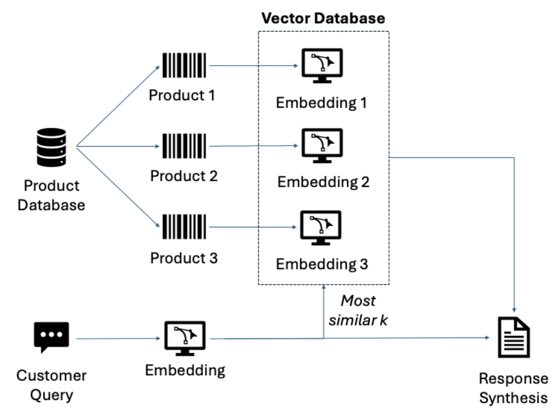

Figure 4. **RAG indexing.** The query is converted into an embedding, which is used to retrieve the top-k most similar products, each with its own embedding in the product database. This is all inserted into the prompt in order to allow the model to produce an accurate and contextually relevant response while adhering to limited context windows. Illustration inspired by [16].

Using this setup, a prompt is constructed by appending the information of the most relevant products into the context window. The model is strictly instructed to base its responses solely on the available product catalog data within the prompt. Through iterative refinement, an optimal system prompt was crafted to guide the model in how to articulate and rationalize its product recommendations to the user (see Figure 5). As will be discussed in subsequent sections, this method produced impressively consistent and relevant results, outperforming the custom GPT approach by a significant margin.

4.3. Fine-Tuned GPT

Fine-tuning is a technique used to further train a pre-trained model – which has vast general knowledge – on a smaller, domain-specific dataset to optimize it for specific

You are tasked with offering customized beauty routine recommendations using only products from Dorabruschi's product line, tailored to the user's specific skincare needs. For each customer query:

- *Recommend products only from the provided Dorabruschi product catalog.*
- *Do not recommend or suggest products outside of this catalog.*
- *For each recommended product, provide a brief explanation of why it has been chosen for the user, detailing its usage and cost.*
- *Limit each routine recommendation to 3-5 products.*
- *If no product in the catalog suits the user's request, clearly state that no suitable product is available. Do not make assumptions about product benefits that are not explicitly supported by the catalog.*
- *In cases of uncertainty, advise the user to consult a skincare specialist or explore other brands for more suitable options.*

Figure 5. **System prompt.** The prompt for the RAG model instructs it to only recommend products from the product catalog, justify each recommendation, and avoid hallucinating in cases of uncertainty.

tasks. In the context of this study, fine-tuning introduces two key benefits: model efficiency and model performance.

4.3.1 Benefits of Fine-Tuning

Firstly, by incorporating product knowledge directly into the model's parameters, fine-tuning reduces the need for extensive prompts with product information. This is a significant advantage over both Custom GPT and RAG, which use a significant portion (all in the case of RAG) of the context window each time to select as many relevant products as possible. Fine-tuning can significantly reduce the number of tokens needed to get the model to perform well on the task, as it requires less external information to generate accurate responses. This means fewer prompt tokens per request, which is both latency and cost-efficient, a significant consideration for a small company like Dorabruschi. Tests from OpenAI have shown that a fine-tuned GPT-3.5-Turbo can produce results of similar or even superior quality to those produced by GPT-4 on certain narrow tasks [2].

Additionally, fine-tuning allows for the incorporation of domain-specific knowledge and vocabulary. By showing many more examples of how to effectively create and recommend a routine from a customer's concern, fine-tuning optimizes the model's ability to understand the nuances, vocabulary, and patterns unique to the cosmetics field [14]. This improves the reliability of the model's outputs, reducing the risk of hallucinations and ensuring more consistent results compared to Custom GPT and RAG models.

4.3.2 Data Preparation

To fine-tune the LLM, a dataset of prompt-completion pairs is required. In this study, the prompt is the question that the customer asks the chatbot, and the completion is the desired routine or product recommendation that the chatbot should generate in response. OpenAI suggests providing 50 to 100 training examples to see clear improvements with GPT-3.5-Turbo [14]. Ideally, these examples would have been crafted by skincare specialists at Dorabruschi, but doing so at scale was impractical due to resource constraints. Instead, they were generated synthetically with human oversight.

ChatGPT was instructed to generate 100 skincare questions that a customer could reasonably ask a store assistant, ensuring diversity in customer preferences, concerns, structure, phrasing, and profiles (see Appendix B). This also ensured that the questions elicited responses covering almost all products in the Dorabruschi database.

The RAG model, which had shown more promising results than Custom GPT, was used to generate the routine for each of the 100 customer questions (see Figure 6 for an example). To ensure all products were included, additional questions were manually added. The Dorabruschi specialist evaluated over 50 responses to ensure quality, noting where more suitable products could have been suggested or where no products should have been recommended. The revised question-completion pairs were converted to JSONL format for fine-tuning. This dataset, totaling 88K tokens, was fine-tuned over three epochs, costing approximately $1.2.

4.3.3 Fine-Tuning Process

The OpenAI API was used with Python to fine-tune the model. Given the limited number of training samples, a batch size of 1 and a learning rate multiplier of 2 were used to prevent overfitting and ensure stable learning. The model was fine-tuned over three epochs to balance training time and performance improvements. Specific details regarding the parameter-efficient fine-tuning technique (PEFT) used are not available in OpenAI documentation, but it likely involves methods such as low-rank adaptation (LoRA) to adapt the pre-trained model to the new task with minimal changes to the model parameters [21].

Preliminary outputs of the fine-tuned model revealed issues with hallucinations, where the model recommended plausible-sounding products that did not exist. To address this, an additional RAG-like semantic search was incorporated. Following Boz et al., 2024 [4], an embedding of the model's recommendation was created, and the top five products with the highest cosine similarity in the Dorabruschi

Prompt: *Can you recommend a product for dull skin that needs brightening?*
Completion: *I recommend the "Revitalizing multivitamin cream" from Dorabruschi for dull skin that needs brightening. This cream is designed to protect the skin from aging and improve hydration, tone, and elasticity. It contains a synergy of beauty vitamins (A, C, E) that accelerate cell turnover, brighten the complexion, and stimulate collagen production. The cream provides immediate softness, radiance, and revitalization to the skin, making it an excellent choice for dull skin in need of brightness and vitality.*

- ***Product:*** *Revitalizing multivitamin cream*
- ***Usage:*** *Apply in the morning and/or evening to perfectly cleansed skin and massage delicately until completely absorbed.*
- ***Price:*** *49.00 euros*
- ***Quantity:*** *50 ml*

This cream will help address the dullness of your skin and give it a more radiant and revitalized appearance.

Figure 6. **Prompt-completion pair.** An example of a short prompt-completion pair from the dataset of 100 used to fine-tune GPT-3.5-Turbo. The prompt was generated using ChatGPT and the response using the RAG-GPT model.

catalog were inserted into the prompt. The model was then instructed to respond using these products, reducing the token count by 80% compared to the RAG model while ensuring all recommended products existed.

4.4. Evaluation Procedure

To assess the performance of the three models (Custom GPT, RAG-GPT, and Fine-Tuned GPT), two product specialists from Dorabruschi were engaged to create 25 evaluation questions based on real-life customer inquiries they had previously received. These questions covered a wide range of scenarios, skin types, and concerns, and varied in complexity. On average, they were considerably more nuanced than those used to fine-tune the final model, given the specificity and detail of each customer inquiry. Three examples can be seen in Appendix C.

Each model was tasked with generating responses to these questions. The specialists then highlighted the correct and incorrect products that were suggested and ranked the responses for each question, providing a qualitative measure of each model's effectiveness in addressing customer needs. Since simple prompt engineering is not sufficient in this context and Dorabruschi has no prior chatbot for comparison, it is difficult to define a baseline for this test. However, the models were evaluated both qualitatively and quantitatively to assess the effectiveness and efficiency of their responses.

The primary metric for comparison was the number of questions for which each model produced the best response. This metric, however, does not consider the degree of incorrectness in the recommendations. For instance, recommending products that have an adverse effect on the customer is significantly worse than suggesting irrelevant ones. Additionally, multiple acceptable products and routines can be recommended for a single customer inquiry, increasing the difficulty in evaluation as different models may provide equally valid yet different recommendations.

To address these complexities, other evaluation metrics were also considered:

1. **Qualitative Evaluation:**
   - 1.1. **Accuracy and Relevance:** Specialists evaluated how accurately and relevantly the models addressed the customer query requirements.
   - 1.2. **User Satisfaction:** Feedback on the clarity and usefulness of the recommendations was gathered to assess general user satisfaction.
2. **Quantitative Evaluation:**
   - 2.1. **Precision:** The proportion of correctly recommended products over all recommended products was measured.
   - 2.2. **Error Analysis:** Common errors or issues in the model responses, such as hallucinations, irrelevant information, or missing critical information, were identified and categorized.

By employing a combination of qualitative and quantitative metrics, this evaluation method provides a comprehensive assessment of the models' performance in real-world scenarios, offering valuable insights into their practical applicability and areas for improvement.

5. Results

The evaluation of the three models—Custom GPT, RAG-GPT, and Fine-Tuned GPT—revealed distinct performance differences in their ability to recommend beauty products based on customer queries. The following sections provide a detailed analysis of their performance, supported by visual representations of key metrics.

5.1. General Performance

Overall, the models demonstrated competence in handling simple, specific queries but struggled significantly with more complex requests requiring comprehensive beauty routines. Out of the 25 questions, 11 were excluded from ranking due to all models providing responses deemed insufficient for customer presentation. The models faced particular difficulty in addressing queries involving multiple skin types or nuanced customer concerns, as shown in Appendix C, where each model recommends a different routine.

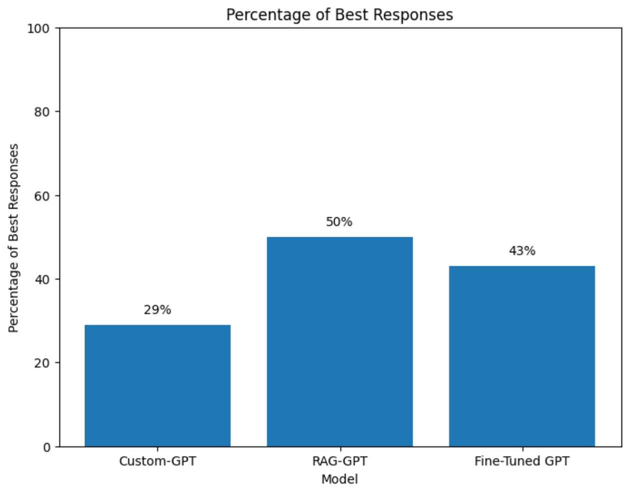

Figure 7. **Percentage of best responses.** Percentage of questions for which each model was voted as providing the best response. Since ties occurred due to indistinguishable quality, percentages do not add up to 100%.

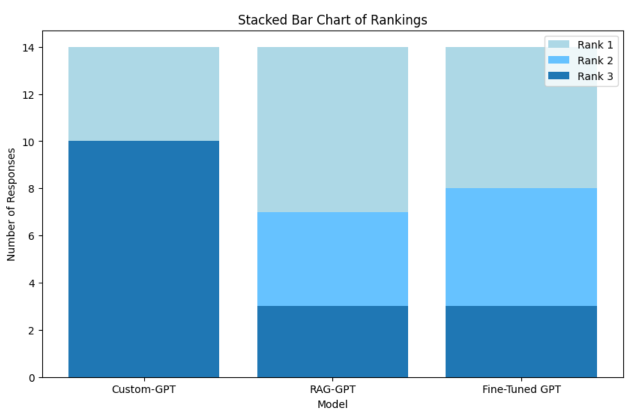

Figure 8. **Rankings.** Stacked bar chart showing the distribution of rankings for each model.

Model	Hallucinated	Adverse
Custom GPT	9	9
RAG-GPT	3	3
Fine-Tuned GPT	0	2

Table 1. **Error analysis.** Number of hallucinated products (non-existent products) and adverse products (inappropriate for the customer concern) recommended by each model.

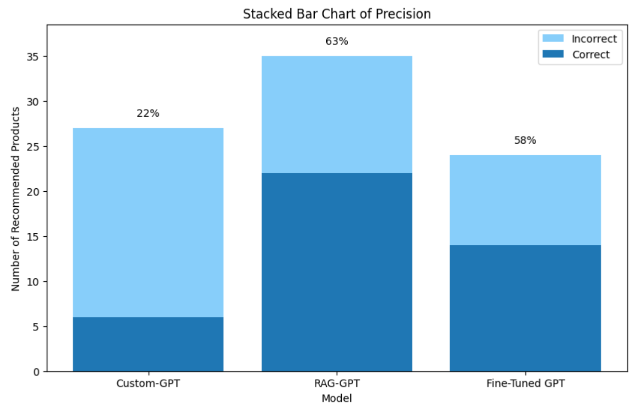

Figure 9. **Precision.** Stacked bar chart showing the precision of each model, defined by the number of correct products recommended over the total products recommended.

5.2. RAG-GPT

The RAG-GPT model was the top performer, providing the best response in 50% of the evaluated questions, as seen in Figure 7. It achieved the highest precision rate of 63% and recommended the most products, with a total of 35 (see 9). This model was chosen as the best due to its ability to produce accurate results and communicate them effectively. Specialists highlighted its clear justifications and tendency to suggest multiple relevant products, often forming adequate mini-routines. However, this approach occasionally led to critical errors, such as including highly incorrect products in its recommendations. This suggests that while the embedding approach of RAG-GPT is effective for clear-cut queries, it struggles with nuanced questions requiring more complex reasoning.

5.3. Fine-Tuned GPT

The Fine-Tuned GPT model provided the best response in 43% of the evaluated questions, with a precision rate of 58%. It recommended a total of 24 products. Specialists noted that its recommendations were often correct but simpler and less detailed compared to RAG-GPT. The Fine-Tuned GPT tended to propose fewer alternatives and sometimes provided single-product responses, which were less eloquent. This model's performance was influenced by its training data, which constrained its output format, affecting its ability to generate comprehensive responses despite its high accuracy rate. Nevertheless, it was the only model that did not hallucinate any products (see Table 1).

5.4. Custom GPT

Custom GPT performed the worst, providing the best response in only 29% of the evaluated questions. It had a precision rate of 22%, recommending a total of 27 products.

This model hallucinated 9 products and recommended another 9 inappropriate products. Specialists found that Custom GPT's responses were often basic and incorrect, showing a tendency to recommend the first few products in its dataset without regard to relevance. This bias resulted in frequent errors and adverse recommendations. The model's poor performance can be attributed to its reliance on detailed input for accurate filtering, which was often lacking, and a tendency to select higher-ranked products in the dataset regardless of their relevance to the customer query.

5.5. Conclusion

In conclusion, the RAG-GPT model demonstrated the best overall performance in recommending beauty products, followed closely by the Fine-Tuned GPT model. Custom GPT lagged significantly behind in accuracy and relevance. Despite these differences, all models showed limitations in handling complex queries requiring comprehensive routines. Future work should focus on improving the models' ability to understand and address multi-faceted customer concerns (example in Figure 10), ensuring recommendations are both accurate and relevant.

Question: *I have combination skin, which tends to be shiny in the T-zone and dry on my cheeks. What beauty routine do you recommend that will be able to give hydration where it is needed and dullness in the troubled spots?*

Figure 10. **Multi-faceted customer concern.** An example of a multi-faceted customer concern in the evaluation dataset, based on real customer interaction. It requires the chatbot to balance providing hydration to dry areas while simultaneously addressing oiliness in the T-zone, demanding a nuanced and tailored recommendation.

6. Discussion

The evaluation of the three models—Custom GPT, RAG-GPT, and Fine-Tuned GPT—revealed distinct performance differences in their ability to recommend beauty products based on customer queries. RAG-GPT emerged as the best performer, demonstrating superior accuracy and relevance in its recommendations. It outperformed both Custom GPT and Fine-Tuned GPT, providing the best responses in 50% of the evaluated questions, achieving the highest precision rate, and recommending the most products.

6.1. Key Findings

The main takeaway of this paper is that simple methods can go a long way. The study highlights the necessity of using retrieval mechanisms for precise product recommendations. Fine-tuning, while useful, was less effective in this context due to issues like hallucinations. The model struggled to acquire sufficient knowledge of the product database with further training on 100 prompt-completion pairs. The fine-tuned model with RAG still underperformed the basic RAG model due to its adherence to an inflexible format, leading to a lower ability to effectively communicate the routine results to the user. The conversational capabilities of pre-trained models were sufficient for presenting results elegantly, suggesting that a simpler approach like RAG may be preferable for this task. Additionally, Custom GPT, which relied on GPT-4 with code interpreter, proved to be an ineffective way to retrieve product information. This also demonstrated that a well-supplemented, smaller model like GPT-3.5-turbo can match and even outperform larger models like GPT-4 on specific tasks.

Compared to previous studies, the findings from this study support the notion that response content is more critical than format in product recommendation tasks. Other studies have also noted the limitations of fine-tuning in handling complex queries without extensive retraining, often supplementing them with additional retrieval mechanisms to avoid hallucination.

6.2. Limitations

Numerous limitations hinder the ability to make a definitive conclusion on the state of LLMs in making effective and comprehensive product recommendations:

1. **Knowledge Base:** The models only had access to product information, which was insufficient to fully understand the complexities of cosmetic routines. Expanding the knowledge base with more dermatological research and cosmetic routine papers is crucial. The product database could also be expanded to better align it with the recommendations, such as including features on what products work well together and how they should be integrated within a routine.

2. **Technical Limitations:** RAG relied on embedding similarity between customer queries and the product database, which could be enhanced by incorporating information retrieval methods like BM25 [17] and HyDE [7] or experimenting with chunking and embedding. The fine-tuned model's performance was constrained by the RAG-generated training dataset, suggesting the need for more high-quality examples written by specialists.

3. **Evaluation Challenges:** 44% of evaluation questions were not rated as all responses were deemed inadequate, significantly limiting the relevance of the results. Future evaluations should include more questions and metrics like the RAGAS score [6] for a more holistic assessment. Additionally, while the models

were evaluated only on single-turn queries, multi-turn queries are essential to mimic real-world interactions and should be included in future tests.

4. **Real-World Applicability:** API calls for closed models like GPT-3.5-turbo are costly, raising concerns about scalability. Exploring cost-effective, scalable open-source alternatives like the new Llama 3 family is essential. Ethical concerns also arise from occasional adverse recommendations. As AirCanada's example showed, even infrequent hallucinations can cause large economic and reputational damage, necessitating the development of ethical frameworks for AI recommendations. In the absence of correctness guarantees for LLM outputs, a 'human in the loop' is essential for checking and fixing the output before the end-user sees the results. Additionally, implementing a helper LLM to verify the output of the product recommendation model can ensure that the products are legitimate and that the recommendation adheres to the standards of a beauty routine.

6.3. Future Directions

The future of generative models is promising. OpenAI's recent release of GPT-4o, which is 2x faster and 50% cheaper with a 128K context window, suggests that retrieval methods may become unnecessary as all product information can fit in the prompt window without excessively large token costs. As models improve in reasoning capabilities, so will their ability to understand product and query dynamics and create comprehensive routines.

In the meantime, future research should focus on i) improving RAG through BM25 and HyDE, experimenting with chunking and embeddings, expanding the knowledge base, and enhancing query handling, ii) improving fine-tuning by expanding the training dataset and finding efficient update methods, iii) extending evaluations to include multi-turn queries, iv) exploring scalable, cost-effective open-source solutions, and v) developing ethical frameworks for AI recommendations. Additionally, extending the RAG approach for multi-turn interactions is not a trivial task since it involves maintaining context across turns by effectively managing dialogue history and leveraging contextual embeddings for more coherent and relevant responses.

Addressing the identified limitations and implementing the suggested future directions will enhance the model's performance and reliability, making it more suitable for real-world applications.

7. Conclusion

The findings of this study underscore the potential of Large Language Models (LLMs) to enhance product recommendation systems, particularly for small companies like Dorabruschi. The evaluation of three distinct LLM-based models—Custom GPT, RAG-GPT, and Fine-Tuned GPT—revealed that RAG-GPT outperformed the other models in terms of accuracy, relevance, and overall user satisfaction. The success of RAG-GPT can be attributed to its effective use of retrieval mechanisms, which allow it to access and integrate relevant product information dynamically, thereby reducing the risk of hallucinations and improving the precision of recommendations.

Key findings from this study highlight several important considerations:

1. **Effectiveness of Retrieval-Augmented Generation (RAG):** RAG-GPT demonstrated superior performance by accurately retrieving and recommending relevant products from Dorabruschi's catalog. This approach effectively addressed the context window limitations of the language models and minimized the generation of irrelevant or hallucinated content.

2. **Limitations of Fine-Tuning:** While fine-tuning provided some improvements in recommendation quality, it struggled with complex queries and was less flexible in generating comprehensive responses compared to the RAG approach. The fine-tuned model's performance was constrained by the limited number of high-quality training examples and an inflexible output format.

3. **Challenges with Custom GPT:** The Custom GPT model, despite leveraging GPT-4 and a code interpreter, performed the worst due to frequent hallucinations and inaccuracies in product recommendations. This underscores the need for robust retrieval mechanisms to supplement language models, especially when dealing with extensive and detailed product catalogs.

The implications of these findings extend beyond the domain of cosmetics. For small companies facing cold-start problems and limited historical data, integrating LLMs with retrieval mechanisms like RAG can significantly enhance the effectiveness of recommendation systems. This approach ensures that recommendations are based on accurate and up-to-date information, providing a better user experience and increasing customer satisfaction.

In conclusion, this study highlights the importance of combining LLMs with robust retrieval mechanisms to enhance the performance of recommendation systems. Future research should focus on expanding the knowledge base of these models, exploring advanced retrieval techniques, and developing scalable, cost-effective solutions to further improve their applicability and reliability in real-world scenarios.

Acknowledgements

The author would like to express sincere gratitude to Prof. Rama Ramakrishnan for his infinite wisdom and passion. His support and guidance were invaluable throughout this research. A special thanks goes to the team at Dorabruschi - Elena Verrazzani, Claudia Guandalini, and Mario Salvatori - for their continuous technical support, evaluation, and enthusiasm for the project.

References

[1] Introducing gpts. https://openai.com/index/introducing-gpts/, November 2023. Accessed: 2024-05-09. 3

[2] Gpt-3.5 turbo fine-tuning and api updates. https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/, 2024. Accessed: 2024-05-15. 5

[3] Pricing. https://openai.com/api/pricing/, May 2024. Accessed: 2024-05-09. 3

[4] Artun Boz, Wouter Zorgdrager, Zoe Kotti, Jesse Harte, Panos Louridas, Dietmar Jannach, and Marios Fragkoulis. Improving sequential recommendations with llms, 2024. 5

[5] Aminu Dau and Naomie Salim. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53, 04 2020. 1

[6] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation, 2023. 8

[7] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022. 8

[8] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommender system, 2023. 1, 2

[9] Liangliang Hou, Di Jin, Xiang Wang, and Defu Li. Prompting large language models for recommender systems: A comprehensive framework and empirical analysis. *arXiv preprint arXiv:2301.04997*, 2023. 2

[10] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008. 2

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. 2

[12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. 2, 4

[13] Qiao Li, Alexandros Karatzoglou, and Xavier Amatriain. A transformer-based sequential recommendation model for news recommendation. *arXiv preprint arXiv:2005.00520*, 2020. 2

[14] OpenAI. Fine-tuning integrations. https://platform.openai.com/docs/guides/fine-tuning/fine-tuning-integrations, 2024. Accessed: 2024-05-15. 5

[15] Rama Ramakrishnan. How to use llms as "helpers" to build and customize other llms. https://www.dropbox.com/scl/fi/ouitsdx6y8kxxi6a5kxya/Helper-LLMs.pdf?rlkey=3xpeew5m2dv5vtyr9vg5pwohv&dl=0, January 2024. Accessed: 2024-05-19. 3

[16] Sebastian Raschka. Finetuning large language models. May 2024. Accessed: 2024-05-17. 4

[17] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. Yes, bm25 is a strong baseline for legal case retrieval, 2021. 8

[18] Tommaso Salvatori. Llm-cosmetics-recommendation. https://github.com/tommysalva/LLM-Cosmetics-Recommendation/tree/main, 2024. Accessed: 2024-05-17. 1

[19] Damien Sileo, Wout Vossen, and Robbe Raymaekers. Zero-shot recommendation as language modeling, 2021. 1

[20] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Sun, and Houqiang Wang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformers. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019. 2

[21] Towards Data Science. Parameter-efficient fine-tuning (peft) for llms: A comprehensive introduction. *Towards Data Science*, 2024. Accessed: 2024-05-15. 5

[22] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation, 2023. 1

[23] Maria Yagoda. Airline held liable for its chatbot giving passenger bad advice - what this means for travellers, 2024. Accessed: 2024-05-10. 1

[24] Li X. Gao M. Wu Z. Zhang, J. and H. Hu. Exploring the impact of large language models on recommender systems: An extensive review. *arXiv preprint arXiv:2302.18590*, 2023. 2

A. Appendix: Custom GPT Prompt

"Utilize the attached Dorabruschi product database to offer customized beauty routine recommendations using Dorabruschi's product line, tailored to the user's skincare needs.

1. **Gathering User Information:** Ask users to provide their skin type (options: normal, combination, oily, sensitive, or dry), primary skin concerns (options: aging, wrinkles, acne, dryness, sagging, blemishes, sensitivity, redness, or dullness), and the specific body area they want to focus on (e.g., face, hands, legs).

2. **Routine Construction:** Use the 'title' column in the spreadsheet for product names. It is vital that you do not deviate from these names or suggest any product that is not from the product database list. For products for feet and hands, the product title will include 'foot' or 'hand' in the name. All other products are for the face. For skin type compatibility, refer to the 'skin_type' column. Products labeled as 'All Skin Types' are universally suitable. Address user concerns by aligning them with the 'intended_concerns' column in the dataset. The 'benefits' and 'properties' columns also give an indication of what types of concerns the product is suitable for. The úsage_instructionsǵive information on how to apply the product, including whether to use it in the morning, evening, or both. Based on the user's responses and these instructions, analyze the Dorabruschi product database and construct a personalized skincare routine, considering the compatibility of each product with the user's skin type, concerns, and body area.

3. **Routine Explanation:** Explain why each product in the routine was chosen, detailing how it addresses the user's concerns and how it fits into their daily skincare regimen, as well as how to apply it. Use comprehensive information from all relevant columns to support this. The products should work well together and it should be clear why, following the conventions of a good beauty routine. Also provide the price of the routine, which is in Euros and available in the 'price' column.

4. **Handling Out-of-Scope Requests:** If a user request does not directly match the dataset's columns, recommend based on available information without causing errors. Always prioritize user-specified needs within the dataset's scope.

5. **Communication Guidelines:** If uncertain about a query, express this honestly rather than making assumptions. Request additional information from the user if their initial inputs are insufficient for a personalized recommendation. Avoid offering medical advice or making claims about product effectiveness not supported by the data.

6. **Limitations and Recommendations:** Limit to a maximum of 3-4 products per routine to maintain clarity and effectiveness. Encourage users with serious skin issues to consult a dermatologist.

"

B. Appendix: ChatGPT Question-Generation Prompt

I need you to generate synthetic question-answer (Q-A) pairs for fine-tuning an LLM model. My goal is to recommend a beauty routine based on customer concerns and demographics. Devise 100 questions that capture various customer scenarios, skin concerns, preferences, and other relevant factors, simulating questions that a customer could reasonably ask a store assistant. The corresponding answers will detail a recommended routine, explaining why each product is suitable. The below guidelines should help you understand how to formulate good questions. Take your time if needed. Store these 100 questions in a CSV file.

Guidelines:

- Ensure Diversity: Diversity in the questions is key to training a robust model. Diverse questions help the model learn to handle a wide range of customer inquiries, which is critical for a retail setting where customer preferences and concerns can vary greatly.
- Vary the Structure and Phrasing: Use different question structures and phrasing to avoid biasing the model towards a particular pattern. This includes mixing simple, compound, and complex sentences.
- Incorporate Different Customer Profiles: Include questions that represent various customer demographics, skin types, concerns, and preferences. This approach ensures that the model can cater to a broad audience.
- Use Contextual and Content Diversity: Implement algorithms that promote diversity by modeling different question types and relating contextual focuses with content selectors

C. Appendix: Example Question and Model Answers

- **Question**: "I have combination skin, which tends to be shiny in the T-zone and dry on my cheeks. What beauty routine do you recommend that will be able to give hydration where it is needed and dullness in the trouble spots?"

- **Custom GPT's Answer**: "Based on your combination skin type and concerns with dryness and dullness, I've curated a personalized beauty routine for you using Dorabruschi products. These selections aim to provide hydration to dry areas and address dullness in your T-zone and cheeks.

  1. **Revitalizing Multivitamin Cream**
     Description: This cream has a velvety and light texture, designed to provide hydration and revitalize the skin. It's perfect for areas of your skin that need a hydration boost. Usage Instructions: Apply in the morning and/or in the evening to a cleansed face and neck. Benefits: The cream is formulated with anti-wrinkle and antioxidant properties, which can help illuminate and revitalize the skin, addressing dullness. Price: €49.00
  2. **ACE 10% Multivitamin Concentrate** ... [TRUNCATED]

  These products have been chosen to work synergistically to provide moisture where needed and enhance the overall brightness of your skin. The routine starts with a serum to prep and protect your skin, followed by a hydrating cream that can be used twice daily. At night, the renewing cream will help in smoothing and improving the skin texture."

- **RAG-GPT's Answer**: "I recommend the following beauty routine using products from Dorabruschi's product line for your combination skin:

  1. **Rebalancing Face Cream:** This light emulsion is formulated with a sebum-regulating action, making it ideal for your T-zone that tends to be shiny. The thiolysin complex in this cream helps to regulate sebum production while providing hydration. Apply this morning and evening to perfectly cleansed skin to help balance your skin's moisture levels. Price: 39.00 euros for 50 ml.
  2. **Cream K Anti-Wrinkle Cream:** ... [TRUNCATED]

  Unfortunately, there is no specific product in the catalog that targets dullness. For more targeted solutions for dullness, you may want to consult a skincare specialist or explore other brands."

- **Fine-Tuned GPT's Answer**: "For your combination skin that is shiny in the T-zone and dry on the cheeks, I recommend the following Dorabruschi beauty routine to provide hydration where needed and address dullness in the trouble spots:

  1. **Delicate sebum-balancing cleansing base**
     - **Description:** Extremely delicate washing base that combines a rebalancing action with a moisturizing and soothing action, ideal for impure and reddened skin. - **Usage:** Apply a small amount to a damp face in the morning and evening, massage gently, and rinse thoroughly. - **Properties:** Cleanses, Rebalances - **Skin Type:** Oily - **Texture:** Foaming gel - **Price:** 22.00 euros - **Quantity:** 165 ml
  2. **Gentle cleansing milk** ... [TRUNCATED]

  This routine will help cleanse your skin effectively, balance sebum production, provide hydration, and address skin imperfections, giving you a more uniform and smoother complexion."