ENGIE4800 - Data Science Capstone

Final Report: Spring 2025

Caterina Almazan[1], ChengHsin Chang[2], Claire Yi-Chen Chen[3], Yu-Heng Chi[4] and Param Sejpal[5],
Columbia Engineering, NY

cga2133@columbia.edu[1], cc5211@columbia.edu[2], yc4562@columbia.edu[3], yc4548@columbia.edu[4], pns2129@columbia.edu[5]

Company: L'Oréal USA

Title: Multi-Modal Search using RAG

Project Mentor: Professor Sining Chen

Industry Mentors: Rémi Ferreira

Table of contents

1 Introduction

As artificial intelligence continues to transform e-commerce, L'Oréal aims to improve product discoverability and user experience through intelligent, context-aware search systems. Traditional keyword-based search engines often struggle with capturing nuanced product preferences, such as specific styles, colors, or design intents—making it difficult for users to find exactly what they're looking for.

To address this, our project introduced a **multi-modal shopping assistant** powered by **Retrieval-Augmented Generation (RAG)** with a **Large Language Model (LLM)**, integrating both **textual and visual product data**. This system enables users to search using natural language queries, like *"small brown wallet under $20 with good reviews"*, and receive curated, relevant product recommendations.

Our pipeline began with **keyword extraction** from user queries using LLMs to separate unstructured product descriptions from structured constraints (e.g., price, rating). These queries were embedded into a vector and compared against a pre-embedded product database built from text and image embeddings. We evaluated multiple embedding strategies—including **cross-modal embeddings (CLIP, BLIP, Sentence Transformer)** and **image-to-text (BLIP)** approaches—and stored the resulting vectors in a **FAISS index** for efficient similarity search.

To generate personalized responses, we used the **OpenAI API** to rank, filter, and describe the most relevant products, incorporating structured filters such as price and ratings. The final results were delivered through an interactive **Streamlined interface**, providing an intuitive and intelligent product discovery experience for L'Oréal customers.

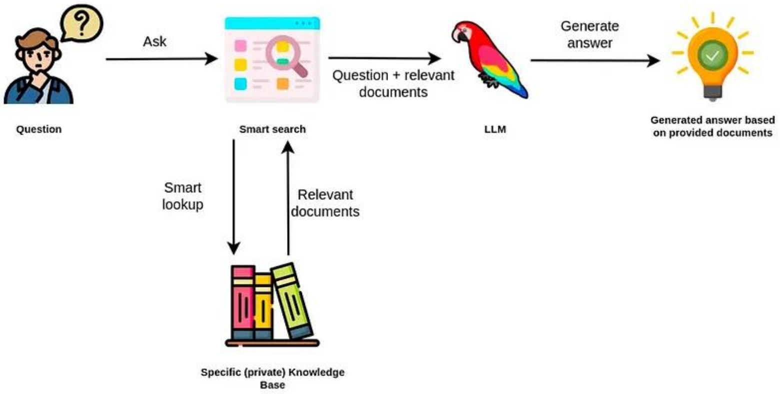

Ask
Question
Smart search
Question + relevant documents
LLM
Generate answer
Generated answer based on provided documents
Smart lookup
Relevant documents
Specific (private) Knowledge Base

Figure 1: Multimodal RAG pipeline

2 Literature Review

Previous data science capstone projects under L'Oréal's AI initiatives provide critical insights into retrieval-based search methods. These studies compared Vector RAG and Graph RAG in AI-driven product recommendation systems, influencing our approach to multi-modal search.

The Fall 2024 project [1], conducted by Aneri Bijal Modi, Ishita Pundir, Shumail Sajjad, Yash Anish Dange, and Zainab Shakruwala, evaluated Vector RAG vs. Graph RAG for L'Oréal's chatbot and product recommendation system. Their results showed Vector RAG is faster and better suited for handling broad, unstructured queries, whereas Graph RAG was more effective in understanding hierarchical relationships but was computationally expensive. Given our goal of integrating image-based retrieval, we build on their findings by optimizing Vector RAG's efficiency in a multi-modal search setting.

Another Fall 2024 project [2], led by Akarsh Rastogi, Leah Uzzan, Nafisa Ali, Rajat Gupta, Tushar Badhwar, and Tushar Prasad, focused on minimizing hallucinations in AI-generated product recommendations by comparing Vanilla RAG and GraphRAG. Their results confirmed GraphRAG enhances precision but requires high processing power, while Vector RAG provides better recall and adaptability, making it preferable for large-scale retrieval. Since our project expands retrieval to both text and image data, we follow their recommendation of prioritizing vector-based retrieval while integrating CLIP embeddings for multi-modal search.

By leveraging these findings, our project extends previous research by integrating multi-modal search with both text and image embeddings, which was not explored in past studies. Unlike prior work that primarily focused on text-based retrieval, our approach incorporates CLIP embeddings for image-driven search, enabling users to find products more intuitively. Additionally, where earlier projects aimed to reduce hallucinations and improve retrieval efficiency, we go further by integrating OpenAI API-driven response generation to enhance contextual understanding and personalization in recommendations. This distinction allows us to refine retrieval mechanisms while ensuring a seamless, dynamic, and intelligent product discovery experience for L'Oréal customers.

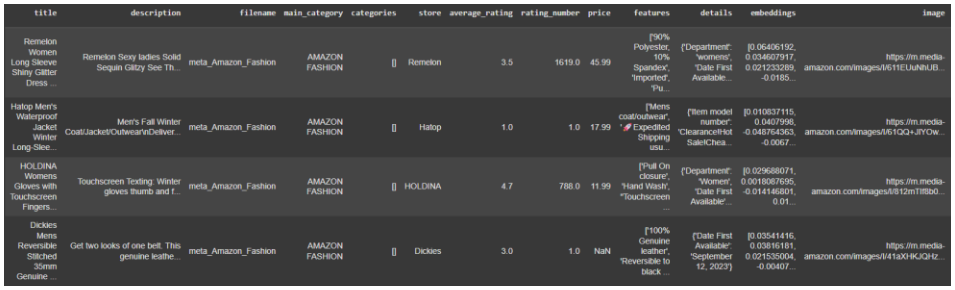
3 Problem Statement & Final Project Goal

L'Oréal's portfolio consists of over 36 global brands, creating a vast array of assorted products that all cater to different customer needs and preferences. Thus, it is easy to understand how overwhelming the shopping experience can be as a beauty consumer. With so many options, how can one possibly narrow-down the options in searching for the best product for *you*? Additionally, when customers are looking for product attributes that are visually apparent - such as the color of a product - but not explicitly stated in text descriptions, traditional search engines, which mostly rely solely on textual data, fall short in capturing such visual details. Therefore, multi-modal search - a search model based on both textual and visual information - becomes a key focus in our project.

Our final goal was to create an intelligent shopping assistant powered by a functional multi-modal search model that leverages the embedding applications of a vector-based RAG on a diverse product dataset containing both image and text data. This shopping assistant could take potential L'Oréal customer queries regarding products as input and outputs certain products that best meet the customer's search needs and/or query constraints. For example, if a potential customer input a search query such as "Find a shampoo in a purple bottle that is under 12 oz and less than $20," the goal for our model was to output the relevant products that fit these descriptions as closely as possible.

4 Dataset

Our project utilized a dataset from Hugging Face called "Amazon Product" [3], which contains approximately 117k products, each with associated data such as title, description, features, category, average ratings, price, and product image url. To enhance model building efficiency, we extracted a representative subset. Specifically, we sampled 10% from each product category, resulting in a final dataset of 11.7k products for our project.



title	description	filename	main_category	categories	store	average_rating	rating_number	price	features	details	embeddings	image
Remelon Women Long Sleeve Shiny Glitter Dress ...	Remelon Sexy ladies Solid Sequin Glitzy See Th...	meta_Amazon_Fashion	AMAZON FASHION	[]	Remelon	3.5	1619.0	45.99	['90% Polyester, 10% Spandex', 'Imported', 'Pu...	{'Department': 'womens', 'Date First Available...	[0.06406192, 0.034607917, 0.021233289, -0.0185...	https://m.media-amazon.com/images/I/611EUuNhUB...
Hatop Men's Waterproof Jacket Winter Long-Slee...	Men's Fall Winter Coat/Jacket/Outwear\nDeliver...	meta_Amazon_Fashion	AMAZON FASHION	[]	Hatop	1.0	1.0	17.99	['Mens coat/outwear', '🚀 Expedited Shipping usu...	{'Item model number': 'Clearance!Hot Sale!Chea...	[0.010837115, 0.0407998, -0.048764363, -0.0067...	https://m.media-amazon.com/images/I/61QQ+JlYOw...
HOLDINA Womens Gloves with Touchscreen Fingers...	Touchscreen Texting: Winter gloves thumb and f...	meta_Amazon_Fashion	AMAZON FASHION	[]	HOLDINA	4.7	788.0	11.99	['Pull On closure', 'Hand Wash', "Touchscreen ...	{'Department': 'Women', 'Date First Available'...	[0.029688071, 0.0018087695, -0.014146801, 0.01...	https://m.media-amazon.com/images/I/812mTlf8b0...
Dickies Mens Reversible Stitched 35mm Genuine ...	Get two looks of one belt. This genuine leathe...	meta_Amazon_Fashion	AMAZON FASHION	[]	Dickies	3.0	1.0	NaN	['100% Genuine leather', 'Reversible to black ...	{'Date First Available': 'September 12, 2023'}	[0.03541416, 0.03816181, 0.021535004, -0.00407...	https://m.media-amazon.com/images/I/41aXHKJQHz...

Figure 2: Dataset Head

4.1 EDA

To understand the product composition of the representative data subset, we created the word cloud image using product titles with the removal of common stopwords and irrelevant key words. Using the WordCloud library, we generated a visualization with a "coolwarm" colormap to highlight frequently occurring words. The resulting image provides insights into dominant themes in our dataset, with larger words representing more frequently mentioned product attributes and categories.

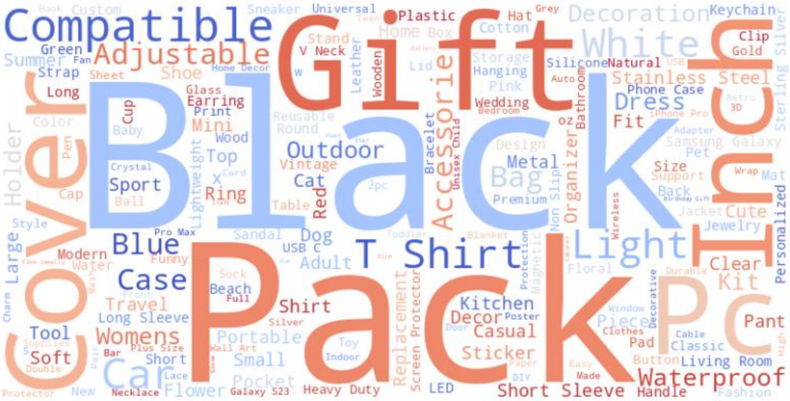


Figure 3: Data Word Cloud

5 Methodology and Structure

Our project implemented a multi-modal product search assistant using a Retrieval-Augmented Generation (RAG) framework designed to support L'Oréal's extensive and diverse product catalog. The system accepts natural language queries and returns personalized product recommendations by leveraging both textual and visual features of the product data. The methodology is composed of five key components: (1) Query understanding and keyword extraction, (2) Query embedding, (3) Vector-based product retrieval, (4) Response generation, and (5) Result visualization through a user interface. Each stage is modular and contributes to building an end-to-end, intelligent, and scalable AI-powered search tool.

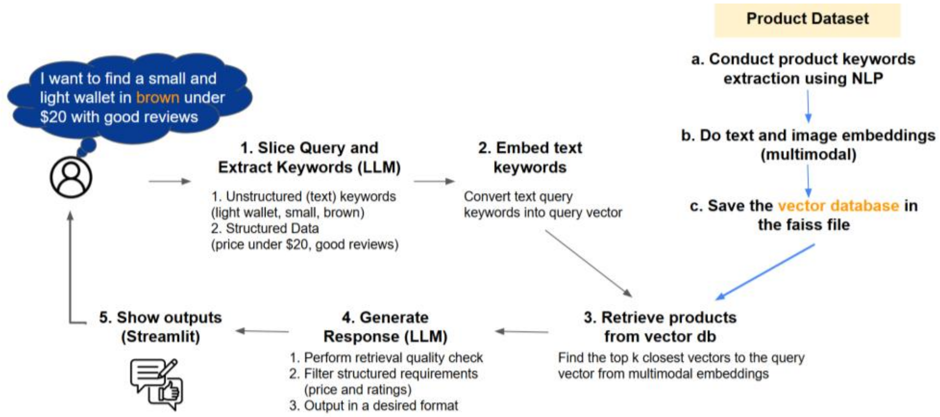

Product Dataset
a. Conduct product keywords extraction using NLP
b. Do text and image embeddings (multimodal)
c. Save the vector database in the faiss file
I want to find a small and light wallet in brown under $20 with good reviews
1. Slice Query and Extract Keywords (LLM)
1. Unstructured (text) keywords (light wallet, small, brown)
2. Structured Data (price under $20, good reviews)
2. Embed text keywords
Convert text query keywords into query vector
3. Retrieve products from vector db
Find the top k closest vectors to the query vector from multimodal embeddings
4. Generate Response (LLM)
1. Perform retrieval quality check
2. Filter structured requirements (price and ratings)
3. Output in a desired format
5. Show outputs (Streamlit)

Figure 5: Project Workflow

5.1 Slice Query and Extract Keywords (LLM)

In the initial stage, the system processes the user's input query using an LLM to extract relevant information for both retrieval and filtering. Natural language queries are often unstructured and may include a mix of descriptive terms and explicit constraints. For instance, the query *"Find me a brown, lightweight wallet under $20 with good reviews"* contains:

- Descriptive (unstructured) terms: "brown", "lightweight wallet"
- Constraints (structured): "under $20", "good reviews"

To address this, we designed a custom **slice_query()** function powered by a large language model (OpenAI's API). This function used a tailored prompt to segment the user query into two parts:

- **Unstructured keywords (output_2)**: Product name and related descriptive terms
- **Structured filters (output_1**): Constraints based on price, rating, etc.

The model was instructed to preserve semantic details while converting them into minimal and searchable tokens. These outputs were then used for the next stages: unstructured keywords guide retrieval, and structured constraints were applied during filtering and ranking.

5.2 Embed Text Keywords

The second stage involved embedding the unstructured keyword component **(output_2**) into a vector space using a **Sentence Transformer** model (e.g., **all-MiniLM-L6-v2**). This model was chosen for its strength in capturing sentence-level semantic similarity while maintaining computational efficiency.

Commented [1]: Shall we write Sentence Transformer only? That's the final model but we actually used the same text embedding method as the one to embed the product set

The embedding process mapped the textual input into a fixed-size high-dimensional vector, enabling similarity-based search across the product vector database. This query vector was passed to the vector similarity engine for product retrieval.

By using Sentence Transformers over traditional TF-IDF or BERT token embeddings, the system can match semantically similar queries even if there is no exact textual overlap—for example, mapping "lightweight" with "compact" or "portable".

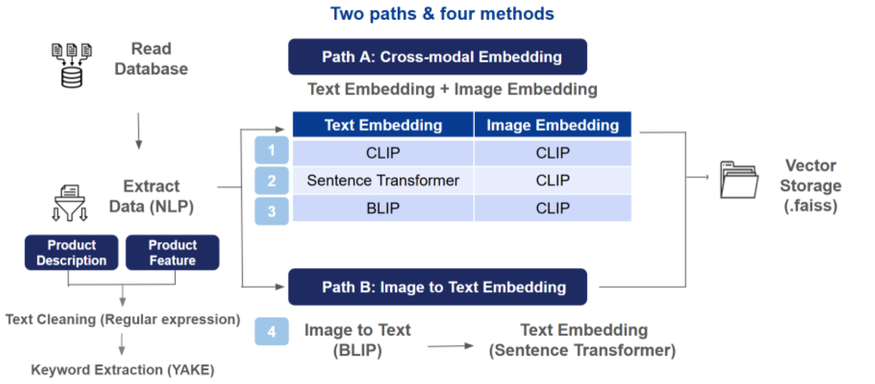
5.3 Product Database Embedding and Retrieval (FAISS)

To enable efficient product retrieval, we built a product vector database by embedding both textual and visual information from our curated dataset. Each product entry contains:

- Textual features: Product title and features
- Image features: Product image url

During EDA, we observed that some textual features were overly lengthy or contained irrelevant information. To improve the quality of the embedded vectors, we applied Natural Language Processing (NLP) techniques to extract product keywords from product title and features to optimize embedded output by improving the input data quality. Specifically, we performed text cleaning using regular expressions and employed the Yet Another Keyword Extraction (YAKE) model to extract keywords from the product description and features.

Note that structured data such as price and customer ratings were excluded from this stage and instead handled during the constraint filtering process in Section 5.4 Response Generation.


Two paths & four methods
Read Database
Extract Data (NLP)
Product Description
Product Feature
Text Cleaning (Regular expression)
Keyword Extraction (YAKE)
Path A: Cross-modal Embedding
Text Embedding + Image Embedding
Path B: Image to Text Embedding
4
Image to Text (BLIP)
Text Embedding (Sentence Transformer)
Vector Storage (.faiss)

	Text Embedding	Image Embedding
1	CLIP	CLIP
2	Sentence Transformer	CLIP
3	BLIP	CLIP

Figure 6: Product database embedding pipeline

The product data was processed through two parallel embedding paths:

(a) Cross-Modal Embedding Path

Each product was embedded along two modalities:

- **Text Embedding**: Using Sentence Transformers, CLIP, or BLIP (for comparative evaluation)
- **Image Embedding**: Using CLIP to convert product images into visual vectors aligned with the textual space

We experimented with the following three combinations:

1. Sentence Transformer (text) + CLIP (image)
2. CLIP (text) + CLIP (image)
3. BLIP (text) + CLIP (image)

The resulting product vectors were stored in a FAISS index. At inference time, a user query was similarly embedded and compared to all product vectors using cosine similarity to retrieve the most relevant items.

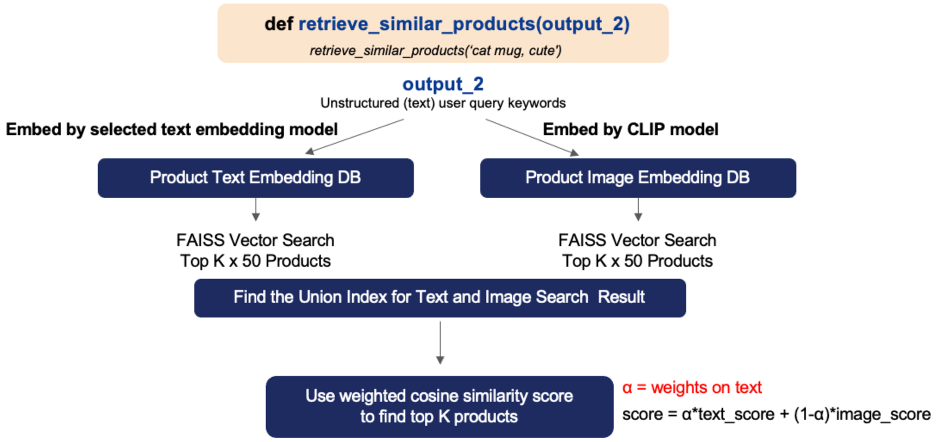

def retrieve_similar_products(output_2)
retrieve_similar_products('cat mug, cute')
output_2
Unstructured (text) user query keywords
Embed by selected text embedding model
Embed by CLIP model
Product Text Embedding DB
Product Image Embedding DB
FAISS Vector Search
Top K x 50 Products
FAISS Vector Search
Top K x 50 Products
Find the Union Index for Text and Image Search Result
Use weighted cosine similarity score to find top K products
α = weights on text
score = α*text_score + (1-α)*image_score

Figure 7: Cross-modal Embedding Retrieval

In the case of cross-modal embeddings, where both text and image embeddings were used, we applied a weighted similarity formula:

$$\text{score} = \alpha \times \text{text\_similarity} + (1 - \alpha) \times \text{image\_similarity}$$

Here, **α** is a tunable parameter (default value: 0.6) that controls the relative weight of text vs. image similarity, allowing the system to adjust emphasis based on the selected embedding model, output quality assessment, specific use cases, or domain requirements.

Commented [2]: Not sure where the 0.6 comes from - I remember that's not the final alpha

For each query, we retrieved the top-*k* products (typically $k = 5$) with the highest similarity scores and forwarded them to the next stage for filtering and final response generation.

(b) Image-to-Text Embedding Path

Here, product images were first converted into captions using BLIP (Bootstrapped Language-Image Pretraining). These generated captions were then embedded using Sentence Transformers, effectively treating the image as text.

All embedded vectors were indexed using Facebook AI Similarity Search (FAISS) to enable fast nearest-neighbor retrieval. During the search process, we computed the cosine similarity between the user's query vector and each stored product vector.

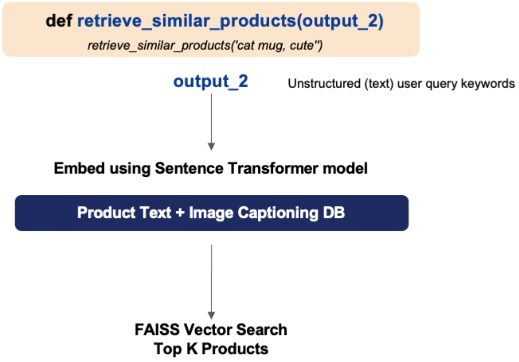

def retrieve_similar_products(output_2)
retrieve_similar_products('cat mug, cute")
output_2
Unstructured (text) user query keywords
Embed using Sentence Transformer model
Product Text + Image Captioning DB
FAISS Vector Search
Top K Products

Figure 8: Image to Text Embedding Retrieval

5.4 Response Generation (LLM-Prompted)

The retrieved product candidates were sent to the OpenAI API to produce a final, formatted product recommendation. A carefully engineered prompt ensures that the LLM performs the following task as an AI Shopping agent:

1. **Relevance Filtering**: Verify whether the retrieved items truly match the user's semantic intent.
2. **Constraint Filtering**: Apply structured conditions such as price ceilings or minimum ratings.
3. **Response Formatting**: Return a readable, structured output for each selected product.

With all information, including the user’s query and candidate products from retrieval, the agent recommends 1-3 products from the retrieved products that best match the user's preferences. Prioritizing products that match the user’s specific product description (e.g., design, style, material, target audience) in query_text first. It considers price and rating only after verifying the product is relevant.

The final response included the following attributes for each recommended item:

- Product name
- Short description
- Price
- Rating
- Key pros/cons (if derivable from metadata)
- Image URL (for display in UI)
- Product URL (if available)

This step allows for **natural language personalization**, going beyond raw similarity to create a context-aware conversational experience.

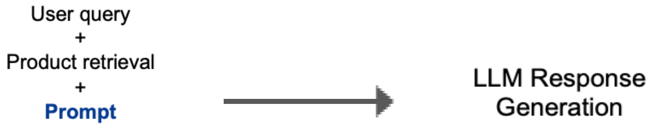


Figure 9: Response Generation Workflow

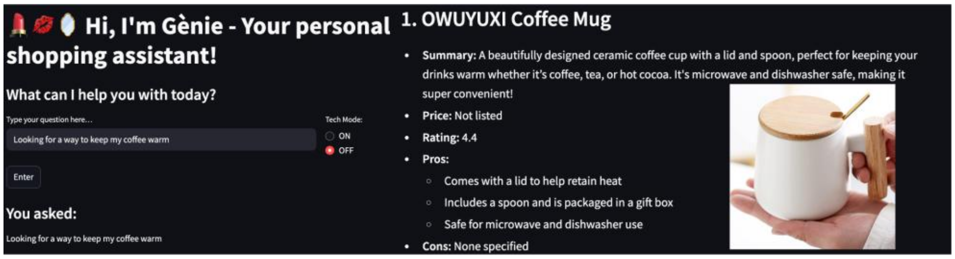
5.5 Output Visualization via Streamlit

The end-user interface was developed using Streamlit, enabling quick deployment of a functional web-based prototype. Named ***Gènie***, the application features:

- A text input field for user queries
- Dynamically rendered product cards with real-time response generation
- An intuitive layout optimized for e-commerce use cases

Each product card includes a product image, name, description, price, rating, and optionally a link to the product page. The UI supports rapid prototyping, user testing, and real-time demonstrations of the underlying AI system.

This interface bridges the technical backend with user-centric design, making the model tangible and interpretable for non-technical stakeholders, customers, and potential integrators within L'Oréal’s digital ecosystem.



💄💋🪞 Hi, I'm Gènie - Your personal shopping assistant!

What can I help you with today?

Type your question here...

Looking for a way to keep my coffee warm

Tech Mode:
○ ON
● OFF

You asked:

Looking for a way to keep my coffee warm

1. OWUYUXI Coffee Mug

- **Summary:** A beautifully designed ceramic coffee cup with a lid and spoon, perfect for keeping your drinks warm whether it’s coffee, tea, or hot cocoa. It's microwave and dishwasher safe, making it super convenient!
- **Price:** Not listed
- **Rating:** 4.4
- **Pros:**
  - Comes with a lid to help retain heat
  - Includes a spoon and is packaged in a gift box
  - Safe for microwave and dishwasher use
- **Cons:** None specified

Figure 10: User Interface - Gènie

6. Evaluation

To objectively compare the performance of different embedding strategies within our multi-modal search pipeline, we designed and implemented an **LLM-based evaluation framework**. This framework simulates human judgment by using a large language model to evaluate and rank search results produced by various embedding models.

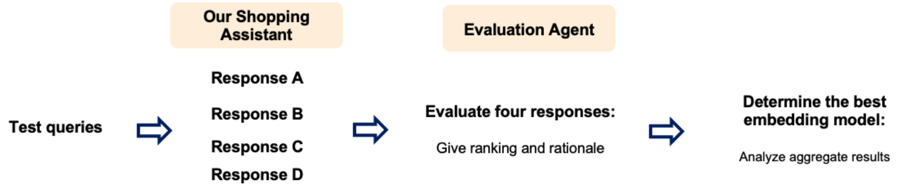


Figure 11: Evaluation framework

6.1 Evaluation Methodology

We curated a set of 30 realistic user queries, covering a range of product complexities—from simple descriptors like "silver necklace" to compound constraints like "purple shampoo under $20 in a travel-size bottle." For each query, our shopping assistant generated four responses using four different embedding configurations:

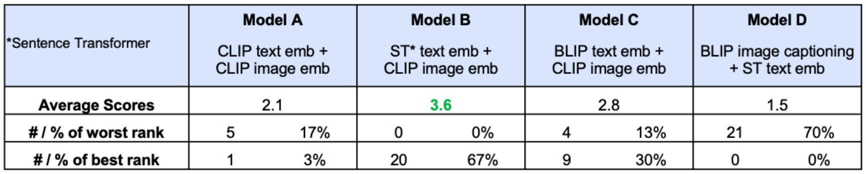
- **Model A**: CLIP for both text and image embeddings
- **Model B**: Sentence Transformer for text + CLIP for image embeddings
- **Model C**: BLIP for text + CLIP for image embeddings
- **Model D**: BLIP-generated captions embedded using Sentence Transformer

Each query produced four responses (A–D), which were then evaluated by a custom LLM evaluation agent. This agent was instructed to:

- Review the four responses
- Assign a ranking from 1 (worst) to 4 (best) for each
- Provide rationale for its scoring, focusing on relevance, completeness, and clarity

6.2 Evaluation Results

The aggregated results from the 30 test queries revealed that **Model B**—which uses **Sentence Transformer for text** and **CLIP for image embeddings**—consistently outperformed all other configurations.



*Sentence Transformer	**Model A** CLIP text emb + CLIP image emb		**Model B** ST* text emb + CLIP image emb		**Model C** BLIP text emb + CLIP image emb		**Model D** BLIP image captioning + ST text emb	
Average Scores	2.1		**3.6**		2.8		1.5	
# / % of worst rank	5	17%	0	0%	4	13%	21	70%
# / % of best rank	1	3%	20	67%	9	30%	0	0%

Figure 12: Evaluation Results

Model B achieved the **highest average ranking (3.6)** and was never ranked the worst across any test case. It also received the top ranking in **67%** of all queries. Conversely, Model D—our image-to-text pipeline—performed the poorest, being ranked lowest in 70% of cases.

7. Conclusion

This project demonstrated the successful development of an AI-powered multi-modal shopping assistant that leveraged both text and image data to enhance product search and recommendation for L'Oréal's diverse catalog. By implementing a RAG framework enriched with cross-modal embeddings, LLM-based filtering, and a user-facing interface, we bridged the gap between complex user intent and accurate product discovery.

Through rigorous testing across multiple embedding strategies, we identified the most effective model—**Sentence Transformer (text) + CLIP (image)**—which was adopted for final deployment due to its superior accuracy and user relevance.

Our evaluation framework further supports the scalability and robustness of the solution by enabling automated assessment of output quality, eliminating the need for manual reviews while maintaining human-level judgment. The final Streamlit user interface design, Gènie, makes the model accessible, interpretable, and ready for real-world application.

In conclusion, this work provides a state-of-the-art, multi-modal, and end-to-end AI solution that redefines product search for L'Oréal, offering enhanced personalization, intelligent filtering, and improved customer experience at scale.

8 Ethical Considerations

As we develop AI systems that influence consumer decision-making, it is imperative to address the ethical dimensions of our multi-modal product search assistant. Our system aims to improve user experience, but also carries responsibilities related to privacy, transparency, and user trust.

8.1 Data Privacy and Consent

Although we use publicly available datasets (e.g., Amazon Product 2023), the potential for future deployment on real L'Oréal customer data warrants strong safeguards. Product reviews and images, even when anonymized, may contain sensitive or personal information. As such, any extension of this system to incorporate proprietary or user-generated content must comply with data protection regulations within each state / country. This includes clear data provenance, informed consent, and mechanisms for data removal or correction.

8.2 Transparency and Explainability

Given that LLMs and multi-modal embeddings are often considered "black-box" systems, transparency is critical—especially when the system influences consumer purchasing behavior. Because the calculations for product similarity are done in the back end of the various embedding models, our applications decisions are not always easy to interpret.

To make our recommendations more transparent, we included interpretable outputs in the UI and ensured that each product output includes the features most relevant to the user's original query—such as price, product style, and rating. Additionally, our use of an interactive UI through Streamlit makes it easier for users to see how their input query leads to the recommended outputs. In the future, adding a brief explanation of why each product was selected—based on similarity score or constraint matching—could further enhance user understanding and trust. However, further improvements such as providing rationale for rankings or similarity scores could enhance user trust and system accountability.

8.3 User Autonomy

AI-powered recommendation systems can shape consumer choices in subtle ways. By filtering and ranking products, our assistant has the power to influence purchasing behavior. While our goal is to assist users in finding relevant items, we recognize the importance of maintaining user autonomy. We designed the system to provide multiple product options, rather than a single definitive answer, so that users can compare and decide for themselves. This design choice helps preserve user agency and avoids reinforcing narrow consumption patterns.

In summary, our multi-modal recommendation system offers valuable improvements to product search, but must be developed and deployed with care. Data privacy, transparency, and respect for user autonomy are critical to building a system that not only performs well but is also responsible and user-aligned.

9 Future Work

While our current multi-modal search assistant demonstrates promising performance and usability, several directions remain for future enhancement to improve scalability, generalization, and real-world applicability.

9.1 Enable Image-Based Query Input

At present, the system only accepts natural language queries as input. A natural extension would be to allow users to upload or capture an image of a product they are looking for—for example, by snapping a photo of a friend's lipstick or a desired packaging design. By embedding this input image and matching it against the existing product image embeddings, the assistant could return visually similar product suggestions, further enhancing discoverability and convenience.

9.2 Expand Dataset for Better Coverage and Realism

The current system is trained and evaluated on a curated subset (11k products) from a public dataset. However, it does not yet include real L'Oréal catalog data. Incorporating a larger and more representative dataset—including actual product titles, descriptions, pricing, packaging images, and customer reviews—would improve both the breadth and precision of recommendations. Partnering with L'Oréal to access or simulate a real product inventory would help bridge this gap and test the assistant in production-like environments.

9.3 Improve Evaluation with Diverse Metrics

Our LLM-based evaluation framework provides a novel way to assess result quality, but it remains qualitative and prompt-sensitive. In future iterations, incorporating additional quantitative evaluation metrics—such as precision@k, recall@k, nDCG (Normalized Discounted Cumulative Gain), or embedding-based retrieval accuracy—would provide a more comprehensive view of system performance. A human evaluation study (e.g., A/B testing with actual users or domain experts) could also validate alignment with user preferences.

9.4 Fine-Tune on L'Oréal-Specific Data and Vocabulary

Although our current system leverages general-purpose models like Sentence Transformers and CLIP, their embeddings are trained on open-domain data. Fine-tuning these models on L’Oréal’s proprietary product language, industry-specific terminology, or even internal taxonomy would significantly improve relevance and brand consistency in recommendations. This would help the assistant better capture subtle product distinctions (e.g., finish, formulation, target audience) often lost in generic embeddings.

10 Acknowledgements

We are sincerely thankful to the L'Oréal Tech Accelerator team, Rémi Ferreira and Nicole Brye, as well as Professor Sining Chen for their mentorship, guidance on project execution and invaluable feedback throughout the course. We also want to thank Savannah Thais for her engaging lectures on AI ethics and the homework assignments that deepened our understanding of responsible data science. Additionally, we appreciate the teaching assistant - Phoebe Chen for her time and support in advising us on course concepts and assisting with project inquiries.

11 References:

[1] A. B. Modi, I. Pundir, S. Sajjad, Y. A. Dange, and Z. Shakruwala, "Vector RAG vs. GraphRAG for AI-powered Chatbots and Product Recommendations," Columbia University Capstone Project, Spring 2023. Available: https://github.com/engie4800/dsi-capstone-fall-2024-loreal-ragvsgraphrag

[2] A. Rastogi, L. Uzzan, N. Ali, R. Gupta, T. Badhwar, and T. Prasad, "Minimizing Hallucinations in AI-Generated Product Recommendations: A Comparison of Vanilla RAG and GraphRAG," Columbia University Capstone Project, Fall 2024. Available: https://github.com/engie4800/dsi-capstone-fall-2024-loreal-rag-capstone

[3] Studeni, "AMAZON-Products-2023 Dataset," Hugging Face, 2023. Available: https://huggingface.co/datasets/Studeni/AMAZON-Products-2023

[4] S. Witalec, "Building Multimodal Search and RAG," DeepLearning.AI, 2024. Available: https://www.deeplearning.ai/short-courses/building-multimodal-search-and-rag/

[5] A. Radford et al., "Learning Transferable Visual Models From Natural Language Supervision," Proceedings of the International Conference on Machine Learning (ICML), 2021. [CLIP paper] Available: https://arxiv.org/abs/2103.00020

[6] J. Li et al., "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," European Conference on Computer Vision (ECCV), 2022. Available: https://arxiv.org/abs/2201.12086

[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," EMNLP, 2019. Available: https://arxiv.org/abs/1908.10084

[8] Facebook AI, "FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors," 2017. Available: https://github.com/facebookresearch/faiss

[9] Y. Liu et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv preprint, 2020. [RAG original paper] Available: https://arxiv.org/abs/2005.11401