

# From Databases to Dishes: Recipe Generation using RAG (Retrieval Augmented Generation)

**Pallavi Udatewar**  
updatewar@usc.edu

**Mahema Reddy**  
mnelatur@usc.edu

**Lakshmi Manasa**  
medepall@usc.edu

**Aditya Jadhav**  
adityadh@usc.edu

**Prathamesh Koranne**  
koranne@usc.edu

## 1 Introduction

A unique feature of human intelligence is the ability to seamlessly combine task-oriented actions with verbal reasoning. Consider the example of cooking up a dish in the kitchen. Between any two specific actions, we may reason in language in order to track progress (“now that everything is cut, I should heat up the pot of water”), to handle exceptions or adjust the plan according to the situation (“I don’t have salt, so let me use soy sauce and pepper instead”), and to realize when external information is needed (“how do I prepare dough? Let me search on the Internet”). The important point of the above quote (and in fact the paper) is the intention to combine two powerful abilities of LLMs reasoning (e.g. chain-of-thought prompting) and acting (e.g. action plan generation). While the former helps with improving the accuracy of an individual task, the latter provides the LLM power to perform multiple tasks. The plan is quite simple ask LLM a question (input) and let it “plan” what to do next (action) by reasoning on the input (thoughts). It can even propose values to the action (action input). Once we perform the action and get an output (observation) the LLM can again reason (thought) to propose the next “plan” (action). In a way, we keep the LLM busy, until it terminates and we get the result we wanted. To summarize we iterate over “input → thoughts → action → action input → observation → thoughts”. The distinguishing factor of our approach lies in the innovative techniques embedded within Retrieval Augmented Generation. Leveraging advanced algorithms, we have devised a system that not only retrieves relevant data from databases but also generates contextually appropriate content for culinary applications. This dual functionality sets our project apart, offering a comprehensive solution that goes beyond the capabilities of existing methods.<sup>1</sup>

<sup>1</sup><https://github.com/pallaviupdatewar/Databases-to-Dishes>

## 2 Datasets

In order to address the limited availability of high-quality public datasets, we took the initiative to create our own dataset. To accomplish this, we employed web scraping techniques on popular food recipe websites like Allrecipes, Epicurious, Bon Appétit, New York Times Cooking, and others. To navigate the complexities of each website’s structure, we developed custom scrapers tailored to each site. To enhance the robustness of our scraping process, we integrated advanced techniques such as proxies and headless browsers. Following data acquisition, we applied preprocessing methods to ensure uniformity and consistency across all the collected data points. This involved transforming the data into a standardized format. As an example of a typical data point within our dataset, imagine a single recipe entry containing information like ingredients, cooking instructions, and other relevant details, all neatly organized and ready for analysis. By creating this comprehensive dataset, we aim to facilitate research and analysis in the realm of food recipes and cooking, making it easier for researchers and enthusiasts to explore culinary data and trends.

## 3 Methodology

### 3.1 Data Pre-processing

Following data acquisition, we applied preprocessing methods to ensure uniformity and consistency across all the collected data points. This involved transforming the data into a standardized format. As an example of a typical data point within our dataset, imagine a single recipe entry containing information like ingredients, cooking instructions, and other relevant details, all neatly organized and ready for analysis. By creating this comprehensive dataset, we aim to facilitate research and analysis in the realm of food recipes and cooking, making it easier for researchers and enthusiasts to explore culinary data and trends.

### 3.2 Implementation of BM25 for Ingredient-Based Search

1. BM25 Algorithm  
BM25, a probabilistic retrieval function, is employed to rank recipes based on ingredient similarity. This algorithm is well-suited for handling sparse and unstructured textual data, like ingredients lists.
2. Search Functionality  
A custom function is developed to perform ingredient-based searches using the BM25 algorithm. This function takes a list of ingredients as input, processes the query, and returns a ranked list of recipes with the most relevant ingredients.

### 3.3 Embedding Generation and Vector Store Creation

1. Embeddings with OpenAI  
OpenAIEmbeddings are utilized to convert textual data into numerical representations, capturing semantic similarities. This conversion is essential for effective retrieval and comparison of recipes based on content.
2. Vector Store with FAISS  
FAISS, a library for efficient similarity search, is employed to create a vector store of recipe steps. This store enables quick and relevant retrieval of recipes based on the embeddings generated in the previous step.

### 3.4 Contextual Retrieval and Document Compression

1. Large Language Model (LLM) Setup  
An instance of the OpenAI model is initialized, configured to operate with a specific temperature setting. This LLM serves as the backbone for generating and refining recipe content.
2. Document Compression for Relevance  
The LLMChainExtractor is implemented to compress and refine the retrieved documents. This step ensures that only the most relevant and concise content is fed into the LLM, enhancing the quality of the generated recipes.
3. Contextual Compression Retrieval  
A Contextual Compression Retriever is developed, combining the document compressor and the retrieval mechanism. This retriever is

key in obtaining relevant and concise recipe information, tailored to specific ingredient queries.

### 3.5 Recipe Generation and QA Pipeline

1. Retrieval-QA Chain  
The RetrievalQA chain integrates the retrieval and compression functionalities. It is responsible for generating recipes in response to specific queries, considering user-specified constraints and preferences.
2. Query Processing  
The system is designed to handle queries that specify particular ingredients and constraints. This functionality allows for the generation of customized recipes, adhering to user requirements and dietary preferences.

## 4 Experiments

The experimental setup is designed to explore and evaluate the performance of a recipe generation system comprising three key components: Recipe Search, Embedding and Retrieval, and Recipe Generation with GPT-4. The aim is to assess the effectiveness and creativity of the system in generating relevant and diverse recipes based on user-input ingredients. The experimental phases are outlined below:

1. Recipe Search:  
BM25Okapi Model: The first phase involves querying the BM25Okapi model with a pre-defined list of ingredients to retrieve relevant recipes. The BM25Okapi model is chosen for its efficiency in information retrieval tasks.  
Assessment of Relevance and Diversity: Following recipe retrieval, the relevance and diversity of the obtained recipes are assessed. Metrics such as relevance score and diversity index may be used to quantify the quality of the retrieved recipes.
2. Embedding and Retrieval:  
OpenAIEmbeddings: Recipe steps are converted into embeddings using the OpenAIEmbeddings model. These embeddings serve as numerical representations of the textual information in recipe steps.  
FAISS Retrieval: The embeddings are utilized to create a retriever using the FAISS library, enabling efficient similarity-based retrieval. This phase focuses on enhancing the precision and speed of recipe retrieval.

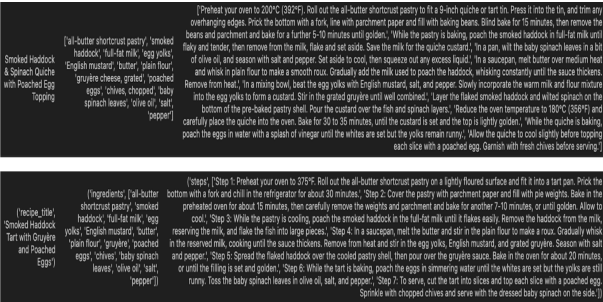
3. Recipe Generation with GPT-4:  
GPT-4 Model: The GPT-4 model is employed to generate recipes based on the input ingredients. The generation process involves leveraging the contextual understanding and language generation capabilities of GPT-4.
4. Comparative Analysis with GPT-4 vs. Rag-Pipeline:  
Rag-Pipeline Comparison: A comparative analysis is conducted between recipes generated by GPT-4 and those produced by the RetrievalQA pipeline, specifically the Rag-Pipeline. Analyzing Differences: Differences in terms of ingredient use, recipe structure, and novelty are scrutinized to understand the strengths and weaknesses of each approach.
5. Ingredient Accuracy Assessment:  
Deviation Analysis: A dedicated function is implemented to count extra ingredients present in the generated recipes compared to the original input list. Frequency Calculation: The frequency of deviation from the original ingredient list is calculated, providing insights into how often the generated recipes introduce additional elements not specified in the input.

5 Evaluation

1. Evaluation Metrics:  
Given our limitations in evaluating open-ended generation, we concentrate on a more tangible aspect: analyzing the additional ingredients introduced by our recipe generator. Human preference modeling, constrained by budget and computational considerations, is set aside in favor of a more practical assessment that provides insights into the generator’s behavior concerning ingredient utilization. Therefore, we establish a unique metric considering the variance in the number of ingredients between the generated recipe and the originally provided ingredients. This metric produces a normalized representation of the disparity, and the average is computed across all samples for comprehensive evaluation.
2. A higher score indicates more deviation from the original list, implying the AI model adds unexpected ingredients, which might not be ideal for users needing strict adherence to their

ingredient list. Conversely, a lower score signifies closer adherence to the original ingredients, preferred when users expect recipes to strictly follow their provided list. A lower score means that the generated recipes closely adhere to the original list of ingredients, with fewer additional ingredients being introduced. This is generally more desirable in scenarios where users expect the recipes to be strictly based on the ingredients they provided. A low score would indicate better performance of the model in this respect.

6 Results



As seen in the steps shown above, the first image which corresponds to the RAG architecture shows a greater level of detail whereas the steps in the second image (corresponding to GPT-4) shows a lack of culinary depth that one would expect from a recipe card.

In the recipe provided from GPT-4, we see a level of creativity and complexity that is typical of GPT models. The recipe for "Smoked Haddock, Spinach Quiche with Poached Egg Topping" includes a diverse range of ingredients and detailed cooking steps. This reflects GPT-4’s ability to generate rich, detailed content, but it can sometimes lead to the inclusion of additional or less common ingredients, as the model aims to create a unique and flavorful dish.

On the other hand, a RAG model would likely generate a recipe that is more straightforward and potentially simpler, drawing directly from existing recipes in its database. The ingredients and steps would be more in line with standard recipes, potentially making it more practical but less creative.

The results from the analysis suggest that, on average, recipes generated by the RAG model are more faithful to the original ingredient list compared to those created by GPT-4. Specifically, the average extra ingredient score for GPT-4 is 0.482, indicating that GPT-4 tends to add nearly

half an ingredient not found in the original list per recipe. In contrast, the RAG model's score is lower at 0.372, demonstrating a closer adherence to the provided ingredient list. This finding aligns with the inherent differences between the two models. GPT-4, as a generative model, is designed to innovate and create new combinations, leading to the inclusion of additional ingredients that may enhance the recipe's uniqueness and flavor profile. On the other hand, the RAG model, with its retrieval-augmented approach, is more likely to stick closely to existing recipe formats and ingredient lists, resulting in fewer deviations. In practical terms, for applications where sticking closely to a given list of ingredients is crucial—such as in situations with dietary restrictions, budget constraints, or limited pantry options—the RAG model would be more suitable. Conversely, for culinary experimentation and more creative recipe generation, GPT-4's approach might be preferred despite its higher tendency to introduce extra ingredients.

## 7 Applications

Recipe generation finds applications across diverse domains. In the culinary realm, it inspires home cooks with fresh ideas, aids chefs in menu planning for restaurants, and serves as an educational tool in culinary schools. Beyond the kitchen, it contributes to personalized meal planning, helping individuals adhere to dietary preferences and nutritional goals. Additionally, recipe generation has practical implications in the food industry by assisting product development, reducing food waste through creative ingredient utilization, and providing content for food bloggers and influencers. It also plays a role in the automation of cooking processes in smart kitchen appliances, offering users a wide array of meal options tailored to their preferences.

Moreover, recipe generation fosters cultural exploration by introducing users to global cuisines. Its potential extends to health and nutrition, promoting the creation of healthier recipes. In event planning and hospitality, it assists in crafting diverse menus for gatherings. Overall, the technology's adaptability makes it a valuable tool for both individuals seeking culinary inspiration and businesses aiming to innovate in the food industry.

## 8 Conclusions

The study conducted on AI models in recipe generation and retrieval, using a combination of

BM25Okapi, OpenAIEmbeddings and FAISS, and GPT-4, has provided valuable insights into the potential of AI in the culinary domain.

The BM25Okapi model proved effective in sourcing recipes based on specific ingredients, offering a practical tool for home cooks and those with dietary restrictions. Meanwhile, the integration of OpenAIEmbeddings and FAISS showcased the AI's capability in intelligently linking recipe steps with ingredients, suggesting the possibility of more intuitive and user-centric recipe recommendation systems.

The use of GPT-4 for creative recipe generation was a highlight of the study, demonstrating the model's ability to craft unique recipes from given ingredients. This opens up exciting opportunities for personalized cooking guides and automated recipe creation. A comparative analysis between GPT-4 and the RetrievalQA pipeline revealed distinct strengths in each system, with variations in ingredient fidelity, recipe structure, and innovation. However, the study also noted challenges in maintaining ingredient accuracy, underscoring the need for further refinement in AI models for enhanced precision and relevance.

In conclusion, this study has demonstrated that AI has significant potential in the culinary field, from assisting in recipe retrieval to creating novel recipes. However, it also highlighted the need for further refinement in AI models to enhance accuracy and relevance. These findings pave the way for future research and development in AI-powered culinary tools, promising to revolutionize how we explore and experience food.

## References

- Kelvin Guu and [other authors' names]. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- LangChain Developers. 2023. [Retrieval using langchain](#). Accessed on Date of Access.
- Patrick Lewis, Mark Neumann, Tim Rocktäschel, Jonathan Schwarz, and Sebastian Riedel. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

David Noever and [other authors' names]. 2023. The multimodal and modular ai chef: Complex recipe generation from imagery. *Journal Name (if available)*.

Revathy Venkataramanan. 2023. Cook-gen: Robust generative modeling of cooking actions from recipes. In *Conference Name or Journal Name (if applicable)*.