Persian Culinary RAG: Multimodal Retrieval and Generation for TextImage Food Queries

MohammadHossein Eslami,Arshia Izadyari,Sadegh Mohammadian,Fateme Askari
Ali RahimiAkbar,MohammdMahdi Vahedi
https://github.com/NLP-Final-Projects/Food_rag_3

Abstract

We present a Persian, food-domain retrieval-augmented generation (RAG) system that combines a dual-modality retriever with a lightweight generator. Building on our prior corpus and an added Kaggle recipe collection (1,737 entries; 1,393 unique dishes), we expand the index with web-sourced photos of dishes *and* systematically collected images of key ingredients to strengthen image-grounded queries. The retriever pairs a Persian text encoder (Glot-500) with a fine-tuned CLIP vision–text encoder (vision-fa-clip/SajjadAyoubi) trained with a multi-positive contrastive objective to handle multiple instructions per dish. Cross-modal embeddings enable answering both text-only and image+text questions by retrieving pertinent evidence and conditioning the generator. On held-out multiple-choice sets, the RAG setup improves performance for ingredient-triggered and image-grounded queries with a lighter generator, while gains are mixed for a stronger generator.

1 Introduction

Retrieval links user queries to relevant evidence, enabling grounded and controllable answers in knowledge-intensive settings (**?**). In the culinary domain, retrieval is especially helpful for (i) surfacing recipes and procedural steps for a named dish, (ii) answering ingredient- or diet-centric questions (e.g., substitutions, halal/vegan constraints), (iii) disambiguating regional variants and homonymous dishes, and (iv) bridging images and text when a photo of plated food, garnish, or a raw ingredient is the primary cue. Pairing retrieval with generation further reduces hallucinations by conditioning responses on retrieved context, including cross-modal evidence (**?**).

Related work. Early cross-modal food retrieval learns joint embeddings of recipes and images (e.g., Recipe1M/Recipe1M+) to support image→recipe and recipe→image search (**??**). Vision–language models such as CLIP enable scalable alignment via contrastive learning (**?**), and multilingual extensions (e.g., M-CLIP) broaden applicability beyond English (**?**). Retrieval-augmented generation (RAG) integrates a retriever with a generator to answer knowledge-heavy queries while citing evidence (**?**). However, prior food-retrieval work focuses on English corpora and finished-dish photos; ingredient-centric imagery and Persian-language pipelines remain underexplored, and label collisions (multiple instructions per dish) are rarely addressed explicitly.

This work. We develop a Persian, food-focused RAG system that unifies text-only and image+text queries through a dual-modality retriever and a lightweight generator. Beyond collecting dish photos, we systematically attach *ingredient images* to each entry to strengthen image-grounded and ingredient-triggered queries. The retriever combines a Persian text encoder with a fine-tuned vision–text encoder trained using a *multi-positive* contrastive objective that treats all instructions for the same dish as positives, mitigating false negatives from near-duplicates and visually similar foods.

Contributions. The primary contributions of our work are outlined as follows:

- **Ingredient-aware cross-modal index.** We construct a Persian culinary index that pairs texts with photos of *both* finished dishes and their salient ingredients, improving retrieval when the visual cue is a raw material or garnish rather than a plated meal.
- **Label-aware contrastive training.** We adopt a multi-positive objective that groups all instances sharing a dish label as positives, reducing false negatives caused by multiple instructions per dish and stabilizing CLIP-

style fine-tuning in this domain.

- **Retriever→generator interface for Persian QA.** We provide a generator-agnostic conditioning layer with Persian prompt templates that consistently injects retrieved cross-modal evidence for text-only and image+text questions.
- **Reproducible pipeline.** We release end-to-end scripts for data acquisition, indexing, and evaluation (text-only and image+text multiple-choice sets), enabling faithful replication without manual curation at test time.

We next describe the dataset construction and enrichment, the retrieval architecture and training objective, and the evaluation protocol, then discuss design choices specific to visually similar foods and regional variants.

2 Methods

This section outlines the core components of our system: a shared imagetext representation, retrieval with evidence pooling in a RAG setting, and answer selection for multiple-choice questions. We also describe how ingredient photos are used to strengthen cross-modal alignment.

2.1 Shared Representation Learning

Vision encoder. A CLIP-style vision tower maps RGB images to fixed-length feature vectors. Prior to encoding, images are EXIF-corrected and normalized with the models official preprocessing. All visual embeddings are L2-normalized so that inner product coincides with cosine similarity.

Text encoder. A transformer-based text encoder maps passages and questions into the same embedding space. Depending on the backbone, we use the models or masked mean pooling over the last hidden states, followed by L2 normalization. To reduce domain shift, the text encoder is fine-tuned on in-domain imagetext pairs (culinary descriptions paired with dish and ingredient photos), improving alignment to the visual distribution.

Dimensional compatibility. We verify that the vision and text heads produce embeddings with matching dimensionality at initialization to ensure that image-to-text comparisons are well-defined in a single shared space.

2.2 Retrieval and Evidence Pooling

Indexing. We maintain FAISS indices (`IndexFlatIP`) over L2-normalized embeddings. A CLIP-text index is built from all corpus passages and supports both text-to-text and image-to-text search. An additional high-recall text-only retriever is used for purely textual queries. Indices are persisted and memory-mapped at runtime.

Initial retrieval. Given a query, we form an evidence pool by retrieving top-k passages:

- **Text-only queries:** the question is encoded with the text encoder and searched against the CLIP-text index (and optionally the text-only retriever).
- **Image queries:** the image is encoded with the vision encoder and searched against the CLIP-text index (image→text).

Option-aware expansion for MCQ. For multiple-choice questions, we issue auxiliary sub-queries of the form *question option: X* to collect option-specific evidence. The union of hits is deduplicated by a dish-level key to avoid clusters of near-duplicates and preserve semantic diversity.

2.3 RAG Prompt Construction

Snippet selection. From the pooled results we extract short snippets (200350 characters) and retain their document identifiers. Snippets are concatenated into a compact evidence block (with `[doc:id]` citations) and injected into the prompt. For image queries, the image is supplied alongside the snippets.

Prompting modes. We consider two modes. In **NoRAG**, the model receives only the question (and the image if present). In **RAG**, the model also receives the retrieved snippets. Prompts explicitly instruct the model to output exactly one of the provided options; the raw output is then coerced to a canonical option string.

2.4 Ingredient Photos

Training-time role. Each dish may include up to three ingredient photos. During fine-tuning, these images are treated as additional positives for the dishs text, encouraging alignment between the textual description and the *set-level concept* of the dish (final plating and its ingredients), rather than a single viewpoint.

Inference-time behavior. At inference, every imagewhether a final dish or an ingredient photois encoded independently and used to query the text index. Multi-positive fine-tuning makes ingredient-driven queries more robust. As an optional enhancement, one can aggregate a dishs ingredient embeddings into a prototype (e.g., mean or attention pooling) and index these prototypes for set-level matching.

2.5 Answer Selection

Visionlanguage head. A visionlanguage model (e.g., LLaVA-style) consumes the prompt (question, options, RAG snippets, and image if present) and generates a single option as the answer.

Heuristic head and fusion. A lightweight heuristic converts lexical matches between each option and the concatenated snippets into a calibrated confidence score. This head can serve as a fallback or be combined with the VLM output via simple tie-breaking or weighted fusion. The final output includes the chosen option, a confidence score, and supporting document IDs to enable transparency and error analysis.

2.6 Dataset

To construct a dataset of Iranian foods from scratch, we designed a multi-stage process aimed at supporting multimodal models. The first step involved collecting data from the Internet. Using web-scraping tools such as Selenium and BeautifulSoup, we extracted information on a wide range of Iranian dishes. The gathered data encompassed several aspects: in addition to the fundamental components of a recipe (ingredients and preparation instructions), we also collected metadata such as the city of origin, the cultural or social occasions in which the dish is typically served, and the meal type (e.g., appetizer, main dish, or dessert). After obtaining this raw collection, we employed a large language model (LLM), specifically GPT, to clean and standardize the data. One of the main challenges in this step was the structural inconsistency of the scraped information, as there is no universally accepted format for documenting recipes. Furthermore, we encountered variations in linguistic structure across different web sources, which could not be addressed through fixed rule-based processing. Leveraging an LLM enabled us to resolve these issues and impose a consistent structure on the dataset.
Once each dish had a well-structured entry, the next step was to create a dedicated document for every food item. Since our system is designed to include a retriever, it requires a collection of documents through which the retriever can search to answer user queries. To generate these documents, we employed another LLM (Gemma3) to produce fluent and coherent passages describing each dish, based on its corresponding JSON entry. This approach ensured that the resulting passages were clean, well-formed, and required no further normalization. Because the system is intended to answer questions grounded in these documents, it was also necessary to generate training questions. To this end, we used an additional prompting strategy to produce at least five questions per passage, each of which could be directly answered from the associated text. This pairing of passages with corresponding questions allowed the model to learn richer document embeddings and better recognize queries related to a specific dish.

To further enhance the retriever, we incorporated an additional modality: images. Using API calls, we collected a set of images for each dish, enabling the model to develop a richer understanding of the visual characteristics of Iranian foods. These images were then paired with the corresponding textual documents, allowing the retriever not only to embed visual information alongside text but also to handle queries where an image is provided as input. For each dish, we curated a small set of representative images, including both the final prepared dish and its three main ingredients. This multimodal pairing allows the model to capture both the overall appearance and the essential components of each dish, thereby strengthening its ability to retrieve and answer queries more accurately.

It is important to note that, aside from normalization, no additional preprocessing was applied to the dataset. For images, preprocessing is handled directly by the model at the input stage, eliminating the need for manual intervention. For textual data, common preprocessing techniques such as lemmatization and stemming were deliberately avoided. While such methods can simplify text, they may also lead to information loss, which could negatively affect the performance of our retrieval-augmented generation (RAG) system in downstream tasks. Since RAG relies on accurately retrieving and grounding responses in the original documents, preserving the full linguistic richness of the data ensures that the retriever maintains access to all potentially informative features [1].

2.7 Divided Work

- Arshia Izadyari: developed the code, prepared the github (code section), ran the experiments
- Mohammad Hossein Eslami: prepared the dataset, wrote the evaluation, challenges and conclusion/future works
- Sadegh Mohammadian: Prepared the demo, prepared the evaluation
- Fateme Asgari: Helped with the dataset, wrote the abstract, introduction and method
- Ali Rahimi Akbar: Helped with the evaluation part of the report, helped with code development
- Mohammad Mahdi Vahedi: helped with model development, helped with evaluation and its report

3 Experiment

In this section, we prepared two sets of multiple-choice questions along with answers to evaluate our model. The first set, which consisted purely of text-based questions, included 30 questions covering several categories and types of questions. Some of these questions were designed about the ingredients needed to prepare a dish. Others focused on the cooking process, while another group asked about the geography of the food. We also had a number of questions about the occasions associated with different foods.

3.1 Evaluation Dataset

The second set, which contained 20 questions, included items accompanied by images. The questions in this set were based on the images and had no meaning on their own without them. These questions were also divided into categories. Some asked about what the food was simply by looking at the picture. Others attempted to identify the geographical origin of a dish from its image. One important category of questions involved giving the main ingredient of a dish and asking for the name of the dish. Finally, a number of more specialized questions tried to identify the decorative ingredient of a dessert by looking at its picture.

We provide some representative examples of the questions in Appendix A.

3.2 Experimental Setup

To evaluate the efficacy of retrieval augmentation, we constructed a multimodal Retrieval-Augmented Generation (RAG) framework designed for a Persian food Visual Question Answering (VQA) task. The system's retrieval component employs modality-specific pathways. For text-based queries, a fine-tuned Glot-500 (?) sentence transformer is used to encode questions into a shared embedding space with the document corpus. For image-based queries, a CLIP-based dual-encoder architecture maps visual inputs to the same textual embedding space (?). The knowledge base consists of a corpus of text passages detailing Persian cuisine, which was indexed using FAISS to enable efficient, high-speed similarity searches. For the generative stage, two distinct Large Multimodal Models (LMMs) were evaluated: LLaVA 1.5 (?) and Gemini Pro 2.5 (?). The end-to-end performance was benchmarked on a custom VQA dataset, comparing a baseline configuration that relied solely on the generator's parametric knowledge against the RAG-enabled configuration, which provided dynamically retrieved context to the generator. This comparative setup allowed for a quantitative assessment of the retrieval component's impact on task accuracy. All computational tasks, from model fine-tuning to final evaluation, were executed within the Kaggle environment, leveraging an NVIDIA T4 GPU.

3.3 Result Evaluation

The results shown in table 1 indicate that our model performs more effectively when handling text-only queries compared to those that involve images. Several factors contribute to this outcome. First, both documents and questions are encoded into the same representation space, which facilitates alignment when only textual inputs are considered. However, due to the limited availability of images in our dataset, the encoder struggles to project visual inputs into the same space as the textual documents.

Second, encoding images is inherently more

[1] The image of food distributions in Iran is appended in the end of the report

Generative Model	Model	Text-only	Image Question
LLaVA 1.5	Generative-only	26.67	20
	RAG	36.67	25
Gemini 2.5	Generative-only	53.33	40
	RAG	46.67	30

Table 1: Accuracies for two models across Text-only and Image questions, with the Generative model only, and Fine-tuned Retrieval settings for getting results.

challenging than encoding text. While textual queries can often be mapped to documents by leveraging key lexical cues, an image represents new and complex data that requires deeper feature extraction. This complexity makes it more difficult for the retriever to correctly identify the relevant document.

Dataset limitations further exacerbate this challenge. The relatively small size of the fine-tuning dataset restricts the models ability to generalize effectively. Moreover, certain Persian dishes exhibit high visual similarity (e.g., Ghorme Sabzi and Fesenjan) or have notable regional variations in preparation and appearance. As a result, the model often lacks sufficient information to reliably distinguish between visually similar dishes and can, at best, classify them at a broader categorical level. Taken together, these factors explain the discrepancy in performance between text-based and image-based queries.

3.4 Question Type Evaluation

When evaluating the models performance on different question types involving images, a clear pattern emerges. The image-based questions can be categorized as follows:

- Classifying a food from an image
- Determining the origin of a food based on its image (geographically)
- Food recommendation based on an image of ingredients
- Identifying the ingredients of a food from the final image of the dish

Our analysis indicates that the model performs best when the input image represents the final prepared form of a dish. This outcome is expected, as a substantial portion of our fine-tuning dataset consists of images depicting the final presentation of foods. While images of individual ingredients were also included, these tend to be less distinctive, as many dishes share the same ingredients. Consequently, embeddings derived from ingredient images are less specific than those from final-dish images. Additionally, we observed that simpler questions improve performance, as they require less reasoning and reduce the potential for error in both retrieval and generation. The portion of answered questions can be seen in table 2. It is evident that the model achieves higher performance when provided with the complete image of a dish, as reflected in the classification results. In contrast, its performance drops when dealing with ingredient images or when answering questions specifically about ingredients. Furthermore, the results in the recommendation column show that the model struggles with questions that demand substantial reasoning.

Focusing on questions with text as the only input, we observe a greater variety in question types, although the differences in model performance across these types are smaller than in the previous evaluation group. The evaluation dataset includes several categories, such as:

- Food recommendation based on ingredients or location
- Differences between the same dish originating from different regions
- Meal type classification (e.g., breakfast, lunch, dinner)
- Questions regarding preparation instructions
- Questions regarding ingredients

As noted previously, the performance gap across these categories is relatively modest. This can be attributed to the models ability to embed questions effectively based on familiar keywords. In this set of evaluation questions, the model generally encounters terms it has seen during fine-tuning, which reduces the challenge of retrieval and reasoning. Nevertheless, differences in performance are still observed for questions that require higher-order reasoning. Specifically, when a question contains keywords present in the fine-tuning dataset, the model performs significantly better than when the question involves unfamiliar terms or necessitates logical connections. In other words, the model is most challenged when it must integrate multiple pieces of information within a question or relate them to the content of a document. The portion of answered questions can be seen in table 3. As observed, the model performs much better on questions that can be answered directly from the text, such as those about ingredients or preparation. However, its performance drops noticeably on questions that require reasoning, such as food recommendations.

3.5 Models

For the generative component of our RAG system, we used LLaVA 1.5 and Gemini 2.5. Our experiments show that Gemini 2.5 Pro achieves significantly better results, primarily due to its strong reasoning capabilities. However, a key limitation we observed is that Gemini 2.5 Pro relies less on the retrieval component, likely because many of the evaluation questions are already within its internal knowledge base. In contrast, LLaVA 1.5 does not exhibit this behavior, which allows the retrieval component to contribute more effectively to improving accuracy.

As reflected in the results, our full pipeline using LLaVA 1.5 shows an overall decrease in performance compared to pipelines with stronger base models. However, a notable observation is the improvement introduced by our retriever in this setting. When paired with Gemini 2.5 Pro, the retriever occasionally acted as an adversary, introducing noise that confused the generator. Because Gemini 2.5 Pro is a strong model trained on a large and diverse dataset, it can achieve high performance on its own. Yet, when it is provided with an incorrectly retrieved document, the model shifts some of its attention away from its internal knowledge and attempts to integrate information from the noisy input, which in turn degrades performance on certain questions.

In contrast, LLaVA 1.5 is a comparatively weaker model, lacking the extensive training data and capabilities of Gemini 2.5 Pro and also not fine-tuned on Persian VQA. As a result, its standalone performance is limited. In this case, however, the addition of the retriever proves beneficial: by supplying relevant external knowledge, the retriever compensates for some of LLaVAs weaknesses and boosts its performance. While the retriever itself is still far from optimal, it provides enough additional context to improve the weaker generators output beyond what the generator could achieve on its own.

4 Challenges

The first challenge arose during data gathering and dataset construction. Although we worked with a relatively large number of data entries, the scale was still insufficient for training a fully robust model. To address this, we relied on APIs to collect a substantial number of images. However, this process introduced several difficulties: restrictions and rate limits imposed by image APIs, the low quality of many retrieved images, and the challenge of verifying their correctness. These factors collectively complicated the construction of a high-quality multimodal dataset and limited the overall performance of the system.

Another challenge was encountered during data preparation for fine-tuning. Choosing the appropriate preprocessing strategy was critical, as unsuitable decisions could negatively impact downstream tasks. For example, while common preprocessing techniques such as stopword removal are often applied in natural language processing, they were not suitable in our case. Since our goal was to preserve the full structure and meaning of the documents, removing words risked causing information loss and reducing the effectiveness of the model.

Lastly, a big challenge arose during model development. The selection of modules for different components of the pipeline proved to be particularly critical, as each choice could alter the output and affect overall performance. Since multiple modules were tested, determining the contribution of each to the final result was non-trivial, and even small design variations often led to noticeable changes in behavior. To address this, we carried out careful prompt engineering to refine the interactions between modules and ensure the system achieved the best possible performance.

Generative Model	Model	classifying	geography	recommedation	ingredients
LLaVA 1.5	RAG	2/6	0/1	1/4	2/9

Table 2: Results for LLaVA 1.5 across Image + Text diffrent categories experiments , with Fine-tuned Retrieval settings for getting results.

Generative Model	Model	recommendation	regions	type	prepartion	ingredients
LLaVA 1.5	RAG	2/9	0/2	2/2	2/5	5/12

Table 3: Results for LLaVA 1.5 across Text-only diffrent categories experiments , with Fine-tuned Retrieval settings for getting results.

5 Conclusion/Future Works

In this work, we introduced a multimodal dataset of Iranian foods and demonstrated its use in building and evaluating a retrieval-augmented generation (RAG) system. Our dataset construction pipeline involved several stages, including large-scale web scraping, data normalization with language models, passage generation, and multimodal extension with images. Through evaluation, we showed that while the system performs well on text-based queries, its performance on image-based queries is more limited. This discrepancy stems from challenges such as the scarcity of high-quality images, the difficulty of aligning text and image embeddings, and the relatively small size of our fine-tuning dataset. Additional difficulties were observed in preprocessing, module selection, and handling out-of-distribution queries, all of which impacted downstream performance.

Despite these limitations, our experiments highlight the potential of incorporating multimodal retrieval into the domain of food-related question answering, particularly in culturally rich contexts such as Iranian dishes. Looking forward, several directions can be pursued to strengthen this work. First, expanding the dataset with a larger number of high-quality and verified images would improve the alignment between visual and textual modalities. Second, incorporating region-specific variations of dishes could help the model distinguish visually similar foods and better capture cultural diversity. Third, experimenting with more advanced retrievers and stronger multimodal generators may improve performance by reducing reliance on manual prompt engineering. Finally, integrating human-in-the-loop evaluation and feedback mechanisms could further refine both the dataset and the system.

By addressing these directions, future research can enhance the robustness, generalizability, and cultural coverage of multimodal RAG systems for food-related applications.

Appendix

A Sample Evaluation Questions

A.1 Text-based Questions

These questions rely entirely on textual information. They are designed to test the models ability to:

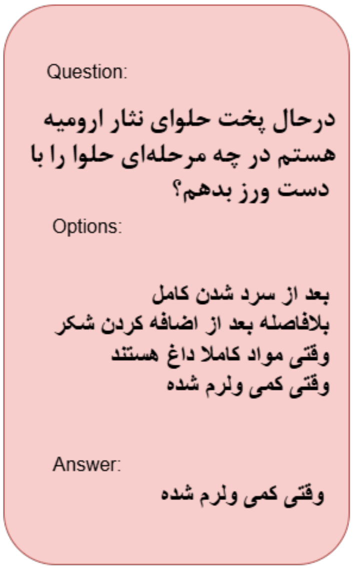
- Understand procedural steps in Persian cooking (e.g., when to knead or mix ingredients).
- Recall key ingredients used in traditional dishes and sweets.
- Recognize dish names associated with particular combinations of ingredients.
- Associate foods with cultural or social occasions.

Together, these items measure the models competence in linguistic comprehension, domain-specific culinary knowledge, and reasoning over text instructions.

A.2 Image-based Questions

These questions require interpreting a visual input in addition to text. They are intended to evaluate the models ability to:

- Identify specific dishes directly from images.
- Link the appearance of a dish to its geographic or cultural origin.
- Infer a dishs name from its main visual ingredient.
- Detect decorative or garnish components in desserts.



Question:

درحال پخت حلوای نثار ارومیه
هستم در چه مرحله‌ای حلوا را با
دست ورز بدهم؟

Options:

بعد از سرد شدن کامل
بلافاصله بعد از اضافه کردن شکر
وقتی مواد کاملا داغ هستند
وقتی کمی ولرم شده

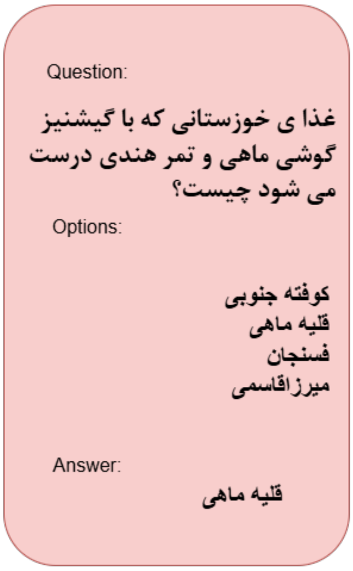
Answer:

وقتی کمی ولرم شده

Figure 1: sample question 1 from evaluation dataset

This category emphasizes multimodal reasoning, cross-modal grounding, and cultural awareness, ensuring that the model can connect visual cues with Persian culinary knowledge.

A.3 Food Distribution



Question:

غذا ی خوزستانی که با گیشنیز
گوشی ماهی و تمر هندی درست
می شود چیست؟

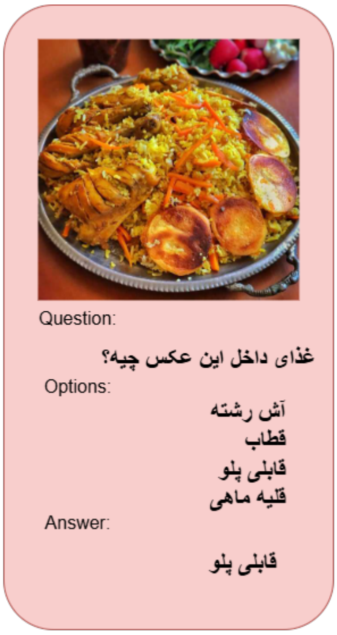
Options:

کوفته جنوبی
قلیه ماهی
فسنجان
میرزاقاسمی

Answer:

قلیه ماهی

Figure 2: sample question 2 from evaluation dataset



Question:

غذای داخل این عکس چیه؟

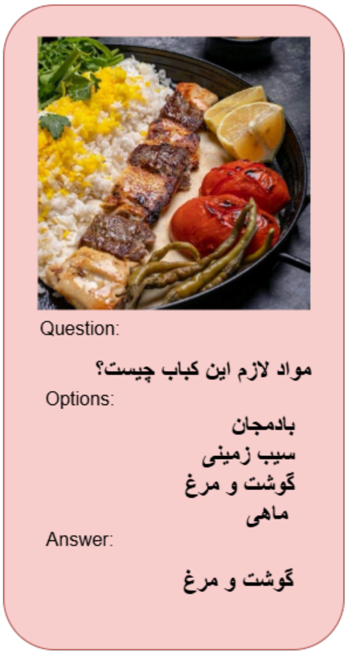
Options:

آش رشته
قطاب
قابلی پلو
قلیه ماهی

Answer:

قابلی پلو

Figure 3: sample question 3 from evaluation dataset



Question:

مواد لازم این کباب چیست؟

Options:

بادمجان

سیب زمینی

گوشت و مرغ

ماهی

Answer:

گوشت و مرغ

Figure 4: sample question 4 from evaluation dataset

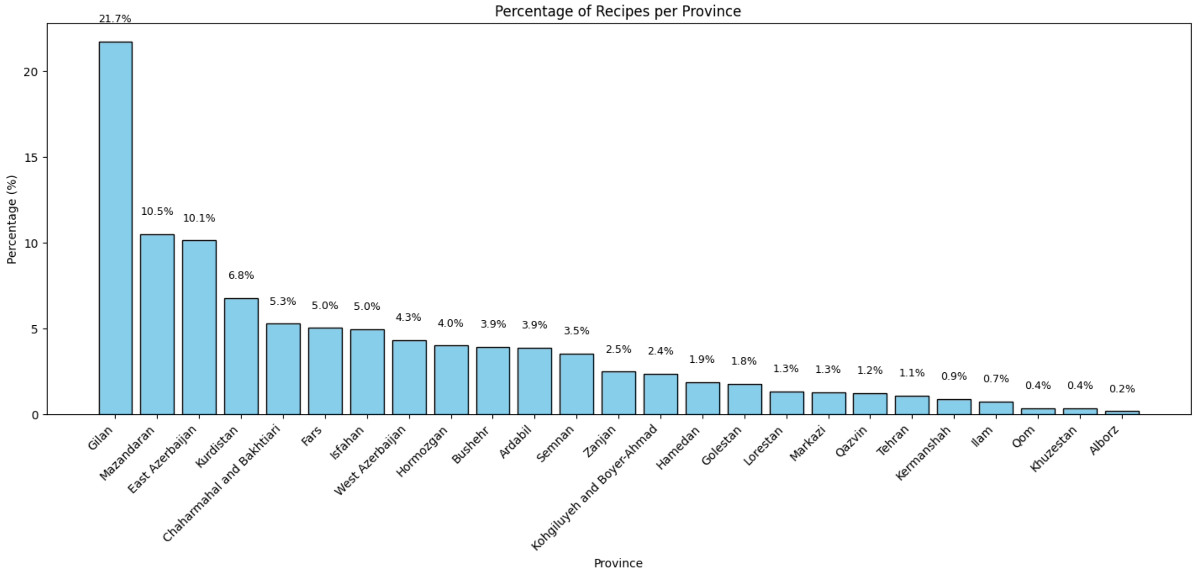


Figure 5: The distribution of foods across provinces