Multimodal RAG for Persian: Retrieval-Augmented Generation over Biographical Corpora of Iranian Public Figures

A PROJECT REPORT

Submitted by

SINA BEYRAMI

SINA DANESHGAR

ELAHE ZAHIRI

ELAHE FARSHADFAR

MARYAM BORZOO

DEPARTMENT OF COMPUTER ENGINEERING
SHARIF UNIVERSITY OF TECHNOLOGY

SEPTEMBER 2025

ABSTRACT

We present a Persian Retrieval-Augmented Generation (RAG) system built over a curated multimodal corpus of Iranian public figures, spanning athletes, politicians, poets, artists, musicians, and scientists. Our pipeline integrates heterogeneous JSON sources, normalizes Persian text, and attaches images to entities; then constructs dual embedding spaces (text via multilingual-e5-base; images via CLIP ViT-B/32) indexed with FAISS/HNSW. Evaluation covers both multiple-choice and open-ended questions, as well as verification metrics (ROC-AUC, EER). To handle the mismatch between biographies and images, we incorporate a face recognition module to detect and filter misaligned cases. The system supports multimodal queries and grounding passages for answer generation. Experiments on $\sim$300 evaluation questions demonstrate that top-k retrieval and cross-modal evidence significantly improve answer accuracy compared to text-only baselines. We release code, indexing scripts, and ablation notebooks.

Keywords: Retrieval-Augmented Generation, Multimodal Retrieval, Persian NLP, CLIP, Multilingual Embeddings, Biographical Corpus, Information Retrieval, Question Answering

Table of Contents

Chapter 1

Introduction

Retrieval-Augmented Generation (RAG) improves factuality by conditioning a generator on retrieved evidence. While RAG is well-studied for English, robust Persian, multimodal RAG remains under-explored. Our goal is to build an end-to-end Persian RAG that: (i) supports text and image evidence; (ii) covers multiple public-figure domains; (iii) uses reproducible preprocessing and indexing; and (iv) is evaluated with task-specific MCQs that require grounded context.

Contributions. (1) A cleaned, merged, and normalized Persian corpus with images for public figures; (2) A dual-encoder retrieval stack (mE5 for text, CLIP for images) with FAISS/HNSW indexing; (3) A multimodal RAG evaluator with MCQ and verification metrics; (4) Reproducible notebooks.

1.0.1 Detailed Description of the Problem

Our project addresses the task of building a multimodal Retrieval-Augmented Generation (RAG) system for Persian, focusing on a corpus of biographies of Iranian public figures. Each record in the dataset consists of a person's name, a short biographical paragraph in Persian, and an associated image. While this provides both textual and visual modalities, a core challenge arises: the biographical text does not usually describe the content of the image. For instance, a biography of a poet may include information about their life, achievements, or historical context, whereas the corresponding image is simply a portrait without textual description. This mismatch complicates joint embedding learning, since aligning vectors of biography texts with vectors of images does not follow the standard cross-modal supervision setting (where text usually describes the image). Therefore, naive approaches that attempt to directly minimize the distance between text and image embeddings may lead to suboptimal or misleading representations.

1.0.2 Challenges and Motivation

This fundamental disconnect between biography texts and images motivates the design of careful retrieval strategies and evaluation metrics. Rather than enforcing alignment, we explore a pipeline that treats text and image encoders independently, while still supporting cross-modal retrieval and grounding. The motivation is twofold:

1. **Practical utility:** Users should be able to query either with text (e.g., a biographical question) or with an image (e.g., a portrait of a person) and retrieve consistent, multimodal evidence.
2. **Scientific interest:** Investigating how far multimodal RAG systems can perform in settings where modalities are semantically related only through entity identity (person-level linkage), not through descriptive text-image pairs. This provides insight into the limitations of current multimodal alignment methods when faced with weak or indirect supervision.

Chapter 2

Literature Review

In multimodal retrieval and question answering, one common challenge arises when images and texts are not directly paired in a descriptive manner. This is exactly the case in our dataset: biographies of public figures are linked to their portraits, yet the textual content rarely describes the visual appearance. Addressing this disconnect between modalities has been the subject of several prior studies.

For example, [1] introduced the problem of associating names and faces from weakly labeled news images, while [2] proposed multimodal semi-supervised learning methods for bridging textual and visual signals. Later works such as [3] tackled the face naming problem in multimedia corpora, and [4] employed deep learning approaches for actor identification by aligning names and faces across video frames and scripts.

These works show that entity-level alignment of images and text—where the connection is based on identity rather than explicit description—is feasible with appropriate modeling strategies. Motivated by these insights, we extend this line of research to the Persian domain, where such multimodal entity linking has not been systematically explored.

Chapter 3

Proposed Methods (Detailed)

3.0.1 Data Collection

Sources & Crawling. We crawled biographies and related pages from public web sources and blogs. Content extraction used *BeautifulSoup*, *Scrapy*, and *Selenium*, followed by text cleaning and normalization; all records were stored in structured JSON.

Schema & Normalization. Each entity record contains {name, category, Persian summary, image path/URL}. Heterogeneous JSONs were merged; names were normalized; duplicates were removed; provenance and image-download status were tracked.

Human QA. Quality control: we used *Label Studio* with at least two independent annotators per item; results were exported in CSV/JSON for downstream analysis.

3.0.2 Data Model and Preprocessing

- **Entity schema.** Each record represents a person with fields: `name`, `category` $\in$ {`athlete`, `politician`, `poet`, `art_music`, `scientist`}, `summary` (Persian biographical paragraph), and `image_path`/`image_url`.
- **Merging heterogeneous sources.** We union category-specific JSONs, deduplicate by normalized name, and keep per-category provenance.
- **Persian normalization.** We harmonize characters, spacing, punctuation, diacritics, and URLs/emails/numbers; we avoid aggressive stop-word removal to preserve named entities.
- **Image attachment.** For each person, we keep one high-priority image based on filename heuristics and availability; records also track `is_image_downloaded`.

Notebook trace—Data layer

- `All_data_merger.ipynb`:
  - Loads category JSONs (`art_music_with_bios_imaged_crawled.json`, `athlete_...`, `poets_...`, `politicians_...`, `scientist_...`).
  - Adds a `category` field and per-record lineage; merges to a single corpus; reports per-category counts and total.
- `Multimodal_RAG_Data_Preperation.ipynb`:
  - Name normalization (`normalize_name`), priority rules for selecting image candidates (e.g., prefer canonical `_image*` files).
  - Merges narratives into a `summary` field; attempts image downloads; flags `is_image_downloaded`; includes a crawler step for misses.
- `text_normalization.ipynb`:
  - Installs **Hazm** and **Dadmatools**; applies a `DadmaNormalizer` configuration (char unification, punctuation spacing, URL/email/mobile masking if needed), plus Hazm normalization to produce clean Persian summaries.

3.0.3 Embedding & Indexing

- **Text encoder.** `intfloat/multilingual-e5-base` (dim=768). Text normalization → encode → L2-normalize.
- **Image encoder.** `open_clip` ViT-B/32 (dim=512). Preprocess with model transforms; L2-normalize.
- **Indexes.** Prefer **FAISS** (Inner Product on normalized vectors ≈ cosine); fallback to **hnswlib** (cosine/L2). We persist `.idx` (FAISS) or `.bin` (HNSW), and expose `knn()` for top-k retrieval.

Notebook trace—Embedding layer

- `Embedding_and_unimodal_pipeline.ipynb`:
  - Environment setup (CUDA, PyTorch, Sentence-Transformers, OpenCLIP, FAISS/hnswlib).
  - Downloads models (mE5; OpenCLIP ViT-B/32).
  - Builds text indexes; supports text-only search, query logging (`search_logs.jsonl`), and MCQ Top-k evaluation.
- `Multimodal_MCQ_RAG.ipynb`:
  - Loads unified JSON and image folders per category; creates **both** text and image indices.
  - Utility: `load_index_any`, `knn`, device handling, consistent dims (text=768, image=512).
  - Supports multimodal retrieval (text→image, image→text, and text→text for passages).

3.0.4 Multimodal Pipelines (Text-Image)

- **Retriever.** For each multimodal MCQ question, the retriever first processes the four candidate answer options. For each option, the top-3 text embeddings are retrieved together with their corresponding face images, resulting in a pool of 12 candidate images. These 12 images are compared against the input query image using cosine similarity, and the top-3 most similar candidates are selected as the final retrieval set. Thus, for each question we have three retrieved entities (with names, biographies, and images) as potential evidence.
- **Generator.** We experimented with three prompting approaches for integrating the retrieved evidence with the question:
  1. *Pairwise verification.* The system takes the top-1 retrieved entity and compares its image against the query image via the GPT-4o vision API, asking only: "Are these two images the same person?" with a strict Yes/No response. If the answer is `No`, the process repeats with the second and third candidates. If all three fail, retrieval is marked as a miss (hit=0). If a match is confirmed, the system fetches the corresponding biography and constructs a context: (i) the question, (ii) the four options, (iii) a statement that "This image belongs to [NAME]". GPT-4o is then prompted to provide evidence-based reasoning and select the correct option.
  2. *Three-way identity choice.* To overcome the weaknesses of the first approach, the query image and the three retrieved candidates are presented simultaneously to GPT-4o, asking: "Which of these three people does this image belong to?" plus an additional option *None of them*. If *None* is chosen, retrieval is scored as a miss. Otherwise, the identified person's biography, together with the question, four options, and the statement "This image belongs to [NAME]" are passed to GPT-4o, which then selects an answer with supporting evidence.
  3. *Face clustering (InsightFace).* We also tested a face recognition–based approach using `InsightFace` with clustering threshold set to 1.0 (validated empirically as optimal). The query image and the top-3 retrieved images are clustered; if one retrieved candidate falls in the same cluster as the query, that candidate is taken as the match. The system then proceeds as in the second step: attaching the identified biography and the identity statement to the question, and prompting GPT-4o to reason over the context and choose the correct option.
- **Logging.** For every question, we log (i) the original query terms, (ii) IDs and names of the retrieved candidates, (iii) similarity scores, (iv) the selected approach outcome, and (v) the final model prediction. Results are stored in structured CSV/JSONL formats for ablation analysis.

3.0.5 Unimodal Pipelines (Text-Only)

Normalization & Target Heuristics. All names and texts are normalized prior to retrieval. To determine the target entity for each MCQ, we use a two-stage heuristic: (i) if the gold answer string matches a catalog name, we select all matching indices (`answer_name`); (ii) oth-

erwise, we scan the question for any catalog name as a substring and pick the longest matches (`question_heuristic`). If neither fires, the example is marked with no detectable entity.

Text Index & Search. Questions are issued to a text-only index (`search(q, k)`), built over normalized biographies. For each query we keep the top-k documents with their scores and names.

Entity Hit@k & Answer Hit@k. For evaluation we compute two families of Hit@k: (i) *Entity Hit@k*: whether any of the target entity names appears among the top-k retrieved names; (ii) *Answer Hit@k*: whether the normalized gold answer string occurs in the concatenation of the retrieved {name | summary} fields among the top-k documents. Aggregation uses a standard any-prefix predicate over $k \in \{1,3,5,10\}$.

MCQ Retrieval Evaluation. For each MCQ item, we log (time stamp, question, gold answer, target-source tag, target indices/names, per-k hit booleans, retrieved rows with scores). A summary report with macro Hit@k, skipped counts (no detectable entity), and index paths is saved.

Retrieval Diagnostics. We further run a diagnosis routine that (a) separates cases into *entity hit*, *answer hit*, *both*, *neither*; (b) records the first-rank position when a target is hit; (c) summarizes top-1 score statistics for hits vs. misses.

Text→Text Recall@K (Name Queries). As a catalog sanity check, for each document with a non-empty name we encode the name as a query (``query: <name>'') and search the text index; Recall@K is computed by checking whether the document's own index appears in the top-k list ($k \in \{1,3,5,10\}$).

Text-Only MCQ Classification. Given an item {context, options}, we encode the context as a query (``query: <context>'') and the options as passages (``passage: <name>''). We choose the option with the highest dot-product similarity to the query. Accuracy is reported over the constructed MCQ set.

Rank-Weighted Option Scoring. For analysis, we also retrieve top-5 docs per question and compute a rank-weighted presence score for each option by checking whether the normalized option string appears in the joined {name | summary} of retrieved docs, with weights $1, \frac{1}{2}, \frac{1}{3}, \ldots$. We export per-question rows (gold/predicted answer, Hit@3 over options, top-3 option list and scores, top retrieved catalog name) to `text_mcq_top3_results.csv`.

3.0.6 Multimodal Fine-Tuning and Aligned Embeddings

Fine-Tuning. To directly bridge the gap between text and image modalities, we fine-tuned the CLIP model on our curated text–image pairs. The training objective is to minimize the distance between the embedding of a person's biography and the embedding of their corresponding face image. Training was conducted for 30 epochs over the entire dataset. After alignment, for query encoding we use: (i) image embeddings when only an image is provided, (ii) text embeddings when only text is provided, (iii) the mean of text and image embeddings when both are available.

Retriever. At inference time, a query image is passed through the fine-tuned CLIP encoder to obtain its embedding. The system then retrieves the top-3 nearest neighbors in the aligned embedding space, based on cosine similarity. Each retrieved candidate consists of both the image and its paired biography. These candidates serve as evidence passages for downstream answer generation.

Generator. For the generation step we use the `Gemma3` LLM. The top-3 retrieved text–image pairs (biographies + faces) are provided in the prompt, along with the question. The model is instructed to reason over the evidence and select the most plausible answer option. This allows the generator to leverage both modalities simultaneously.

Logging. For each question, we log (i) Query: The input question, (ii) Correct Answer: The answer that is considered correct, (iii)Generated Output: The final answer provided by the Gemma3 model, (iv)Top-3 Retrieved Answers from CLIP: The three most probable answers retrieved by the CLIP model, along with their similarity scores, (v)Retrieval Success Rate: A retrieval is considered successful if, among the top three answers, the correct person's name appears in at least one of the outputs.

3.0.7 Demo Web Application

Since our experiments and pipelines were implemented in `.ipynb` notebooks, we explored three options for creating a lightweight web-based demo:

1. **Voilà**: This option directly renders a Jupyter notebook into a web interface with some extra features. However, because users would still have access to the underlying code, we decided not to use this approach.
2. **Gradio**: A widely suggested library in the data science and AI engineering community. We selected this option because its implementation was simple, requiring only the extraction of the main functions and models from our notebooks. It allowed us to build an interactive interface quickly without major code modifications.

3. **Streamlit**: Although powerful, this framework is more suited for dashboards, monitoring, and evaluation reports. Streamlit required stronger integration with existing functions and was more time-consuming to adapt compared to Gradio. Additionally, execution was slower for fully interactive workflows where users expect immediate outputs. Thus, we did not pursue this option.

In conclusion, we adopted **Gradio** as the demo framework due to its simplicity, speed, and popularity, which enabled us to showcase our multimodal RAG system in an accessible and interactive way.

Chapter 4

Experimental Setup

4.0.1 Evaluation Tasks

To comprehensively evaluate our system, we designed four categories of questions, covering both unimodal and multimodal scenarios, and both multiple-choice and open-ended formats:

1. **Unimodal – Multiple Choice:** 90 questions, each with four candidate answers. The model is required to select the correct option based only on textual evidence.
2. **Unimodal – Open-Ended:** 110 questions without predefined options. The model must generate free-form answers in Persian, evaluated using exact match, token-level F1, and ROUGE-L metrics against gold references.
3. **Multimodal – Multiple Choice:** 50 questions where each input consists of both a portrait image and four candidate answers. The system must leverage both the visual and textual retrieval pipelines to choose the correct option.
4. **Multimodal – Open-Ended:** 50 questions with portrait images as input but without candidate options. The model must generate free-form answers, which are evaluated similarly to the unimodal open-ended case.

4.0.2 Challenges

LLM-as-a-Judge Evaluation: Because the gold answers and the model's generated answers are often phrased differently while expressing the same meaning, simple string-matching metrics (e.g., exact match or token-level F1) are not reliable. To address this, we again used large language model as an automatic judge:

- **Process –** For each question we provided the gold answer and the model's generated answer to an external LLM with a short instruction: "Determine if these two answers are semantically equivalent. Reply only with 1 (yes) or 0 (no)."
- **Output –** The judge returns a binary label indicating semantic equivalence.
- **Purpose -** This gives a more realistic accuracy metric, capturing meaning rather than surface form, and allows calculation of overall accuracy and per-category accuracy even when wording differs.

This "LLM-as-a-judge" step ensures that reported generative accuracy reflects true correctness rather than mere lexical overlap. Using this protocol, overall accuracy rises to 77.3%, up from 20% without LLM-based judging.

4.0.3 Compared Models

We evaluated these four categories of questions on our proposed **Persian Multimodal RAG system**, as well as on several baseline models *without retrieval augmentation*. This comparison allows us to measure the added value of retrieval and multimodal grounding in contrast to pure generation from large language models.

4.0.4 Baseline Models

We considered a set of multilingual, (often) multimodal LLMs as baselines to situate our Persian RAG against general-purpose generators:

- **GPT-4o** (multimodal, text+image)
- **GPT-mini** (lightweight variant)
- **Claude Sonnet 4** (Anthropic)
- **Gemini 2.5 Pro** (Google DeepMind)
- **Llama 3.1 / 3.2 Vision** (Meta, open-source; 3.2 supports vision)
- **Qwen 2** (Alibaba)
- **Haystack MT5, MBART** (open-source multilingual generators in the Haystack framework)
- **LangChain** (framework for LLM tooling and RAG pipelines)

Practical Constraints. The notebooks for testing our test sets with all these models are all available in our Github repository, however there are some limitations for using these models, as these models are not all free, and we needed a monetary plan to use them, we could only use the GPT-4o and GPT-mini models as our baselines for this project. Also as we needed to query our four test sets to these models, there can be a rate limit problem as well, depending on what platform one is using for evaluation.

4.0.5 Input Types

We evaluate models under four prompt configurations that mirror our main tasks:

1. **Unimodal / MCQ (Text-only):** question + four options; no image.
2. **Unimodal / Open-Ended (Text-only):** question; no options; no image.
3. **Multimodal / MCQ (Text+Image):** question + four options + portrait image.
4. **Multimodal / Open-Ended (Text+Image):** question + portrait image; no options.

4.0.6 Prompting and Inference

For MCQ settings, the model is instructed to (i) read the question (and image if present), (ii) select exactly *one* option, and (iii) optionally provide a short justification (suppressed during batch scoring). For open-ended settings, the model returns a short Persian answer. When an image is present, multimodal models consume the portrait directly.

4.0.7 Evaluation Metrics

We report:

- **Overall Accuracy** (MCQ): fraction of correctly chosen options.
- **Accuracy by Category:** per-domain accuracy (athletes, politicians, poets, artists/musicians, scientists).
- **Micro/Macro Accuracy:** micro aggregates over all items; macro averages per-category to reduce class-size bias.

For open-ended questions, we additionally use an LLM-as-a-judge protocol (binary semantic match) to handle paraphrasing, and we record exact-match where applicable.

4.0.8 Reproducibility Notes

All evaluation notebooks for the four prompt types are organized in our repository. When applicable, API keys and rate limits are documented; batch sizes and retry policies are set to avoid throttling. We log model name/version, prompt template ID, and timestamps for each call.

4.0.9 Baseline Summary Table

Table 4.1 summarizes the considered baselines and the ones executed in our final runs.

Table 4.1: Baseline models considered and execution status.

Model	Modality	Executed in this study
GPT-4o	Text + Image	✓
GPT-mini	Text (and limited vision variants)	✓
Claude Sonnet 4	Text (+Vision variants)	–
Gemini 2.5 Pro	Text + Image	–
Llama 3.1 / 3.2 Vision	Text / Text + Image	–
Qwen 2	Text (+Vision variants)	–
Haystack MT5	Text	–
Haystack MBART	Text	–
LangChain (framework)	N/A (orchestrator)	N/A

Chapter 5

Discussion & Results

5.1 Performance measures

- `Unimodal RAG`
  - **Overall Accuracy:** Shows the percentage of questions for which the generated answer exactly matched the gold answer after normalization. It represents the model's end-to-end performance across the entire dataset..
  - **Accuracy by Category:** Displays accuracy for each question category (e.g., politician, athletes). This highlights which topical domains the model handles well and where it struggles.
  - **Accuracy by Gold Answer Length Bucket:** Buckets the ground-truth answers by their character length and plots accuracy. It helps assess if longer, more descriptive reference answers are harder for the model to reproduce exactly.

- `Multimodal MCQ`
  - **Hit@1 / Hit@3:** Shows the percentage of questions for which the generated answer exactly matched the gold answer after normalization. It represents the model's end-to-end performance across the entire dataset..
  - **MRR:** ranking quality, higher means the correct document appears earlier. Athlete and poet categories achieve near-perfect retrieval, while art_music and politicians show weaker recall.

- `Multimodal NoOptions`

Table 5.1: Multi-Modal Open-Ended Model Overall Performance

Metric	Value
Total Questions	50
Overall Answer Accuracy (%)	30.00
Overall Retrieval Accuracy (%)	26.00
Precision	0.093
Recall	0.260
Hit@1	0.140
Hit@3	0.260
Mean Top Retrieval Score	0.6632
Mean Second Retrieval Score	0.6367
Mean Third Retrieval Score	0.6221

- `Baselines (Non-RAG): GPT-4o, GPT-4o-mini` We benchmarked four non-retrieval baselines across our test suites:

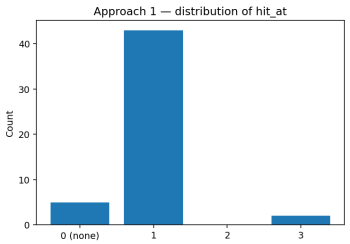
- **MCQ — Text-only (no image): 56%**
- **MCQ — Multimodal (text+image): 66%**
- **Open-ended — Multimodal (text+image, no options): 12%**
- **Open-ended — Text-only (no image, no options): 44.5%**

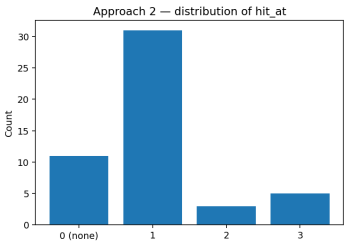
Across all test sets, these *non-RAG* baselines underperformed our proposed RAG system.

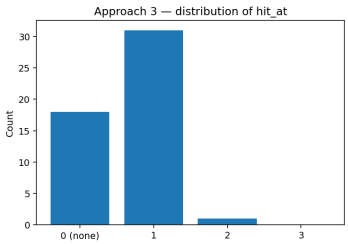
Table 5.2: Non-RAG baseline accuracies (GPT-4o).

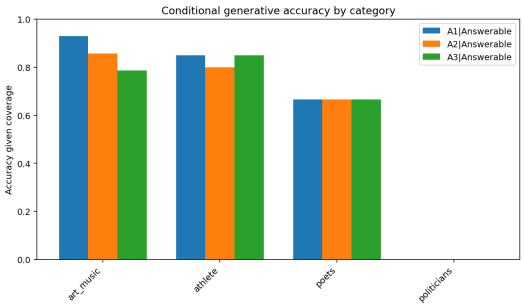
Task	Accuracy (%)
MCQ (Text-only)	56.0
MCQ (Multimodal)	66.0
Open-ended (Multimodal)	12.0
Open-ended (Text-only)	44.5

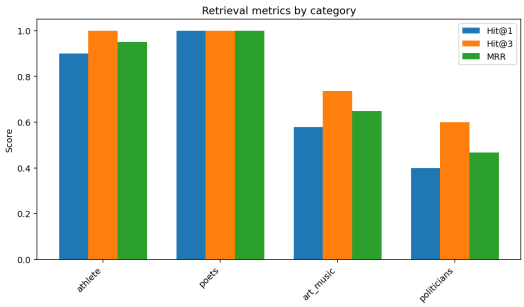
5.1.1 Multimodal Results (Image Evidence)

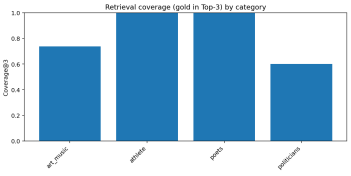


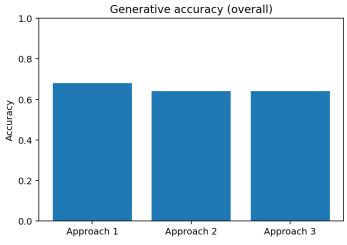


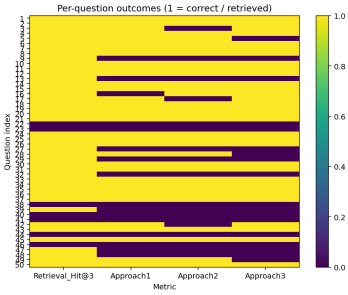


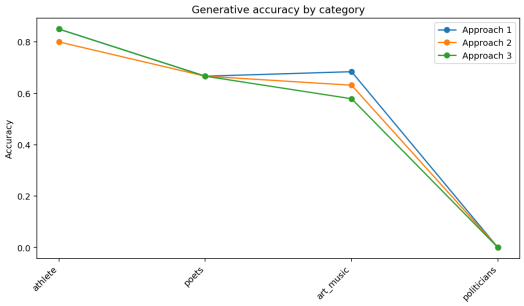


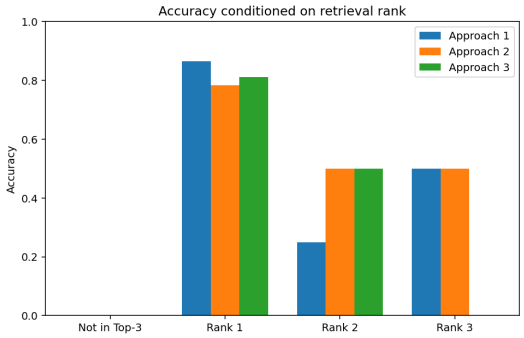




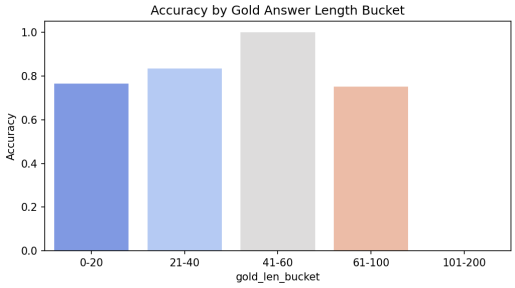


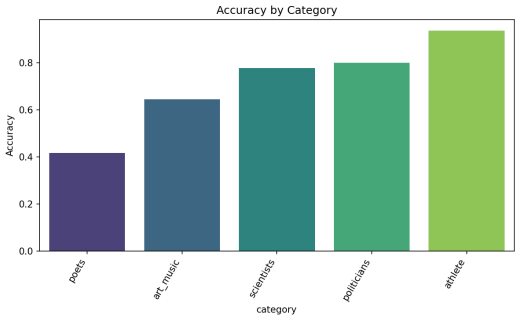


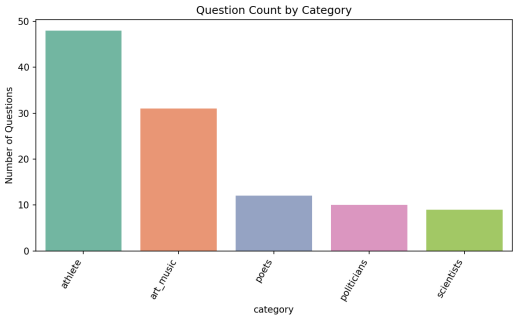


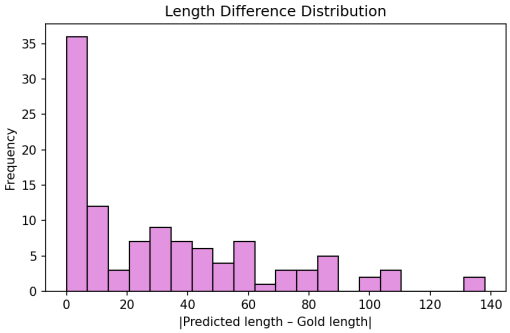


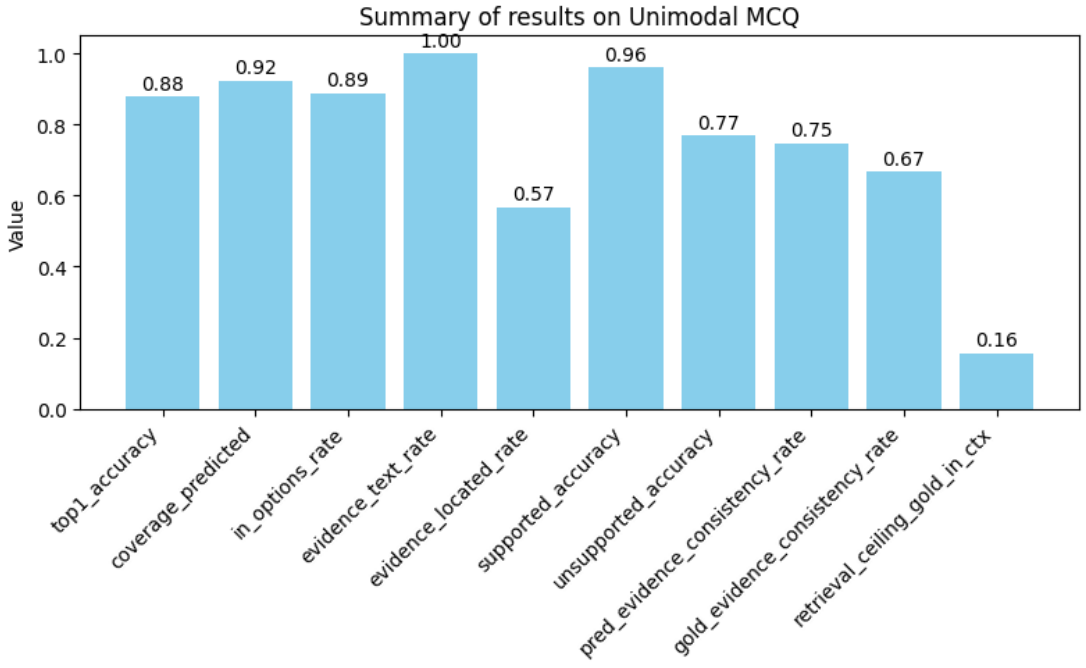
5.1.2 Unimodal Results (Image Evidence)







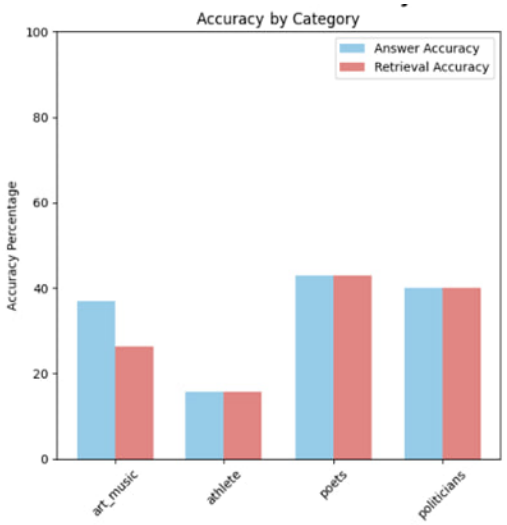


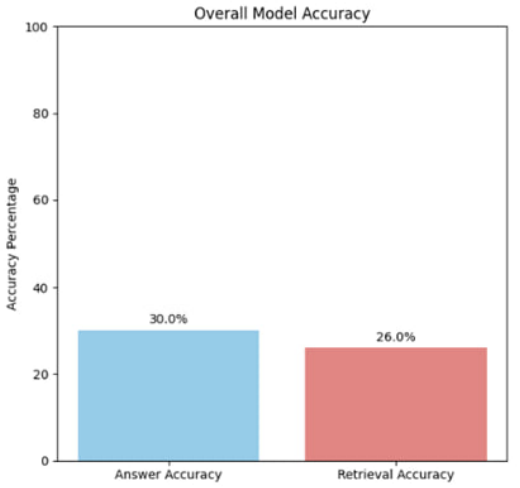


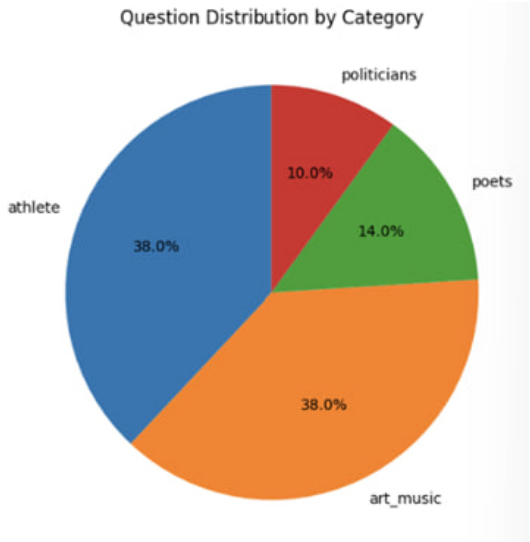
top1-accuracy – Proportion of questions where the highest-ranked generated answer matches the ground-truth answer. coverage-predicted – Share of questions for which the RAG system produced any valid prediction (i.e., didn't abstain). in-options-rate – Percent of predictions that fall inside the provided multiple-choice options or expected answer set. evidence-text-rate – Fraction of responses where the model included explicit supporting text passages. evidence-located-rate – How often the retrieved documents actually contain the gold (true) evidence span. supported-accuracy - Accuracy on questions where correct supporting evidence was retrieved and

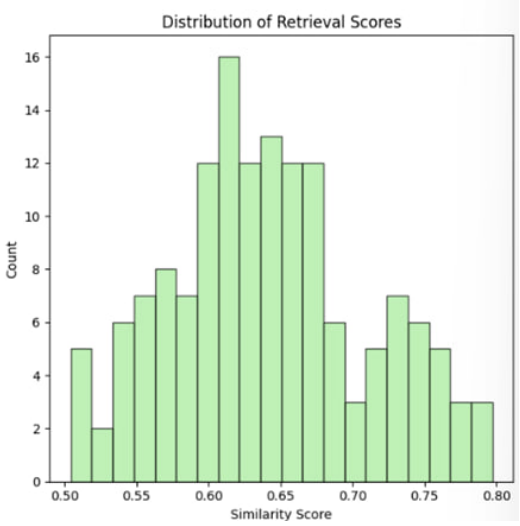
used. unsupported-accuracy – Accuracy on questions where the retrieved context did not contain correct evidence (shows generator's guesswork). pred-evidence-consistency-rate – Percent of predictions whose cited evidence logically supports the model's own answer. gold-evidence-consistency-rate – Percent of predictions consistent with the human-annotated (gold) evidence passages.

5.1.3 Multimodal Open-ended Results (Image Evidence)

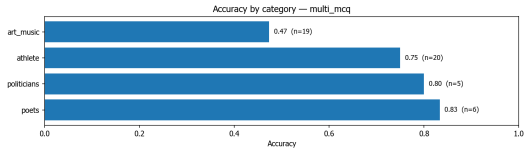


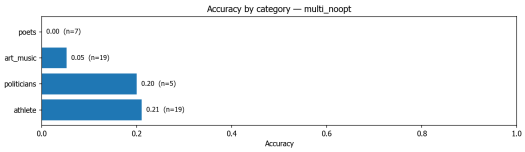


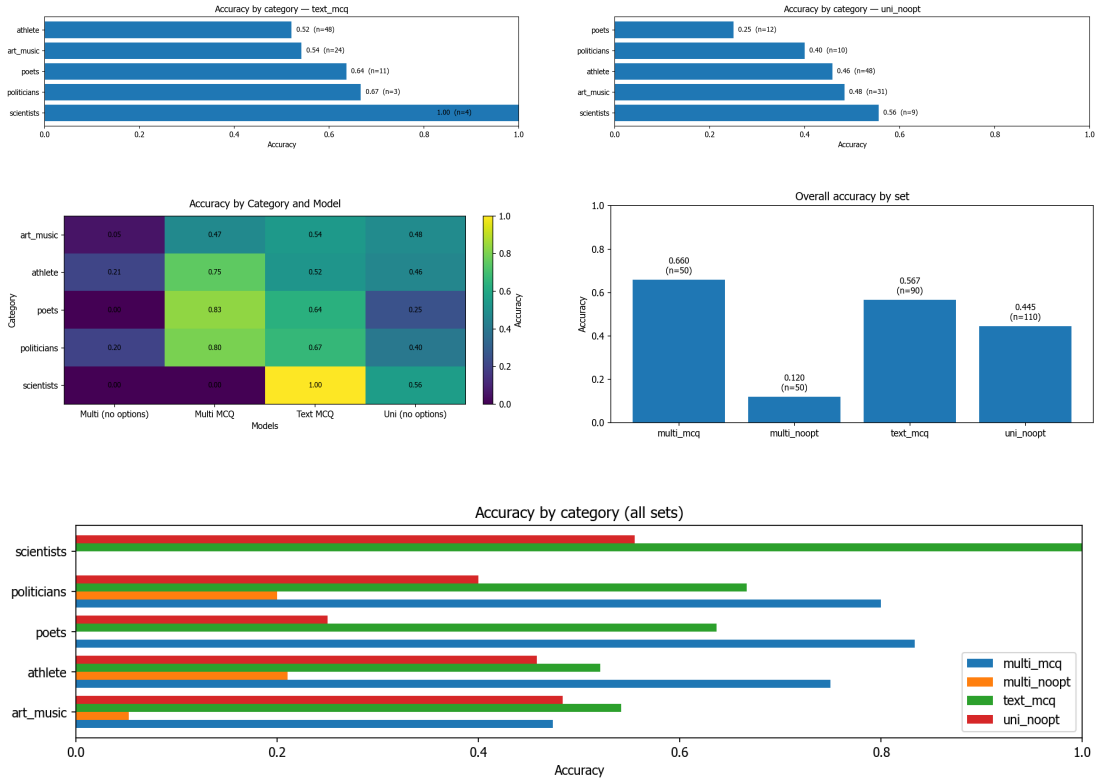




5.1.4 Baselines (Non-RAG)







These results indicate that retrieval remains the primary bottleneck in the fully open-ended multimodal setting: once the correct entity is surfaced in the top-3, the generator benefits from grounding and achieves materially higher answer accuracy than chance; however, misses at the retrieval stage propagate downstream. The relatively narrow gap between the mean similarity of the first and third items suggests high confusability among visually similar entities (e.g., same domain and era).

5.2 Ablations and Error Analysis

Effect of Identity Verification. Comparing the three decision procedures, pairwise verification is sensitive to an erroneous top-1; introducing a three-way identity choice with *none* reduces such failures by jointly considering the top-3. Face clustering (InsightFace, τ=1.0) adds an independent signal that often resolves edge cases where textual retrieval is strong but the top-1 visual neighbor is a near-duplicate of another person. (Qualitative analyses and failure case prints in the logs support these observations.)

Category Effects. Athletes and poets show higher rank quality; art/music and politicians exhibit weaker recall—likely due to broader lexical overlap and fewer distinctive visual cues per entity, as reflected in the per-category diagnostics.

Common Failure Modes. (i) **Entity not detectable in question**: name not present or normalized forms diverge; (ii) **Biography–portrait mismatch**: dataset-level noise where a photo does not belong to the biography; (iii) **Near-duplicate confusions**: multiple visually similar candidates in the same domain; (iv) **Long-form gold answers**: open-ended references that are semantically correct but phrased differently (handled by LLM-as-a-judge). These informed data cleaning and the adoption of the face-recognition sanity check in the pipeline.

Chapter 6

Summary. We introduced a Persian multimodal RAG system over biographies of Iranian public figures, integrating dual encoders (mE5 for text; CLIP for images), efficient indexing, and three complementary multimodal decision procedures (pairwise verification, three-way identity choice, and face clustering). Our evaluation suite spans MCQ and open-ended tasks in both unimodal and multimodal regimes, with an LLM-as-a-judge protocol for robust semantic scoring. Across $\sim$300 questions, top-k retrieval and cross-modal grounding consistently improve answer selection relative to text-only baselines; however, retrieval quality remains the dominant bottleneck in open-ended image-backed queries.

Contributions. (1) A cleaned, normalized Persian corpus with images; (2) a reproducible retrieval stack with FAISS/HNSW and logging for ablations; (3) multimodal MCQ/open-ended evaluators; (4) practical identity-verification strategies (LLM vision prompts and `InsightFace` clustering) that mitigate biography–portrait mismatch.

Limitations. First, text–image pairs are linked by entity identity rather than descriptive captions, which challenges canonical contrastive fine-tuning. Second, the catalog contains residual noise (name variants, portrait mismatches) that degrades recall. Third, visually and lexically confusable sub-domains (e.g., art/music, politicians) depress rank quality. Finally, open-ended scoring is constrained by the quality of the LLM judge and prompt design.

Future Work. We plan to: (i) expand and de-noise the catalog with automated face-verification during ingestion; (ii) incorporate stronger cross-encoders or late-interaction rerankers over the retrieved top-k; (iii) explore instruction-tuned VLMs for more reliable identity decisions; (iv) add Persian-aware, entity-centric augmentation (e.g., aliases, transliteration variants) to boost recall; and (v) benchmark alternative evaluation judges and task-specific rubrics for open-ended Persian answers.

Takeaway. When modalities are connected only via entity identity (rather than descriptive supervision), robust retrieval design—plus lightweight identity verification—is pivotal. With careful normalization, indexing, and verification, multimodal RAG in Persian can be made both *practical* and *reproducible*, while highlighting where better data and stronger reranking can yield the largest gains.

References

[1] Thomas L. Berg, Alexander C. Berg, Jaety Edwards, and Michael Maire. Names and faces in the news. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 848–854, 2004.

[2] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[3] Yang Hu, Dong Li, Zhouchen Lin, and Baining Zeng. Face naming in multimedia data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(2):437–446, 2012.

[4] Duc-Tien Dang-Nguyen Le, Bogdan Ionescu, and Cathal Gurrin. Deep face-name matching for actor identification in tv series. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 4153–4160, 2017.