A Comparative Analysis of Retrieval Models for Persian Question Answering

Amir Malekhosseini , Amin Fadaee
Sharif University of Technology

August 9, 2025

Abstract

This report presents the results of a comprehensive project in Persian-language Information Retrieval. The project's goal was to build, train, and evaluate a question-answering system based on a custom-generated corpus of texts about Iranian provinces. Three different models were compared: (1) a statistical baseline based on TF-IDF, (2) a large multilingual language model (`cis-lmu/glot500-base`) in a zero-shot setting, and (3) the same language model fine-tuned on our custom question-answering dataset. The final evaluation was conducted using human judgment via the Label Studio tool. The results indicate that the statistical TF-IDF model and the fine-tuned model both performed exceptionally well and were highly competitive, with each significantly outperforming the zero-shot baseline. This finding clearly demonstrates the importance of domain-specific training for language models in specialized applications.
The project is loacted in `https://drive.google.com/drive/folders/1GjRoy9Big1WV_IrVreDwZQm3geO8mAbx`.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated remarkable advancements in the field of Natural Language Processing (NLP). However, the performance of these models in specialized applications and on domain-specific datasets remains an active area of research. This project, conducted as part of the Natural Language Processing course at Sharif University of Technology, investigates and compares the performance of classical and modern methods for the task of information retrieval in the Persian language. The primary objective is to answer whether a large, domain-finetuned language model can outperform a strong statistical method like TF-IDF in a specialized domain. In this report, we first describe the project methodology, including the dataset construction, the models used, and the evaluation framework. We then present and analyze the quantitative and qualitative results from our human evaluation, concluding with a summary of the strengths and weaknesses of each approach.

2 Methodology

The project workflow consisted of several key stages, from data preparation to the final analysis of the results.

2.1 Dataset and Corpus

The project's corpus comprises textual summaries of the geography, history, and tourist attractions of various provinces in Iran. These texts were generated from structured JSON files using the Gemini language model. Based on this corpus, two question-answering datasets were created:

- **Training Set:** Contained several questions for each text passage, used for fine-tuning the language model.
- **Evaluation Set:** Consisted of 50 independent questions used exclusively for the final evaluation of the models. This set was not used in the training process.

2.2 Models Under Evaluation

Three different models were selected for a comparative performance analysis in the information retrieval task.

1. **TF-IDF:** As a statistical baseline, the `TfidfVectorizer` from the `scikit-learn` library was used to convert texts and queries into numerical vectors. The similarity between a query and the corpus documents was calculated using Cosine Similarity.
2. **Zero-shot GLOT500:** The `cis-lmu/glot500-base` model from the `sentence-transformers` library was used directly without any training on our data. This model leverages its general pre-trained knowledge to find the most relevant document for each query.
3. **Fine-tuned GLOT500:** The same `glot500-base` model was fine-tuned on our custom training set using the `MultipleNegativesRankingLoss` function. This approach, a form of Contrastive Learning, helps the model generate better semantic representations for question-answer pairs within our specific domain.

2.3 Human Evaluation Framework

For the final evaluation, the top three outputs from each model for all 50 evaluation questions were collected. These retrieved passages were then shuffled randomly and presented to two human annotators via the **Label Studio** software, without revealing the source model of each passage. Each annotator rated the relevance of every passage to the corresponding question on a scale from 1 to 9. This blind evaluation method ensures the removal of any annotator bias towards a particular model. The final analysis was based on the average of the scores provided by the two annotators.

3 Results and Analysis

After collecting and processing the human evaluation data, the following quantitative and qualitative analyses were performed.

3.1 Overall Model Performance

A key metric for performance is the number of answers that received a perfect average score of '9' from the annotators. Table 1 shows the distribution of these perfect scores based on the rank of the answer as proposed by each model.

Table 1: Distribution of perfect scores (average rating of 9) by the output rank of each model.

Model	**Rank 1**	**Rank 2**	**Rank 3**	**Total '9' Scores**
`tfidf_top3`	44	3	1	48
`finetuned_glot500_top3`	43	3	1	47
`zeroshot_glot500_top3`	13	0	1	14

Based on the results in Table 1, the following conclusions can be drawn:

- The **TF-IDF** model delivered a surprisingly strong performance, achieving the highest number of perfect answers (48). In over 90% of its correct retrievals, it presented the best answer as its top choice (Rank 1), indicating high confidence.

- The **Fine-tuned GLOT500** model was a very close competitor with 47 perfect answers. This result validates the effectiveness of the fine-tuning process, as it successfully transformed a weak base model into a powerful, specialized retriever.

- The **Zero-shot GLOT500** model performed significantly worse, with only 14 perfect answers. The stark difference between the base model and its fine-tuned version clearly demonstrates the necessity of domain-specific training for specialized tasks.

3.2 Performance by Question Category

The 50 evaluation questions were grouped into six thematic categories to identify the strengths and weaknesses of each model in different areas. Table 2 displays the success rate of each model per category.

Table 2: Model success rates across different question categories.

Question Category	**TF-IDF**		**Fine-tuned GLOT500**		**Zero-shot GLOT500**	
	Success	Success Rate	Success	Success Rate	Success	Success Rate
Provincial Capitals	7/8	87.5%	8/8	100.0%	1/8	12.5%
Natural Features & Attractions	13/13	100.0%	11/13	84.6%	4/13	30.8%
Historical & Ancient Sites	12/12	100.0%	12/12	100.0%	5/12	41.7%
Personalities & Historical Periods	3/3	100.0%	3/3	100.0%	0/3	0.0%
Cultural, Industrial, & Modern Sites	6/6	100.0%	5/6	83.3%	1/6	16.7%
Numbers, Figures, & Classifications	7/8	87.5%	8/8	100.0%	3/8	37.5%

The analysis of Table 2 reveals several interesting points:

- **TF-IDF** achieved a perfect success rate (100%) in most categories, underscoring its power for fact-based questions where keyword matching is highly effective. Its only minor weaknesses were on questions about provincial capitals and specific figures.

- The **Fine-tuned Model** managed to achieve perfect accuracy in categories requiring more precise understanding (like capitals and figures), covering the minor gaps left by TF-IDF. However, it was slightly weaker than TF-IDF on categories where keywords were very prominent, such as "Natural Features" and "Industrial Sites".

- The **Zero-shot Model** was weak across all categories but had its best relative performance on "Historical & Ancient Sites" (41.7%). This suggests the model's general pre-trained knowledge contains more information about famous landmarks than about other specialized topics.

4 Conclusion

This project provided a comprehensive comparison of three distinct approaches to information retrieval for the Persian language. The results highlight several key takeaways.

First, classical statistical methods like **TF-IDF** remain powerful and highly effective baselines. For extractive question-answering tasks where exact keyword matching is crucial, these models can produce excellent results with very low computational cost.

Second, the most significant finding of this study is the demonstrated power of **Fine-tuning**. The `glot500` model was ineffective in its base form but became a top-tier competitor after being specialized on our domain-specific data. This confirms that language models possess immense potential for deep learning and specialization in niche areas.

Finally, the choice between the top two models depends on the specific priorities of the application:

- For maximum **accuracy and simplicity of implementation** on this dataset, TF-IDF is the winner.
- For **semantic understanding and the potential to handle more complex queries** in the future, the Fine-tuned model is a better long-term investment, as it has proven its ability to learn deeply and compete at the highest level.

A Project Execution Guide

This section details the steps required to fully reproduce the project's results.

A.1 Prerequisites

The project was run in a Google Colab environment with a GPU enabled. The file structure in Google Drive must also match the layout described in Section 3.2. The following Python libraries are required:

```
sentence-transformers
transformers==4.30.0
torch
huggingface_hub==0.16.4
torchvision
protobuf==3.20.3
pandas
matplotlib
seaborn
```

A.2 Execution Steps

1. **Model Training:** Open the `NLP_HW2.ipynb` notebook. Run the cells for installing dependencies, mounting Google Drive, and logging into Hugging Face. Then, execute the main model training cell. This will save the fine-tuned model to the `Models/` directory.
2. **Generate Retrieval Results:** In the same notebook, run the cell under "Part 4: Retrieval Evaluation". This script will execute all three models against the 50 evaluation questions and save the consolidated output to `Retrieval_Results.json`.
3. **Prepare for Human Annotation:** Run the final cell of `NLP_HW2.ipynb` to generate the `label_studio_ready.json` file, which is formatted for import into Label Studio.
4. **Human Annotation:** Import the generated JSON file into a Label Studio project. Have at least two annotators complete the evaluation. Export the results as two separate JSON files: `Label_Studio_Output_1.json` and `Label_Studio_Output_2.json`.
5. **Final Analysis:** Open the `Analytical Report.ipynb` notebook. Execute all cells in order. This notebook will process the Label Studio outputs and generate the final tables and analyses presented in this report.