NUR at IslamicEval 2025 Shared Task: Retrieval-Augmented LLMs for Qur'an and Hadith QA

Serag Amin[1]* **Ranwa Aly**[1]* **Yara Allam**[1]*
Yomna Eid[2] **Ensaf H. Mohamed**[2]
[1] Faculty of Computers and Data Science, Alexandria University
[2] Center for Informatics Science (CIS), School of Information Technology and Computer Science, Nile University
cds.{seragamin23144,ranwakhaled30408,yaraibrahim23394}@alexu.edu.eg
{YEid,EnMohamed}@nu.edu.eg

Abstract

In this paper, we present our contribution to the IslamicEval 2025 shared task. More specifically, we address subtask 2, which is a passage retrieval (PR) system for Qur'an and Hadith, the two central bodies of text in Islam. Basing off of a fine-tuned BERT-based sentence transformer retrieval model, we explore several approaches, including pipelined fine-tuning of cross-encoders, as well as using a state-of-the-art LLM for filtering and reranking of relevant passages, and identification of zero-answer questions. Our best-performing system achieves a MAP@10 of 0.1235, MAP_Q@5 of 0.1556, and MAP_H@5 of 0.2642 on the test set.

1 Introduction

As the two primary sources for Islamic teachings, the Holy Qur'an and the Hadith are essential to the lives of roughly 2 billion Muslims. They contain rulings, moral and spiritual guidance, and general ways of life, making Islamic question answering (QA) systems extremely important for those practising, and even inquisitive non-Muslims. It is also important that such systems maintain high accuracy and reliability, as small errors or hallucinations may have significant implications due to the sensitivity of the materials.

While QA in Arabic has been tackled previously (Koto et al., 2024) and remains an active research area, the challenge with Arabic morphological richness is amplified even more when it comes to religious texts, where context, syntax, or vocabulary can change a passage's meaning entirely. Previously, the Qur'an QA 2022 (Malhas et al., 2022) and Qur'an QA 2023 (Malhas et al., 2023) shared tasks addressed this challenge, but only within the scope of the Holy Qur'an. In comparison, Hadith collections present a broader, more complex challenge for information retrieval (IR). Hadith is built upon a chain of narrators quoting the Prophet Muhammad, peace be upon him, varying in length, phrasing, and authenticity, and spread across multiple compilations. They also lack a unified indexing system, as opposed to the Qur'an, which constitutes a singular source of information. This leads to a more dynamic and realistic approach in the IslamicEval shared task (Mubarak et al., 2025). For a free-text question in Modern Standard Arabic (MSA), the system must retrieve a ranked list of up to 20 Qur'anic passages or Hadiths that may contain the answer to the question. The question could also be unanswerable. In some cases, the question may also have no relevant answer in the Qur'an but one in the Hadith, or vice versa, requiring systems to be versatile in searching across both corpora.

Similar to the previous editions, the task provides us with a set of thematic Qur'an passages, as well as the Sahih Al-Bukhari Hadith collection. We are also provided with the *AyaTEC* Qur'an QA dataset (Malhas and Elsayed, 2020), as discussed further in Section 2. However, no equivalent exists for Hadith QA, prompting us to search for relevant external sources for training our systems.

Our contribution to the subtask involves pipelined fine-tuning of BERT-based sentence transformer models for the retrieval of relevant documents, followed by either a fine-tuned cross-encoder or a state-of-the-art LLM for filtering and identification of zero-answer questions. The system is then evaluated on mean average precision, specifically, MAP@10 and MAP@5 for the Qur'an and Hadith passages independently. The paper is structured as follows: Section 2 describes the data used for our experiments, Section 3 goes into the details of the experiments and provides an overview of the results achieved, and Section 5 discussing and drawing insights from these results. Lastly, Section 6 offers a conclusion to our work. We release our code and data publicly on GitHub[1].

*These authors contributed equally to this work.

[1] https://github.com/Yoriis/IslamicEval2025

Split	Train	Dev	Test
# Question–passage pairs	1261	298	–
# Questions			
Multi-answer	131 (62%)	26 (65%)	–
Single-answer	48 (23%)	8 (20%)	–
Zero-answer	31 (15%)	6 (15%)	–
Total	**210**	**40**	**71**

Table 1: AyaTEC v1.3 Split Distribution

2 Data

The task data consisted of three parts: the Thematic Qur'anic Passage collection (QPC) (Swar, 2007), containing *1,266* thematic passages that cover the whole Holy Qur'an in a simple-clean text style, without diacritics, the Sahih Al-Bukhari Hadith collection (Al-Sharjy and Al-Zubaidi, 2009), comprising *2,254* Hadiths, the authors having excluded redundant Hadiths and Arabic commentary, and the *AyaTEC v1.3* (Malhas and Elsayed, 2020) dataset, composed of question-passage pairs. A brief description of the split can be found in Table 1.

As the dataset size remains limited, we adopt a sequential fine-tuning strategy, adding increasingly task-specific datasets to enhance the model's adaptation to our domain. We use the Arabic portion of the TyDi dataset (Clark et al., 2020), containing about 15 thousand QA pairs. We use the Jalalayn Tafseer of the Qur'an, aggregated to the thematic passages provided. Additionally, to address the limited size of task-specific data, especially for Hadith, we use the QuQA and HaQA datasets (Alnefaie et al., 2023), which contain *3382* and *1598* QA pairs, respectively. Lastly, to increase models' sensitivity to zero-answer questions, we augment each of our datasets with several random negative samples - 5 negatives per sample for HaQA, and 3 negatives per sample for the others.

3 System

Our system has 2 stages: retrieval & re-ranking, discussed in this section & illustrated in figure 1.

3.1 Retrieval

To retrieve the top-K passages for a question, we encode the question and all thematic Qur'anic passages and Hadiths using a sentence embedding model, compute cosine similarity, and rank the passages. We evaluated several Arabic embedding models on the shared task's Qur'an-only dev set using **Recall@30** to establish baseline performance; results are in Table 10. The best model, `AraModernBert`[2], achieved a Recall@30 of **0.445**.

Retriever Fine-Tuning Starting from the AraModernBERT model, we fine-tuned for the target domains using the shared task's Qur'an-only training set and additional data (Section 2). Following prior work on dense retrieval with hard negatives (Zhan et al., 2021; ElKomy and Sarhan, 2023), we retrieved top-ranked Qur'anic and Hadith passages per question using the base model for fine-tuning. For each query, we sampled K passages in total, including multiple positives and treating the rest as hard negatives. We also tested positive-only fine-tuning to assess the impact of excluding negatives.

We implemented two fine-tuning pipelines, each using both cosine and contrastive loss:

- **Pipeline A**: Single-stage fine-tuning on the shared task's Qur'an-only training data.
- **Pipeline B**: Multi-stage curriculum fine-tuning using additional QA datasets (Section 2), starting with TyDiQA, followed by Tafseer, QuQA, HaQA, and finally the Qur'an-only set.

For both pipelines, we varied the number of *passages* (K) used during fine-tuning. Each K includes multiple positive and hard negative passages, retrieved from both Qur'an and Hadith corpora. We evaluated performance using recall at multiple retrieval depths, excluding unanswerable questions.

Pipeline A Direct fine-tuning (Table 2) shows strong gains over the positive-only baseline, with its best Recall@30 at **0.491** exceeding the baseline by over 20 percent. At larger retrieval depths, Recall@70 peaks at **0.592**.

Passages	Loss	Search	R@30	R@50	R@70
Positive only	Contrastive	Cosine	0.285	0.329	0.385
50	Contrastive	Cosine	0.462	**0.537**	0.555
70	Cosine	Cosine	0.446	**0.537**	**0.592**
	Contrastive	L2	**0.491**	0.521	0.552

Table 2: Top-performing configurations in Pipeline A by number of passages. Full results in Table 11.

Pipeline B The multi-stage curriculum (Table 3) surpasses Pipeline A at both shallow and deep retrieval. Its best Recall@30 reaches **0.541**, around 5 percent higher than Pipeline A, while its Recall@70 climbs to **0.688**, nearly 10 percent above Pipeline A's top result.

[2] https://huggingface.co/NAMAA-Space/AraModernBert-Base-STS

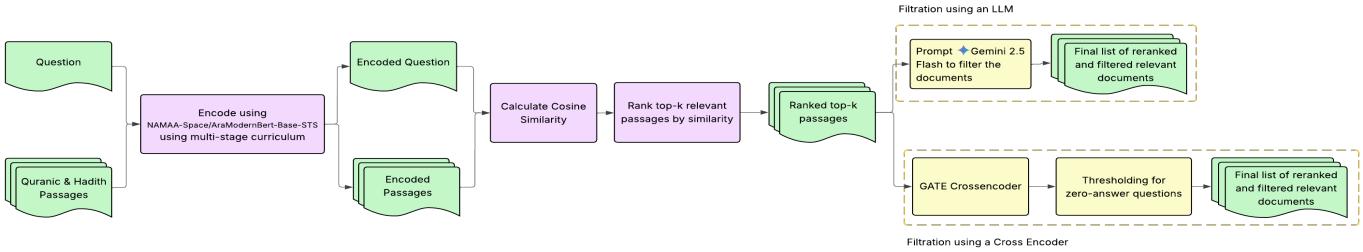

Figure 1: Figure showing the final pipeline used for the submitted runs. **Green** shapes represent input and output data modules, **purple** boxes denote retrieval processes, and **yellow** boxes signify reranking and filtering stages.

Passages	Loss	Search	R@30	R@50	R@70
60	Cosine	L2	0.505	0.600	**0.688**
70	Contrastive	Cosine	**0.541**	0.581	0.640
80	Contrastive	L2	0.537	**0.620**	0.645

Table 3: Top-performing configurations in Pipeline B by number of passages. Full results in Table 12

3.2 Reranking

To re-rank the retrieved documents, we experimented with two approaches: a cross-encoder architecture and a large language model.

3.2.1 Cross-Encoder Architecture

Building on the fine-tuned retrieval model, we use **Pipeline B** to fine-tune two cross-encoders: `AraBERTv0.2-base` (Antoun et al., 2020), and `NAMAA Space GATE Reranker V1` (GATE) (NAMAA-Space, 2025). Our choice of models is guided by the Arabic RAG leaderboard (Mohaned A. Rashad, 2025), which evaluates retrieval and reranking systems. GATE, built on `AraBERT` and `Arabic Triplet Matryoshka` (Nacar et al., 2025), ranks highly on this benchmark while also remaining resource-efficient. `AraBERTv0.2-base`, as one of the earliest widely adopted Arabic Transformers and GATE's predecessor, serves as a baseline for comparison. For identification of zero-answer questions, we use a thresholding-based approach. If all passages, after reranking, have scores below the threshold, the question is deemed to have no answers, and the systems returns -1.

Two versions of **Pipeline B** were experimented with. In one configuration, we drop the Tafseer dataset for fine-tuning, and exclude the task data as well **(Pipeline B1)**. This generally led to better results, as seen in Table 4. In the other, we utilize the full pipeline, ending with fine-tuning independently on two versions of the task data: one with only positive passages sampling, and one with Top-70 **(Pipeline B2)**. A representation of both pipelines can be found in Figure 2. Table 5 shows the **MAP@5** and **MAP@10** for the dev set after each fine-tuning step. Interestingly, in both scenarios, fine-tuning on the task data decreases performance.

It's also important to note that a k-value of *70* was used to retrieve the relevant passages, which were then reranked, and the scoring threshold for zero-answer questions was set at *0.15* for these experiments. We experimented with the thresholding hyperparameter, as can be seen in Appendix C.

Model	Metric	Baseline	TYDI	QUQA	HAQA
GATE	MAP@5	**0.3172**	0.2319	0.2372	0.2548
	MAP@10	**0.3215**	0.2503	0.2574	0.2786
AraBERT	MAP@5	0.0278	0.1712	0.1965	**0.2186**
	MAP@10	0.0371	0.1972	0.2138	**0.2334**

Table 4: MAP@5 and MAP@10 scores without Tafseer

The regression in GATE's performance could be attributed to several factors. This model has already been pretrained on large-scale Arabic corpora, and further fine-tuning likely introduced overfitting and reduced the model's ability to generalize. Additionally, the negative sampling strategies may not have been comprehensive enough to evaluate the reranker's ability to improve from the baseline. This suggests that, for already high-performing rerankers, there's a need for more careful design of fine-tuning data, otherwise it might be better to use the reranker without further training.

3.2.2 LLM-based Approach

We used Gemini 2.5 Flash (Comanici et al., 2025) to rerank retrieved documents by instructing it to get an ordered list of passage IDs that have the answers to a given question according to their relevance. The prompt design process included adding more instructions about the format of the answers to avoid hallucination of passages and emphasizing the importance of relevance and order of the returned passage IDs. The final prompt used is found in Appendix A.

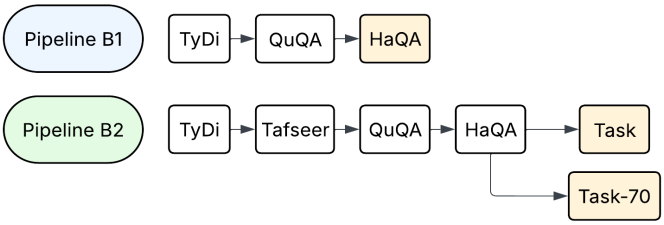

Figure 2: Figure showing cross-encoder finetuning configurations.

Model	Metric	Baseline	TyDi	Tafseer	QuQA	HaQA	Task	Task-70
GATE	MAP@5	**0.3172**	0.2319	0.2504	0.2318	0.2499	0.2107	0.2480
	MAP@10	**0.3215**	0.2503	0.2642	0.2563	0.2736	0.2367	0.2680
AraBERT	MAP@5	0.0278	0.1712	0.2099	0.1899	0.1884	0.1733	**0.2039**
	MAP@10	0.0371	0.1972	0.2099	0.2093	0.2081	0.1967	**0.2267**

Table 5: MAP@5 and MAP@10 cross-encoder scores on the dev set for full Pipeline B

Experimentation with different k values showed that higher values produced inconsistent results with Gemini, with MAP ranges varying drastically (Table 6). However, Gemini showed relatively reliable performance with the top 70 passages to filter across runs and models.

Pre and Post Retrieval Enhancements: To improve the performance of our pipeline, we experimented with two approaches: one for *pre-retrieval* and one for *post-retrieval*.

Our proposed technique for **pre-retrieval** is to use *topic filtering* before passing the question to our RAG model. This method uses Latent Dirichlet Allocation (LDA) to find the topics in the reranking stage (Ampazis, 2024). We applied it as a pre-retrieval technique by assigning, using the LLM, each question and Qur'anic passage a list of one or more topics out of 40 relevant topics in Islam, found in Appendix B. The filtering reduced the search space for the RAG model by providing it only with the documents matching the topics in the question to encode. Results in Table 7 show that performance increases without topic filtering, with MAP improving by **2%+**.

For **post-retrieval**, to enhance the LLM's understanding of the retrieved Qur'anic passages, we expanded each passage with its interpretation by aggregating the Jalalayn Tafseer. We observe that adding Tafseer reduced performance, as Gemini struggles with longer inputs, yielding at best MAP@10 of **0.15**.

Model Name	Top K	MAP@5	MAP@10
Baseline Model	70	0.2983	0.3137
	80	**0.3048**	**0.3294**
	100	0.2552	0.2902
Pipeline A	70	**0.3311**	**0.3579**
	80	0.2777	0.3049
	100	0.2913	0.3185
Pipeline B	70	0.3506	0.3801
	80	0.3550	0.3550
	100	**0.3598**	**0.3888**

Table 6: MAP@5 and MAP@10 scores for different models across varying Top K values.

Model Name	Top K	MAP@5	MAP@10
With Topic Modeling	30	0.3991	0.4299
	70	0.3958	0.4365
Without Topic Modeling	30	0.4228	0.4491
	70	**0.4407**	**0.4591**

Table 7: MAP@5 and MAP@10 scores for different models across varying Top K values.

4 Results

For evaluation on the test set, we chose three configurations: the first two use Gemini, with the first retrieving the top 70 most relevant documents from the combined collection Qur'an and Hadith passages, and the second retrieving 50 from Qur'an and 20 from Hadith to allow for higher representation of Hadith. The last approach also followed this method with the fine-tuned GATE model **(Pipeline B1)** used for filtering. It's important to note that when retrieving Hadith passages, we removed the diacritics from the texts. Results can be seen in

Table 8.

Model	MAP@10	MAP_Q@5	MAP_H@5
Gemini	0.1233	**0.1694**	0.1606
Gemini (50-20)	**0.1234**	0.1556	**0.2642**
GATE (50-20)	0.1159	0.1303	0.1774

Table 8: Subtask 2 Test Set Results

Gemini achieved a higher performance than GATE in both its configurations, with a significant boost in Hadith MAP when retrieving documents from Qur'an and Hadith individually. However, in all three test runs, we can observe a consistent and significant drop from dev set results.

This drop may be attributed to domain shift between the Qur'an-only development set and the mixed-source test set, or to overfitting on the fine-tuning data. While reranking with Gemini improved overall relevance, its performance on previously unseen questions proved less stable. Interestingly, the 50-20 retrieval strategy led to a much higher MAP for Hadith questions (MAP_H@5 = 0.2642), suggesting that explicitly balancing source representation improves answerability for Hadith-heavy queries. GATE performed more consistently but remained slightly behind Gemini, likely due to its more limited capacity to adapt to question semantics compared to LLM-based reranking.

5 Discussion

Our experiments on the retrieval model reveal 3 key insights. First, **positive-only fine-tuning consistently underperformed** compared to using hard negatives, as both cosine and contrastive losses benefit from distinguishing relevant from highly similar but irrelevant passages. Second, the **optimal top-K passages for positive and hard negative sampling** was typically **60–80 passages**; larger values often introduced easy negatives that weakened learning. Third, there was **no single best loss–search pairing**, with outcomes varying across settings. Finally, multi-stage curriculum (Pipeline B) consistently outperformed direct fine-tuning (Pipeline A), with **up to a 10% recall improvement** at higher Recall@K values. This demonstrates the advantage of gradual domain adaptation, moving from general Arabic QA to Qur'anic and Hadith retrieval, which helps the model capture the linguistic and semantic characteristics. For filtering, Gemini had a better performance; its understanding of the passages led to an increase of more than **5%** in MAP compared to the cross-encoder results. However, adding more context - whether by increasing the number of retrieved documents or by adding Tafseer - resulted in a substantial drop in scores.

6 Conclusion

In this study, we explore QA techniques for subtask B of the IslamicEval 2025 shared task. We compare direct fine-tuning and a multi-stage approach for retrieval, and a cross-encoder and LLM for reranking. Our experiments led to an increase in Recall for retrieval and MAP for reranking compared to prior models, demonstrating the potential of our approach for building more accurate and reliable Islamic QA systems.

Limitations The main challenge is dataset size and a lack of Hadith QA pairs. Additionally, Gemini fluctuated and produced inconsistent scores across runs. GPU limitations also prevented us from carrying out experiments using larger models. The limited timeline of our experiments also prevented us from exhausting all possible configurations, hyperparameters, and other approaches.

References

A. B. A Al-Sharjy and Z. Al-Zubaidi. 2009. *Al-Tajreed Al-Sareeh of Collective Sahih Hadith*. .

Sarah Alnefaie, Eric Atwell, and Mohammad Ammar Alsalka. 2023. Haqa and quqa: Constructing two arabic question-answering corpora for the quran and hadith. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 90–97.

Nikolaos Ampazis. 2024. Improving RAG quality for large language models with topic-enhanced reranking. In *Proceedings of the 2024 IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI 2024)*, pages 74–87, Cham. Springer Nature Switzerland.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, and et al. 2025. Gemini 2.5: Pushing

the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Mohammed ElKomy and Amany Sarhan. 2023. Tce at qur'an qa 2023 shared task: Low resource enhanced transformer-based ensemble approach for qur'anic qa. In *Proceedings of the Qur'an QA 2023 Shared Task*. Tanta University, Egypt.

Fajri Koto, Haonan Li, Sara Shatanawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Rana Malhas and Tamer Elsayed. 2020. Ayatec: Building a reusable verse-based test collection for arabic question answering on the holy qur'an. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(6).

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2022. Qur'an QA 2022: Overview of the first shared task on question answering over the holy qur'an. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 79–87, Marseille, France. European Language Resources Association.

Rana Malhas, Watheq Mansour, and Tamer Elsayed. 2023. Qur'an QA 2023 shared task: Overview of passage retrieval and reading comprehension tasks over the holy qur'an. In *Proceedings of ArabicNLP 2023*, pages 690–701, Singapore (Hybrid). Association for Computational Linguistics.

Hamza Shahid Mohaned A. Rashad. 2025. The arabic rag leaderboard. urlhttps://huggingface.co/spaces/Navid-AI/The-Arabic-Rag-Leaderboard.

Hamdy Mubarak, Rana Malhas, Watheq Mansour, Abubakr Mohamed, Mahmoud Fawzi, Majd Hawasly, Tamer Elsayed, Kareem Darwish, and Walid Magdy. 2025. IslamicEval 2025: The First Shared Task of Capturing LLMs Hallucination in Islamic Content. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

Omer Nacar and Anis Koubaa. 2024. Enhancing semantic similarity understanding in arabic nlp with nested embedding learning.

Omer Nacar, Anis Koubaa, Serry Sibaee, Yasser Al-Habashi, Adel Ammar, and Wadii Boulila. 2025. Gate: General arabic text embedding for enhanced semantic textual similarity with matryoshka representation learning and hybrid loss training.

NAMAA-Space. 2025. Gate-reranker-v1. https://huggingface.co/NAMAA-Space/GATE-Reranker-V1. Hugging Face model, Apache-2.0 license. Accessed: 15 August 2025.

Marwan N. Swar. 2007. *Mushaf Al-Tafseel Al Mawdoo'ee*. Dar Al-Fajr Al-Islami, Damascus.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 1503–1512, New York, NY, USA. Association for Computing Machinery.

A Prompting

The following prompt was used for filtering and reranking using Gemini2.5 Flash:

Given a question in Modern Standard Arabic (MSA) and a list of Quranic and Hadith verses (each with an associated ID), identify the IDs of the verses that contain the answer to the question.
Instructions:
*- Return only the **IDs** of the extremely relevant verses in a **list**, **ordered** from most relevant to least relevant.*
- Do not explain your answer or provide verse text.
*- If the answer is not found in any verse, or you are unsure, **you must return [-1]**.*
*- Use the verse ID **exactly as provided** (e.g., if the verse ID is 23:14-16, return [23:14-16]).*
Question: <QUESTION-TEXT>
Verses: <RETRIEVED-PASSAGES>

B Topic Modeling

To reduce the search space of the retrieval model, we adapted a pre-retreival topic filtering approach where we assign the questions and documents one or more of the topics from Table 9.

C Thresholding Experimentation

Using our best available model, the GATE baseline, we experimented with different values of the scoring threshold (T). Intuitively, the most optimal values lie between 0.10 - 0.20 as can be seen in Figure 3.

Topics	التوحيد، أسماء الله وصفاته، الملائكة، القدر، اليوم الآخر، الطهارة، الصلاة، الصيام، الزكاة، الحج والعمرة، الأذكار والدعاء، الزواج، الطلاق، قضايا المرأة، الميراث، تربية الأبناء، البيع والشراء، الربا، العقود، الصدقات، التأمين، المعاملات الحديثة، الطعام والشراب، الترفيه، الأخلاق، العلاقات مع غير المسلمين، قصص الأنبياء، الرؤى والرؤية، الصيام وشهر رمضان، الزكاة والصدقات، الذكر والدعاء والرقية الشرعية، العقيدة، النبوة والسيرة النبوية، الزواج وحقوق الزوجين، الميراث والوصايا، المعاملات التجارية والمالية الحديثة، الأطعمة والأشربة والمكونات الحلال والحرام، اللباس والزينة والتجميل، المباحات، الجنايات والقضاء والسياسة الشرعية

Table 9: The list of 40 topics assigned to questions and Qur'anic passages used for filtering before retrieval.

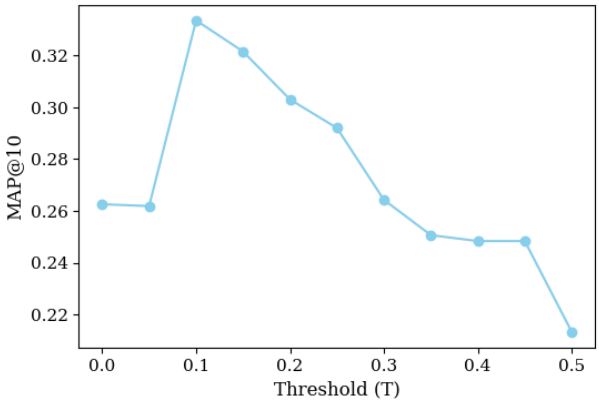

Figure 3: MAP@10 on GATE for Dev Set VS Threshold

D Arabic Embedding Model Evaluation

To identify suitable retriever models, we evaluated a broad set of Arabic (and multilingual) embedding models using cosine similarity ranking and Recall@30 on the Qur'anic development set. Due to limited computational resources, we were unable to run inference on larger-scale models (e.g., >500M parameters) with extensive batch processing, and thus prioritized models that were feasible for our hardware budget.

Model	Recall@30	Trainable Params (M)
NAMAA-Space/AraModernBert-Base-STS[1]	0.4451	149
silma-ai/silma-embeddding-sts-v0.1[2]	0.4136	135
omarelshehy/Arabic-Retrieval-v1.0[3]	0.3880	135
omarelshehy/Arabic-STS-Matryoshka-V2[4]	0.3876	135
Omartificial-Intelligence-Space/GATE-AraBert-v1(Nacar and Koubaa, 2024)	0.3663	135
ALJIACHI/bte-base-ar[5]	0.3627	149
mohamed2811/Muffakir_Embedding[6]	0.3576	135
silma-ai/silma-embeddding-matryoshka-v0.1[7]	0.3517	135
Omartificial-Intelligence-Space/Arabic-Triplet-Matryoshka-V2(Nacar and Koubaa, 2024)	0.3478	135
AhmedZaky1/arabic-bert-sts-matryoshka[8]	0.3235	135
Alibaba-NLP/gte-multilingual-base[9]	0.3073	305
Omartificial-Intelligence-Space/Arabert-all-nli-triplet-Matryoshka(Nacar and Koubaa, 2024)	0.3053	135
AhmedZaky1/arabic-bert-nli-matryoshka[10]	0.3028	135
AhmedZaky1/DIMI-embedding-v2[11]	0.2924	305
ibm-granite/granite-embedding-278m-multilingual[12]	0.2701	278
omarelshehy/arabic-english-sts-matryoshka-v2.0[13]	0.2680	560
OmarAlsaabi/e5-base-mlqa-finetuned-arabic-for-rag[14]	0.2622	278
intfloat/multilingual-e5-base(Wang et al., 2024)	0.2599	278
ibm-granite/granite-embedding-107m-multilingual[15]	0.2598	107
Abdelkareem/zaraah_jina_v3[16]	0.2443	64
AhmedZaky1/DIMI-embedding-v4[17]	0.2322	305
Snowflake/snowflake-arctic-embed-m-v2.0[18]	0.1745	305
Abdelkareem/abjd[19]	0.1677	438
Abdelkareem/ara-qwen3-18[20]	0.1677	438
Omartificial-Intelligence-Space/Arabic-labse-Matryoshka(Nacar and Koubaa, 2024)	0.1579	471
sentence-transformers/LaBSE[21]	0.1575	471
Omartificial-Intelligence-Space/Arabic-MiniLM-L12-v2-all-nli-triplet[22]	0.0973	118
mixedbread-ai/mxbai-embed-large-v1(Li and Li, 2023)	0.0357	335
metga97/Modern-EgyBert-Base[23]	0.0145	159
metga97/Modern-EgyBert-Embedding[24]	0.0145	159
sentence-transformers/all-mpnet-base-v2[25]	0.0057	109
sentence-transformers/all-MiniLM-L6-v2[26]	0.0008	23

Table 10: Recall@30 and parameter counts for reviewed sentence embedding models on the Qur'anic dev set.

[1] https://huggingface.co/NAMAA-Space/AraModernBert-Base-STS
[2] https://huggingface.co/silma-ai/silma-embedding-sts-0.1
[3] https://huggingface.co/omarelshehy/Arabic-Retrieval-v1.0
[4] https://huggingface.co/omarelshehy/Arabic-STS-Matryoshka-V2
[5] https://huggingface.co/ALJIACHI/bte-base-ar
[6] https://huggingface.co/mohamed2811/Muffakir_Embedding
[7] https://huggingface.co/silma-ai/silma-embedding-matryoshka-0.1
[8] https://huggingface.co/AhmedZaky1/arabic-bert-sts-matryoshka
[9] https://huggingface.co/Alibaba-NLP/gte-multilingual-base
[10] https://huggingface.co/AhmedZaky1/arabic-bert-nli-matryoshka
[11] https://huggingface.co/AhmedZaky1/DIMI-embedding-v2
[12] https://huggingface.co/ibm-granite/granite-embedding-278m-multilingual
[13] https://huggingface.co/omarelshehy/arabic-english-sts-matryoshka-v2.0
[14] https://huggingface.co/OmarAlsaabi/e5-base-mlqa-finetuned-arabic-for-rag
[15] https://huggingface.co/ibm-granite/granite-embedding-107m-multilingual
[16] https://huggingface.co/Abdelkareem/zaraah_jina_v3
[17] https://huggingface.co/AhmedZaky1/DIMI-embedding-v4
[18] https://huggingface.co/Snowflake/snowflake-arctic-embed-m-v2.0
[19] https://huggingface.co/Abdelkareem/abjd
[20] https://huggingface.co/Abdelkareem/ara-qwen3-18
[21] https://huggingface.co/sentence-transformers/LaBSE
[22] https://huggingface.co/Omartificial-Intelligence-Space/Arabic-MiniLM-L12-v2-all-nli-triplet
[23] https://huggingface.co/metga97/Modern-EgyBert-Base
[24] https://huggingface.co/metga97/Modern-EgyBert-Embedding
[25] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[26] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Passages	Loss Function	Search Method	Recall@30	Recall@50	Recall@70
Positive only	Cosine	Cosine	0.269	0.320	0.341
	Cosine	L2	0.211	0.250	0.283
	Contrastive	Cosine	0.285	0.329	0.385
	Contrastive	L2	0.242	0.288	0.346
30	Cosine	Cosine	0.417	0.480	0.537
	Cosine	L2	0.439	0.497	0.523
	Contrastive	Cosine	0.440	0.497	0.537
	Contrastive	L2	0.447	0.494	0.548
50	Cosine	Cosine	0.425	0.488	0.531
	Cosine	L2	0.431	0.476	0.501
	Contrastive	Cosine	0.462	**0.537**	0.555
	Contrastive	L2	0.457	0.536	0.555
70	Cosine	Cosine	0.446	**0.537**	**0.592**
	Cosine	L2	0.424	0.496	0.545
	Contrastive	Cosine	0.472	0.510	0.583
	Contrastive	L2	**0.491**	0.521	0.552
90	Cosine	Cosine	0.436	0.494	0.558
	Cosine	L2	0.428	0.466	0.501
	Contrastive	Cosine	0.477	0.517	0.559
	Contrastive	L2	0.460	0.518	0.555

Table 11: Performance of Fine-Tuned Configurations (Pipeline A) on Dev Set (Quran)

Passages	Loss Function	Search Method	Recall@30	Recall@50	Recall@70
60	Cosine	Cosine	0.508	0.586	0.675
	Cosine	L2	0.505	0.600	**0.688**
	Contrastive	Cosine	0.539	0.603	0.621
	Contrastive	L2	0.538	0.602	0.634
70	Cosine	Cosine	0.521	0.596	0.663
	Cosine	L2	0.464	0.577	0.636
	Contrastive	Cosine	**0.541**	0.581	0.640
	Contrastive	L2	0.539	0.606	0.634
80	Cosine	Cosine	0.446	0.548	0.646
	Cosine	L2	0.462	0.501	0.574
	Contrastive	Cosine	0.520	0.619	0.649
	Contrastive	L2	0.537	**0.620**	0.645

Table 12: Performance of Fine-Tuned Configurations (Pipeline B) on Dev Set (Quran)