GATE-Reranker: A Strong Arabic Cross-Encoder for Document Reranking

Omer Nacar[1] **Mohamed Zaytoon**[2,6] **Omar Elshehy**[3,6] **Khloud Al Jallad**[4,6]
[1]Tuwaiq Academy
[2]Alexandria University , [3]Universität des Saarlandes , [4]Arab International University , [6]NAMAA Community
o.najar@tuwaiq.edu.sa

Abstract

Arabic information retrieval increasingly relies on multi-stage pipelines in which a fast first-stage retriever produces candidate passages and a neural reranker refines relevance. While transformer cross-encoders deliver strong effectiveness through joint query–passage encoding, multilingual rerankers achieve competitive performance on Arabic benchmarks. However, systematic analysis of calibration, robustness, and deployment behavior in Arabic-specific settings remains limited. We present **GATE-Reranker**, a compact Arabic cross-encoder initialized from an Arabic semantic embedding backbone and fine-tuned on large-scale mMARCO-style Arabic triplets. The model scores each query–passage pair via full self-attention and a lightweight regression head, enabling plug-and-play second-stage reranking for Arabic search and RAG systems. We evaluate on three Arabic benchmarks covering binary relevance discrimination, controlled multi-negative reranking, and large-scale mMARCO evaluation. While remaining competitive with strong multilingual rerankers in ranking effectiveness, GATE-Reranker demonstrates significantly improved calibration and discriminative behavior. These properties translate into more reliable downstream performance in retrieval and RAG pipelines, while maintaining low GPU memory and latency on a Tesla T4.

Keywords: Arabic, information retrieval, reranking, cross-encoder, mMARCO

1. Introduction

Document reranking has become a central component of modern Information Retrieval (IR) systems. In multi-stage retrieval pipelines, a first-stage retriever efficiently narrows down a large corpus to a manageable candidate set, after which a more expressive reranker refines the final ranking. Cross-encoder architectures, built upon large pre-trained language models, have emerged as a dominant reranking paradigm due to their ability to jointly encode query–document pairs and leverage full self-attention for fine-grained semantic interaction (Déjean et al., 2024). By modeling contextualized token-level relationships across the query and the document, cross-encoders consistently achieve strong ranking effectiveness. Alternative interaction mechanisms, such as late-interaction models (e.g., ColBERT) and listwise LLM-based rerankers, aim to balance effectiveness and efficiency, but often introduce additional computational cost or complexity.

Most high-performing rerankers are trained primarily on English-centric datasets such as MS MARCO (Bajaj et al., 2016) and evaluated on widely adopted benchmarks like BEIR (Thakur et al., 2021). While multilingual models nominally support Arabic, they are rarely optimized specifically for its linguistic properties, including rich morphology, orthographic variation, and syntactic flexibility.

In this work, we introduce **GATE-Reranker**, a compact Arabic cross-encoder reranker trained on large-scale Arabic MS MARCO-style triplets. Our learning formulation follows *pointwise relevance regression*: each query–passage pair is assigned a continuous score, and the model is optimized to score relevant pairs higher than non-relevant pairs. The model processes the query and passage jointly within a single input sequence, enabling full cross-attention and contextual interaction across segments. Training is conducted using curated Arabic triplets with controlled positive-to-negative sampling, and optimization is performed using mixed-precision to ensure stable and efficient training.

Rather than proposing a new interaction mechanism, our contribution lies in establishing a robust, reproducible, and empirically validated Arabic cross-encoder baseline, together with systematic Arabic-focused evaluation. We evaluate across multiple Arabic reranking benchmarks spanning binary relevance, controlled multi-negative reranking, and large-scale mMARCO evaluation. We further analyze sensitivity to retrieval depth and report efficiency measurements on a Tesla T4 GPU, allowing direct comparison of effectiveness–latency trade-offs. Overall, our results show that an Arabic-specialized cross-encoder can achieve competitive effectiveness relative to strong multilingual rerankers while offering favorable deployment characteristics for Arabic retrieval and RAG pipelines.

By grounding our study in systematic experimentation and Arabic-specific evaluation, we aim to provide a solid empirical foundation for Arabic neural

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 40–48

reranking, showing that Arabic-specialized cross-encoders can achieve competitive effectiveness while offering favourable efficiency trade-offs for deployment.

Approach	Objective	Interaction	Practical Limitation
BERT rerankers (Nogueira and Cho, 2019)	Pointwise	Cross-encoder	High per-pair inference cost
ColBERT (Khattab and Zaharia, 2020)	Pairwise	Late interaction	Multi-vector indexing/storage overhead
Jina reranker v3 (Wang et al., 2025)	Listwise	Joint attention	Cost grows with list/context length
RankGPT (Huang et al., 2025)	Listwise	Generative	Prompt and ordering sensitivity
BGE reranker v2-m3 (Chen et al., 2024)	Pointwise	Cross-encoder	General-purpose; not Arabic-optimized
GATE-Reranker (ours)	Pointwise	Cross-encoder	Second-stage reranking (top-k)

Table 1: Comparison of representative reranking approaches.

2. Related Work

Document reranking methods are typically characterized by their training objectives and query–document interaction mechanisms. Table 1 summarizes representative approaches, highlighting trade-offs between effectiveness and efficiency.

2.1. Learning-to-Rank Paradigms

Reranking approaches are commonly categorized into pointwise, pairwise, and listwise methods. Pointwise models predict absolute relevance scores for each query–document pair, while pairwise approaches learn relative preferences between documents (e.g., RankNet (Burges et al., 2005), PRP (Qin et al., 2024)). Listwise methods optimize ranking metrics over multiple candidates simultaneously (Wang et al., 2013), often leveraging large language models such as Jina Reranker v3 (Wang et al., 2025) and RankGPT (Huang et al., 2025). While listwise approaches can model richer interactions, they typically incur higher computational cost and increased sensitivity to input ordering.

2.2. Neural Ranking Architectures

Neural rerankers are commonly implemented as bi-encoders or cross-encoders. Bi-encoders independently encode queries and documents into fixed representations, enabling efficient large-scale retrieval but limiting token-level interaction (e.g., DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021)). In contrast, cross-encoders jointly encode query–document pairs with full self-attention, enabling fine-grained interaction and improved ranking accuracy (e.g., BERT rerankers (Nogueira and Cho, 2019), MonoBERT (Nogueira et al., 2019)), at the cost of higher inference complexity.

2.3. LLM-Based Rerankers

Recent work explores LLM-based reranking using generative or long-context formulations. Generative approaches such as RankGPT (Huang et al., 2025) produce ordered outputs via prompting, while long-context models such as jina-reranker-v3 (Wang et al., 2025) jointly process multiple candidates. Although effective, these methods are computationally expensive and sensitive to prompt design, while traditional cross-encoders remain competitive and more efficient in production settings (Déjean et al., 2024).

2.4. Multilingual and Arabic Reranking

Recent multilingual rerankers, such as BGE v2-m3 (Chen et al., 2024; Li et al., 2023) and Qwen-based models (Zhang et al., 2025), demonstrate strong performance across languages, including Arabic. However, most prior work focuses primarily on ranking effectiveness, with limited analysis of calibration, robustness, and deployment behavior in Arabic-specific settings. Our work addresses this gap by emphasizing calibration-aware evaluation and reliability-focused analysis.

Arabic representation backbones. The quality of the underlying encoder is a key factor in reranking performance. Nacar et al. (2025) introduce GATE, a General Arabic Text Embedding model trained with multi-task objectives and Matryoshka representation learning. We initialize our reranker from a GATE-based checkpoint and adapt it to cross-encoder reranking via end-to-end fine-tuning on Arabic mMARCO-style triplets.

3. Methodology

This section presents the architecture, learning formulation, data construction strategy, and optimization procedure used to train **GATE-Reranker**. The proposed system follows a transformer cross-encoder design trained with *pointwise relevance regression* on positive/negative query–passage pairs derived from Arabic triplets. An overview of the architecture is illustrated in Figure 1.

3.1. Model Architecture

GATE-Reranker is implemented as a transformer-based cross-encoder. Given a query q and a pas-

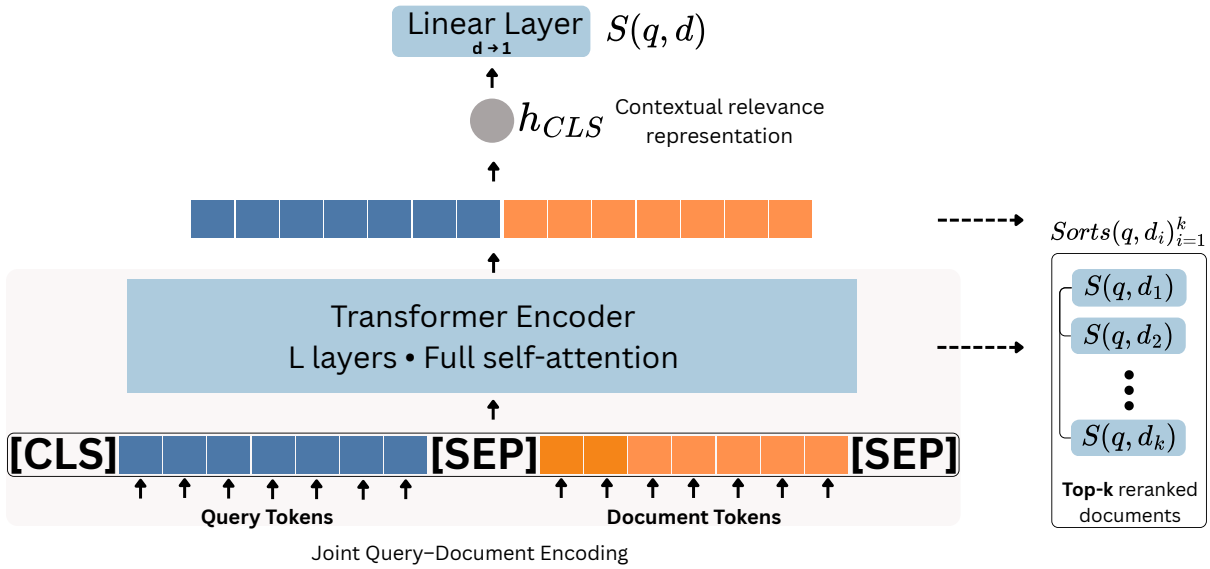

Figure 1: Architecture of GATE-Reranker. The query and document are concatenated and jointly encoded using a transformer-based cross-encoder with full self-attention. The contextualized representation of the [CLS] token is passed through a linear regression head to produce a relevance score $s(q, d)$. Candidate documents are reranked by sorting predicted scores within the top-k set.

sage d, the model constructs a single input sequence:

$$x = [\text{CLS}]\ q\ [\text{SEP}]\ d\ [\text{SEP}]. \tag{1}$$

The concatenated sequence is processed by a transformer encoder composed of L layers with full self-attention. Unlike bi-encoder architectures that encode queries and documents independently, the cross-encoder jointly models token-level interactions across both segments. This enables contextual alignment between query and document tokens through shared attention mechanisms.

Let $H = \text{Encoder}_\theta(x)$ denote the contextualized hidden representations produced by the transformer. The final hidden state corresponding to the [CLS] token, denoted by h_{CLS}, is used as a holistic relevance representation:

$$h_{\text{CLS}} = H_0. \tag{2}$$

A linear regression head then maps this representation to a scalar relevance score:

$$s(q, d) = W h_{\text{CLS}} + b, \tag{3}$$

where W and b are learnable parameters. The resulting score $s(q, d)$ reflects the model's estimate of semantic relevance between the query and the passage.

Backbone initialization from GATE. We initialize the encoder from a GATE-based Arabic embedding model trained with multi-task objectives (Nacar et al., 2025).

3.2. Training Objective

We adopt a *pointwise* learning formulation expressed as binary relevance regression. For each query, we construct positive and negative query–passage pairs. Positive pairs are labeled with $y = 1$, while negative pairs are labeled with $y = 0$.

Given a mini-batch of N training instances $\{(q_i, d_i, y_i)\}_{i=1}^{N}$, the model is optimized using the mean squared error (MSE) loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(s(q_i, d_i) - y_i \right)^2. \tag{4}$$

This objective encourages higher scores for relevant passages and lower scores for non-relevant ones. At inference time, ranking is performed by sorting passages according to their predicted scores.

3.3. Justification of Pointwise Regression

We adopt a pointwise regression formulation with MSE loss for its simplicity, stability, and compatibility with score calibration. Unlike pairwise or listwise objectives that optimize relative ordering, pointwise regression directly models absolute relevance scores, enabling better calibration and interpretability.

While pairwise and listwise approaches can improve ranking performance, they often produce uncalibrated scores that are harder to interpret and threshold in practice. In contrast, our objective pri-

oritizes score reliability and deployment readiness, rather than purely optimizing ranking metrics.

As shown in Section 6, this choice leads to well-separated and calibrated score distributions, which are critical for downstream applications requiring threshold-based decisions.

3.4. Training Data Construction

Training data is derived from Arabic MS MARCO-style triplets (Bajaj et al., 2016) consisting of a query q, a relevant passage d^+, and a non-relevant passage d^-. For each triplet, we generate two training instances:

$$(q, d^+, 1), \quad (q, d^-, 0). \tag{5}$$

We employ a balanced positive-to-negative sampling ratio of $1{:}1$, ensuring stable optimization and preventing negative dominance during training. The data is constructed from curated Arabic queries and passages within the mMARCO collection. A separate development subset is maintained for intermediate evaluation during training.

The model is trained with a maximum sequence length of 512 tokens and a batch size of 8. The relatively small batch size accommodates the computational cost of joint encoding at full context length. We train for a single epoch to avoid overfitting and preserve the semantic structure of the pretrained backbone. Empirically, additional epochs did not yield consistent improvements and occasionally degraded calibration performance.

Automatic Mixed Precision (AMP) is enabled to improve computational efficiency and reduce memory footprint without compromising numerical stability. During training, periodic evaluation is conducted using a cross-encoder reranking evaluator constructed from held-out query–positive–negative samples.

Training was conducted using PyTorch and SentenceTransformers. The model was fine-tuned with a batch size of 8, maximum sequence length of 512 tokens, and trained for one epoch using the AdamW optimizer with a learning rate of 2e-5. Mixed precision (AMP) was enabled for efficiency. All experiments were conducted on a single Tesla T4 GPU.

4. Experiments

This section describes the evaluation setup used to assess GATE-Reranker. We detail the benchmarks, comparison models, evaluation metrics, and experimental protocol, followed by an ablation analysis designed to isolate the effect of retrieval depth on Arabic reranking.

4.1. Benchmarks

We evaluate on three Arabic reranking benchmarks covering both large-scale retrieval and controlled reranking settings.

The first benchmark is the Arabic portion of the mMARCO development set. This dataset follows the MS MARCO reranking protocol, where each query is associated with a candidate set produced by a first-stage retriever over the full Arabic collection. We use this benchmark to assess reranking behavior at scale (55,560 evaluation queries over a collection of approximately 8.8M passages).

The second benchmark, *Arabic-Reranking-Triplet-5-Eval*, consists of 500 Arabic queries, each paired with one relevant passage and four non-relevant passages. This controlled multi-negative setup evaluates whether a reranker can consistently place the positive passage above several competitive distractors under a fixed candidate set size ($k = 5$).

The third benchmark, *Ar-Reranking-Eval*, is a binary relevance dataset of 500 samples in which each query is paired with candidate documents labeled as relevant (1) or non-relevant (0). This benchmark primarily evaluates pair-level relevance discrimination and serves as a sanity check for Arabic query–passage matching.

Together, these datasets provide complementary perspectives: large-scale realistic reranking (mMARCO), controlled multi-negative competition (Triplet-5), and binary discrimination (Ar-Reranking-Eval).

4.2. Compared Models

Our primary contribution in this paper is **GATE-Reranker-V1**, which is trained using the methodology described in Section 3 and is designed to be a compact Arabic cross-encoder suitable for practical second-stage reranking. We additionally report results for a higher-capacity scaling variant of the same training recipe, denoted **GATE-Reranker-LC**. We treat GATE-Reranker-LC as an internal scaling variant rather than a separate competing method, enabling an explicit effectiveness–efficiency comparison within the same Arabic reranker family while keeping the narrative centered on GATE-Reranker-V1.

For external comparison, we evaluate publicly available rerankers spanning multilingual and Arabic-focused cross-encoder models, including mMARCO MiniLM (`cross-encoder/mmarco-mMiniLMv2-L12-H384-v1`), BGE-Reranker-v2-m3 (`BAAI/bge-reranker-v2-m3`), Mizan-Rerank-v1 (`ALJIACHI/Mizan-Rerank-v1`), lightblue/lb-reranker-0.5B-v1.0, and mxbai-rerank-base-v2.

4.3. Evaluation Metrics

We report standard Information Retrieval metrics: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and nDCG@10. For controlled multi-negative settings, correctness at the top rank is also considered when informative. All metrics are computed consistently across models.

4.4. Evaluation Protocol

All models are evaluated in a pure reranking setting. For each query q and its candidate set $\{d_i\}_{i=1}^{k}$, the reranker computes relevance scores independently:

$$s_i = s(q, d_i), \tag{6}$$

and produces the final ranking by sorting scores in descending order:

$$\text{Rank}(q) = \text{Sort}\left(\{s(q, d_i)\}_{i=1}^{k}\right). \tag{7}$$

For *Arabic-Reranking-Triplet-5-Eval*, $k = 5$ by construction. For *Ar-Reranking-Eval*, scoring is performed directly on labeled query–document pairs. For mMARCO Arabic dev, candidate sets are produced by a first-stage retriever; in the ablation study below, we use an oracle-controlled construction to ensure the relevant passage is present and to isolate reranker behavior from retriever recall.

4.5. Ablation Study: Effect of Retrieval Depth

On mMARCO Arabic dev, we conduct an ablation study that isolates the effect of varying the number of candidates reranked (k). For each query, we assume a single positive (relevant) passage and treat all remaining candidates as negatives (irrelevant). We evaluate reranking under $k = 2$ and $k = 3$ by pairing the positive passage with one or two negatives sampled from the retrieval pool. This setup enables a controlled analysis of how additional negatives and retrieval depth influence the model's ability to distinguish relevant from irrelevant passages.

5. Results

This section reports the effectiveness and efficiency of our Arabic reranker family across multiple benchmarks. We focus on **GATE-Reranker-V1** as the primary model described in this paper, and additionally report a higher-capacity scaling variant, denoted **GATE-Reranker-LC**. We treat it as a scaling variant of the same training recipe to analyze the accuracy–efficiency trade-off, while keeping the paper centered on GATE-Reranker-V1. We then present a controlled retrieval-depth ablation on mMARCO Arabic dev, a GPU efficiency study, and domain-oriented results from the Arabic RAG Leaderboard (Mohaned A. Rashad, 2025). Importantly, we observe that multilingual rerankers, while competitive in ranking metrics, exhibit severe calibration issues and near-zero discriminative margins (Section 6), limiting their reliability in downstream retrieval pipelines.

Model	MRR	MAP	nDCG@10
GATE-Reranker-V1	1.000	1.000	1.000
GATE-Reranker-LC	1.000	1.000	1.000
mMARCO-mMiniLMv2	1.000	1.000	1.000
BGE-Reranker-v2-m3	1.000	1.000	1.000
Mizan-Rerank-v1	0.998	0.998	0.998
mxbai-rerank-base-v2	0.991	0.990	0.993
lb-reranker-0.5B	0.942	0.943	0.958

Table 2: Results on Ar-Reranking-Eval.

5.1. Ar-Reranking-Eval

Table 2 reports results on *Ar-Reranking-Eval*. Several cross-encoder rerankers achieve ceiling performance, including our models (**GATE-Reranker-V1** and **GATE-Reranker-LC**) as well as strong multilingual baselines (mMARCO-mMiniLMv2 and BGE-Reranker-v2-m3). This saturation indicates that the benchmark provides limited ranking ambiguity for modern rerankers—the relevant passage is often highly separable from the non-relevant candidates.

While *Ar-Reranking-Eval* therefore serves as a useful sanity check that confirms the effectiveness of our Arabic reranker family under binary relevance conditions, it is not sufficiently discriminative for comparing top-tier systems. For this reason, we rely on the more competitive multi-negative evaluation (Triplet-5) and the large-scale mMARCO setting to characterize robustness and practical trade-offs.

Model	MRR	MAP	nDCG@10
mMARCO-mMiniLMv2	0.999	0.999	0.9993
BGE-v2-m3	0.999	0.999	0.9993
GATE-Reranker-LC	0.997	0.997	0.9978
GATE-Reranker-V1	0.996	0.996	0.9970
Mizan-Rerank-v1	0.9799	0.9799	0.9850
lb-reranker-0.5B	0.5610	0.5610	0.6700
mxbai-rerank-base-v2	0.4724	0.4724	0.6019

Table 3: Results on Arabic-Reranking-Triplet-5-Eval (500 queries, $k = 5$).

5.2. Arabic-Reranking-Triplet-5-Eval

Table 3 reports results on *Arabic-Reranking-Triplet-5-Eval*, a controlled multi-negative benchmark ($k = 5$) that better separates model quality. Unlike the saturated binary setting, this benchmark reveals meaningful differences among rerankers. Strong multilingual cross-encoders such as mMARCO-mMiniLMv2 and BGE-v2-m3 achieve the highest effectiveness, while our Arabic reranker family

Model	MRR	MAP	nDCG@10
$k = 2$ *(oracle-controlled)*			
BGE-v2-m3	0.99933	0.99931	0.99941
GATE-Reranker-LC	0.99877	0.99873	0.99898
mMARCO-mMiniLMv2	0.99842	0.99834	0.99871
GATE-Reranker-V1	0.99825	0.99814	0.99857
Mizan-Rerank-v1	0.95817	0.95717	0.96860
lb-reranker-0.5B	0.76811	0.76564	0.82786
mxbai-rerank-base-v2	0.69286	0.69288	0.77396
$k = 3$ *(oracle-controlled)*			
BGE-v2-m3	0.99922	0.99919	0.99932
GATE-Reranker-LC	0.99821	0.99816	0.99856
GATE-Reranker-V1	0.99792	0.99779	0.99831
mMARCO-mMiniLMv2	0.99745	0.99734	0.99797
Mizan-Rerank-v1	0.95203	0.95082	0.96387
lb-reranker-0.5B	0.65198	0.64985	0.74061
mxbai-rerank-base-v2	0.51829	0.51697	0.64103

Table 4: mMARCO Arabic dev ablation showing the effect of retrieval depth ($k = 2$ vs. $k = 3$).

(**GATE-Reranker-V1** and **GATE-Reranker-LC**) follows closely with consistently high performance. In contrast, more general rerankers (e.g., mxbai-rerank-base-v2 and lb-reranker) degrade substantially, suggesting that Arabic relevance modeling under multi-negative competition benefits from Arabic-focused training signals.

Crucially, the gap between **GATE-Reranker-V1** and the best-performing models is small, supporting the central motivation of this work: a dedicated Arabic cross-encoder can achieve near state-of-the-art reranking quality while remaining practical under deployment constraints.

5.3. Ablation: Retrieval Depth on mMARCO Arabic Dev

Table 4 summarizes retrieval-depth ablation on mMARCO Arabic dev. Increasing k from 2 to 3 introduces an additional negative passage per query, increasing ranking difficulty. Top rerankers remain stable, whereas weaker models show larger degradations, indicating sensitivity to additional distractors. Both of our models are robust under this shift, and **GATE-Reranker-V1** remains close to the best multilingual baselines, supporting its suitability for practical multi-stage retrieval where candidate lists often contain multiple hard negatives.

Importantly, comparing **GATE-Reranker-V1** against **GATE-Reranker-LC** isolates the effect of scaling within the same training recipe: the higher-capacity variant provides a modest effectiveness gain, which we later contextualize against its substantially higher inference cost (Section 5.4).

5.4. Inference Efficiency on Tesla T4

Table 5 reports inference efficiency on a Tesla T4 GPU (approximately 14.6 GB VRAM), measured over 250 iterations with 5 warmup steps. The results clarify an important practical trade-off: although scaling can provide small effectiveness gains (as seen for **GATE-Reranker-LC**), the inference cost may become prohibitive for latency-sensitive pipelines.

GATE-Reranker-V1 provides a particularly strong effectiveness–efficiency balance: it achieves near state-of-the-art ranking quality while using substantially less GPU memory and offering low per-pair latency. In contrast, **GATE-Reranker-LC** is significantly slower and more memory-intensive, illustrating that the compact model is better suited for real-time reranking or reranking larger candidate sets. Heavier rerankers such as lb-reranker-0.5B and mxbai-rerank-base-v2 are substantially slower still, making them challenging to deploy at scale.

Model	Latency (ms)	Pairs/s	GPU Mem (MB)
mMARCO-mMiniLMv2	19.62	1274.01	2691.80
GATE-Reranker-V1	28.55	875.80	516.73
Mizan-Rerank-v1	45.95	544.08	3128.02
GATE-Reranker-LC	150.95	165.62	2691.80
BGE-v2-m3	161.59	154.71	3128.02
mxbai-rerank-base-v2	278.08	89.90	3575.04
lb-reranker-0.5B	1115.61	22.41	4091.27

Table 5: Inference efficiency on Tesla T4 (250 iterations; lower latency and memory are better, higher throughput is better).

5.5. Domain-Oriented Evaluation on the Arabic RAG Leaderboard

To assess domain-oriented performance in realistic Arabic RAG scenarios, we additionally report scores from the Arabic RAG Leaderboard (Mohaned A. Rashad, 2025), which evaluates rerankers across multiple domain slices (e.g., global knowledge, tourism, and legal). Table 6 summarizes results for selected rerankers. Notably, **GATE-Reranker-V1** remains competitive across domains despite operating at a shorter context length (512 tokens), whereas several strong multilingual rerankers and our scaling variant **GATE-Reranker-LC** leverage longer context windows (e.g., 8192 tokens). This comparison highlights that Arabic-specialized reranking can deliver strong domain performance even under constrained input budgets, while also offering a clear path for scaling when longer-context deployment is feasible.

6. Error Analysis and Score Calibration

While aggregate ranking metrics provide a high-level comparison, they do not capture how reliably a reranker separates relevant from non-relevant documents. We therefore conduct a detailed error analysis focusing on score calibration, discriminative margin, and false-positive behavior.

Model	Avg	Global	Tourism	Legal
BGE-v2-m3	87.44	81.27	80.96	88.58
GATE-Reranker-LC	85.82	80.18	77.70	87.62
gte-multilingual-reranker-base	85.03	76.76	77.10	85.87
GATE-Reranker-V1	83.96	77.02	79.60	84.41
Mizan-Rerank-v1	76.26	71.68	70.80	77.21

Table 6: Arabic RAG Leaderboard scores for selected rerankers (subset of domains shown for brevity).

6.1. Calibration and Discriminative Power

Table 7 reports calibration and discrimination metrics for **GATE-Reranker** and a strong multilingual baseline (MiniLM-L12-v2).

We observe that **GATE-Reranker** achieves strong calibration (low ECE and Brier score) and a large discriminative margin, while MiniLM exhibits severe miscalibration and near-zero margin. Notably, MiniLM assigns almost identical scores to both positive and negative samples, indicating a collapse of the scoring function.

6.2. Large-Margin Separation

We measure the average margin between positive and negative samples:

$$\Delta = E[s(q, d^+) - s(q, d^-)]. \tag{8}$$

GATE-Reranker achieves a margin of 0.9382, while MiniLM produces a near-zero margin of 0.0002. This difference spans several orders of magnitude, indicating that GATE-Reranker forms a clear decision boundary, whereas MiniLM fails to meaningfully separate relevant from non-relevant documents.

6.3. False Positive Behavior

We further analyze the false positive rate at high recall (FPR@95TPR). GATE-Reranker achieves 0.0, indicating perfect rejection of negative samples under high recall conditions. In contrast, MiniLM reaches 0.6453, meaning that a large proportion of negative documents are incorrectly classified as relevant.

This result highlights a critical limitation of multilingual rerankers: while they may achieve reasonable ranking metrics, they can produce unreliable decisions when used for filtering or threshold-based retrieval.

6.4. Score Distribution Analysis

Distribution Analysis. Figure 2 provides a visual explanation of the observed quantitative differences. GATE-Reranker produces a well-separated bimodal distribution, with relevant documents concentrated near 1 and non-relevant documents near 0. This separation reflects strong discriminative capacity and enables reliable threshold-based decision-making.

In contrast, MiniLM exhibits a collapsed distribution where both positive and negative samples are concentrated in a narrow region near 1 (right panel, zoomed). This collapse indicates that the model fails to meaningfully separate relevant from non-relevant documents, explaining its near-zero margin and high false positive rate.

Notably, MiniLM assigns nearly identical score distributions to both classes, effectively collapsing the scoring function into a non-discriminative regime.

6.5. Implications for Retrieval Systems

These findings highlight an important distinction between ranking performance and decision reliability. While multilingual rerankers may remain competitive on ranking metrics, they often exhibit poor calibration and limited robustness to topical noise. In contrast, GATE-Reranker produces well-separated and calibrated scores, enabling more reliable relevance estimation. We attribute these gains to the pointwise regression formulation and the Arabic-specific semantic initialization, which together encourage consistent score scaling and stronger separation between relevant and non-relevant documents. As shown in Section 6, this results in significantly larger margins and lower false positive rates compared to multilingual baselines.

These improvements have direct implications for downstream systems such as RAG pipelines. Well-calibrated scores enable the use of global thresholds to filter irrelevant documents before generation, reducing noise propagation and improving answer quality. In contrast, poorly calibrated rerankers may pass irrelevant documents despite correct ranking order, leading to degraded generation and increased hallucination risk. Furthermore, strong score separation allows more effective top-k pruning, where fewer but higher-quality documents are selected for downstream processing. This reduces computational cost while maintaining relevance, improving both efficiency and reliability in end-to-end retrieval systems.

6.6. Effectiveness vs. Reliability Trade-off

Although GATE-Reranker does not always outperform strong multilingual rerankers in ranking metrics, it provides significantly better calibration and discriminative margins. This suggests a trade-off between ranking optimization and score reliability.

Multilingual models may achieve strong ranking performance by leveraging large-scale cross-

Model	ECE ↓	Brier ↓	Margin ↑	FPR@95 ↓
GATE-Reranker	0.0316	0.0077	0.9382	0.0000
MiniLM-L12-v2	0.4995	0.4994	0.0002	0.6453

Table 7: Calibration and discriminative analysis. Lower is better for ECE, Brier score, and FPR@95TPR; higher is better for margin.

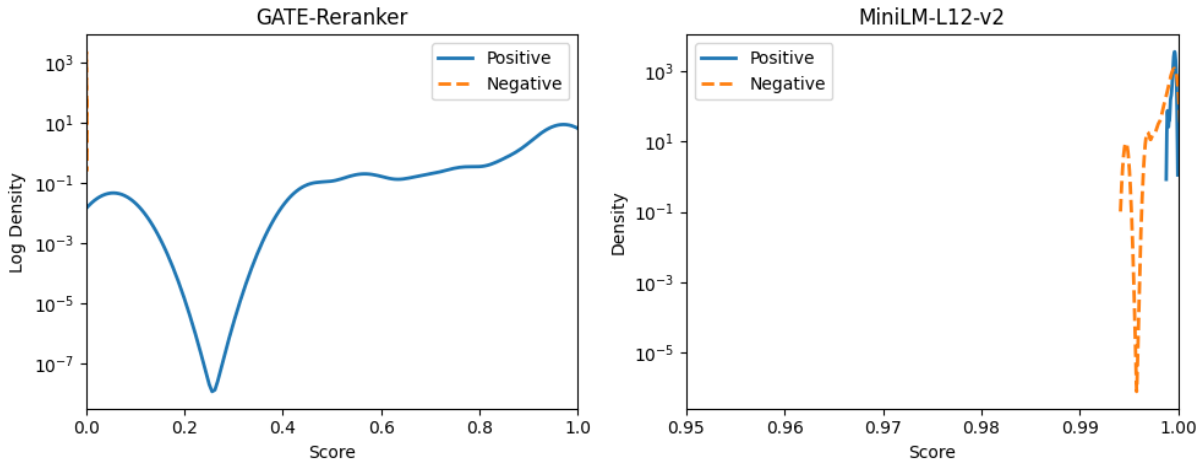

Figure 2: Log-scale score distributions for GATE-Reranker (left) and MiniLM-L12-v2 (right). GATE exhibits a clear bimodal distribution with strong separation between relevant and non-relevant documents. In contrast, MiniLM collapses both classes into a narrow high-score region (zoomed), indicating severe miscalibration and near-zero discriminative margin.

lingual data, but often lack precise score separation, as shown in Section 6. In contrast, GATE-Reranker prioritizes reliable scoring behavior, producing well-calibrated outputs that are more suitable for real-world deployment scenarios.

7. Conclusion

We introduced **GATE-Reranker**, a compact Arabic cross-encoder trained with pointwise relevance regression on Arabic mMARCO-style triplets. Our results across binary, multi-negative, and large-scale Arabic benchmarks show that an Arabic-specialized reranker can achieve competitive effectiveness relative to strong multilingual baselines while maintaining substantially lower inference cost. In particular, GATE-Reranker-V1 provides a favorable effectiveness–efficiency trade-off, combining near state-of-the-art ranking quality with low GPU memory usage and latency. We additionally analyzed a higher-capacity scaling variant to quantify the accuracy–efficiency frontier within the same training paradigm. Our findings suggest that calibration-aware reranking is as important as ranking effectiveness for real-world deployment. Overall, this work establishes a strong and reproducible Arabic reranking baseline and highlights the importance of Arabic-focused evaluation and deployment-aware modeling in neural retrieval systems.

8. Limitations

GATE-Reranker provides strong discriminative power and well-calibrated scores, enabling reliable threshold-based filtering and robust behavior in downstream systems such as RAG, while maintaining a favorable effectiveness–efficiency trade-off. However, the study assumes a two-stage retrieval pipeline and does not evaluate end-to-end recall. The model is also limited by a 512-token context window, which may affect performance on long passages, and some evaluation benchmarks exhibit near-ceiling results, reducing their discriminative value. Additionally, the approach may inherit biases from mMARCO-style training data, including domain imbalance and limited dialectal coverage, as well as from the underlying Arabic embedding backbone, potentially affecting generalization across domains and linguistic variations.

9. References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96, New York, NY, USA. Association for Computing Machinery.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Hervé Déjean, Stéphane Clinchant, and Thibault Formal. 2024. A thorough comparison of cross-encoders and llms for reranking splade. *arXiv preprint arXiv:2403.10407*.

Yanhua Huang, Yuqi Chen, Xiong Cao, Rui Yang, Mingliang Qi, Yinghao Zhu, Qingchang Han, Yaowei Liu, Zhaoyu Liu, Xuefeng Yao, Yuting Jia, Leilei Ma, Yinqi Zhang, Taoyu Zhu, Liujie Zhang, Lei Chen, Weihang Chen, Min Zhu, Ruiwen Xu, and Lei Zhang. 2025. Towards large-scale generative ranking.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval.

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

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. *arXiv preprint arXiv:2505.24581*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *ArXiv*, abs/1901.04085.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Feng Wang, Yuqing Li, and Han Xiao. 2025. jina-reranker-v3: Last but not late interaction for listwise document reranking.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Yanzhao Zhang, M Li, D Long, X Zhang, H Lin, B Yang, P Xie, A Yang, D Liu, J Lin, et al. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.