

GATE: General Arabic Text Embedding for Enhanced Semantic Textual Similarity with Matryoshka Representation Learning and Hybrid Loss Training

Omer Nacar^{1*} Anis Koubaa² Serry Sibae¹
Yasser Al-Habashi¹ Adel Ammar¹ Wadii Boulila¹

¹Prince Sultan University, Riyadh, Saudi Arabia

²Alfaisal University, Riyadh, Saudi Arabia

{onajar, ssibae, yalhabashi, aammar, wboulila}@psu.edu.sa, akoubaa@alfaisal.edu.sa

*Corresponding author: onajar@psu.edu.sa

Abstract

Semantic textual similarity (STS) is a critical task in natural language processing (NLP), enabling applications in retrieval, clustering, and understanding semantic relationships between texts. However, research in this area for the Arabic language remains limited due to the lack of high-quality datasets and pre-trained models. This scarcity of resources has restricted the accurate evaluation and advance of semantic similarity in Arabic text. This paper introduces General Arabic Text Embedding (GATE) models that achieve state-of-the-art performance on the Semantic Textual Similarity task within the MTEB benchmark. GATE leverages Matryoshka Representation Learning and a hybrid loss training approach with Arabic triplet datasets for Natural Language Inference, which are essential for enhancing model performance in tasks that demand fine-grained semantic understanding. GATE outperforms larger models, including OpenAI, with a 20-25% performance improvement on STS benchmarks, effectively capturing the unique semantic nuances of Arabic.

1 Introduction

Text embeddings drive advances in clustering, information retrieval, and semantic similarity (Reimers, 2019; Gao et al., 2023; Asai et al., 2023; Gao et al., 2021). These models aim to map textual information into dense, low-dimensional vector representations that preserve nuanced semantic and contextual relationships. At the heart of many highly effective embedding models lies contrastive learning, a paradigm that optimizes the quality of representation by pulling semantically similar (positive) samples closer while pushing dissimilar (negative) samples apart (Gao et al., 2021; He et al., 2020; Radford et al., 2021).

Despite the versatility and success of contrastive learning, most existing text embedding pipelines rely on a two-stage pre-train-fine-tuning process:

weakly supervised large-scale pre-training followed by fine-tuning on high-quality text pairs acquired through data mining or manual annotation (Li et al., 2023; Wang et al., 2022; Xiao et al., 2023). Although effective, this approach often relies on the standard InfoNCE loss with in-batch negative samples (He et al., 2020), achieving robust representations predominantly by using large batch sizes and numerous negative samples. However, InfoNCE alone is not sufficient for all downstream tasks. In particular, sentence-level tasks such as Semantic Textual Similarity (STS) have been shown to benefit less from InfoNCE-based training, indicating a limitation in capturing fine-grained similarity cues (Huang et al., 2024). Likewise, key NLP tasks such as STS and classification have yet to be thoroughly integrated into general embedding training objectives.

Arabic presents specific linguistic challenges that complicate Semantic Textual Similarity (STS) tasks. Although Arabic is the fourth most used language on the Internet (Li and Yang, 2018) and the fifth most spoken language worldwide (Bourahouat et al., 2024), high-quality Arabic text embeddings are scarce. This scarcity exacerbates issues arising from Arabic’s rich morphological structure, characterized by a root-and-pattern system that generates a multitude of derivations, and its flexible syntax, where variable word orders can obscure semantic parallels. Additionally, the frequent omission of diacritics in written Arabic leads to significant ambiguity, as identical word forms may convey different meanings in context. These challenges collectively restrict the accurate capture of semantic nuances, making STS tasks particularly demanding for Arabic NLP applications.

This paper tackles these issues by introducing GATE, a General Arabic Text Embedding model designed to excel in semantic textual similarity and other downstream tasks. Our approach integrates Matryoshka Representation Learning

(MRL) (Kusupati et al., 2022) with a multitask hybrid loss training method. More specifically, we exploit various loss functions tailored to different task objectives—e.g., cosine similarity-based loss for STS and classification-oriented loss for downstream classification tasks. GATE improves semantic distinction by leveraging hard negative datasets and a flexible embedding structure. It addresses the limitations of single-loss approaches like InfoNCE-only training (Gutmann and Hyvärinen, 2010).

Our contributions are the following:

- **Hybrid Loss Strategy:** We propose a hybrid loss combining cosine similarity for semantic tasks and softmax-based classification, improving Arabic textual similarity beyond standard InfoNCE.
- **Enhanced Model Robustness:** We incorporate curated Arabic NLI triplet and labeled pair datasets, capturing nuanced semantic relationships crucial for downstream tasks.
- **Scalable Arabic Embeddings:** We adapt Matryoshka Representation Learning to Arabic, enabling efficient multi-dimensional embeddings (768, 512, 256, 128, and 64) with strong performance across tasks.

Our models, and data for reproducibility are publicly available [GATE Collection](#).

The paper is organized as follows. Section 2 reviews related work. Section 3 covers the proposed GATE framework, including our annotated datasets, Matryoshka embeddings training settings, along with a hybrid loss training approach. Section 4 presents the experimental results, evaluation, benchmarking, and error analysis.

2 Related Work

Semantic Textual Similarity (STS) (Cer et al., 2017) is a fundamental task in Natural Language Processing (NLP) that measures how closely two sentences align in meaning. Unlike binary classification tasks such as textual entailment or paraphrase detection, STS provides a graded measure of semantic equivalence (Zhao et al., 2024). It serves as a cornerstone for various NLP applications, including machine translation (Pathak et al., 2019), text summarization (Liu et al., 2022), and question answering (Wu et al., 2021), making it a crucial benchmark for evaluating embedding models.

Matryoshka Representation Learning (MRL) has emerged as an innovative approach to enhancing text embeddings by introducing hierarchical embedding representations, enabling models to capture multiple fidelity levels while optimizing computational efficiency (Kusupati et al., 2022). By dynamically encoding information across varying dimensions, MRL reduces storage requirements and computational overhead without compromising accuracy. Recent advancements, such as OpenAI’s text-embedding v3 (OpenAI, 2024), have demonstrated the effectiveness of MRL in semantic representation learning, influencing modern embedding architectures (Koenig et al., 2024; Lee et al., 2024; Infgrad, 2024).

Large language models (LLMs) have significantly advanced text embeddings, leveraging massive parameter spaces for complex semantic representations. Models such as *E5-Mistral-7B-Instruct* (Wang et al., 2023) and *Udever-Bloom-1B1* (Zhang et al., 2023) enhance generalization across domains but remain predominantly optimized for English. OpenAI’s third-generation embedding models (OpenAI, 2023) offer strong multilingual performance but are computationally expensive and lack adaptability for Arabic-specific tasks.

In Arabic NLP, models like *AraBERT* and *MARBERT* have improved language understanding by transitioning from masked language models (MLMs) to sentence embeddings (Reimers and Gurevych, 2019). While *AraBERT* focuses on formal Arabic (Antoun et al., 2020), *MARBERT* extends coverage to dialectal Arabic through large-scale pretraining (Abdul-Mageed et al., 2020).

Multilingual models such as *LaBSE*, *SBERT*, and *Multilingual E5* aim to bridge cross-lingual gaps, supporting over 100 languages. However, they struggle with fine-grained Arabic semantics, particularly in STS tasks (Wang et al., 2024).

To contextualize GATE’s advancements, Table 1 presents a comparative analysis of various text embedding models based on key features, including loss type, embedding dimensionality, fine-tuning methodology, and language specialization. As shown in Table 1, most existing models lack hybrid loss strategies, rely on fixed-dimensional embeddings, and are either multilingual or English-centric, making them suboptimal for fine-grained Arabic NLP tasks. GATE addresses these gaps by integrating hybrid loss training, leveraging Matryoshka embeddings, and fine-tuning Arabic se-

Work	Embedding Size	Primary Language Focus	Hybrid Loss	Multi-Dimensional	Semantic-Rich Fine-Tuning
OpenAI Text-Embedding v3 (OpenAI, 2023)	1536 / 3072	Multilingual	✗	✓	✗
E5-Mistral-7B-Instruct (Wang et al., 2023)	4096	English-Focused	✗	✗	✗
Udever-Bloom-1B1 (Zhang et al., 2023)	1536	Multilingual	✗	✗	✗
AraBERT (Antoun et al., 2020)	768	Arabic-Specific	✗	✗	✗
MARBERT (Abdul-Mageed et al., 2020)	768	Arabic-Specific	✗	✗	✗
LaBSE (Feng et al., 2020)	768	Multilingual	✗	✗	✗
Multilingual E5 (Wang et al., 2024)	384	Multilingual	✗	✗	✗
GATE Models (Proposed)	768, 512, 256, 128, 64	Semantic Arabic-Specific	✓	✓	✓

Table 1: Comparison of GATE with existing models by loss type, embedding dimensions, and training.

mantic datasets, setting a new benchmark for Arabic text embeddings.

3 GATE Framework

The GATE framework focuses on Matryoshka representation learning and a multi-task hybrid training approach to enhance Arabic text embeddings. Utilizing the Arabic versions of the Stanford Natural Language Inference (SNLI) and Multi Natural Language Inference (MultiNLI) datasets refines embeddings for optimal performance across various NLP tasks.

3.1 Dataset

Our study utilizes Arabic-adapted subsets derived from the Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and MultiNLI datasets (Kim et al., 2019), originally designed for natural language inference (NLI) (MacCartney and Manning, 2008) tasks. Table 2 summarizes the composition of the Arabic proposed dataset.

Subset	Columns	Training	Test
STS	text, text pair, score	8.63K	1.68K
Triplet	text, text triplet	571K	6.58K
Pair Classification	text, text pair, label	981K	19.7K

Table 2: Overview of datasets used in training and test.

As shown in Table 2, the primary datasets used in this study include the Triplet Subset (571K training, 6.58K test) for contrastive learning, the STS Subset (8.63K training, 1.68K test) for semantic textual similarity evaluation, and the Pair Classification Subset (981K training, 19.7K test) for entailment, neutral, and contradiction classification in hybrid loss training. To adapt NLI datasets for Arabic, we used Neural Machine Translation (NMT) (Klein et al., 2017) with CTranslate2, applying SentencePiece tokenization for efficient processing. Manual reviews ensured high translation accuracy.

3.2 Proposed Arabic Matryoshka Models

We introduce a diverse set of Matryoshka-based models optimized for Arabic semantic similarity

and natural language inference (NLI). These models enhance representation learning by leveraging hybrid loss training and Matryoshka loss, refining embeddings across different Arabic linguistic contexts.

At the core of our framework is *GATE-AraBERT-V1*, a multi-task trained Arabic embedding model fine-tuned on AllNLI and STS datasets. It is derived from *Arabic-Triplet-Matryoshka-V2*, which extends *AraBERT* using Matryoshka loss and triplet-based training, significantly improving Arabic sentence representations.

Other key models include *Arabic-all-nli-triplet-Matryoshka*, derived from *paraphrase-multilingual-mpnet-base-v2*, optimized for Arabic NLI through triplet learning. *Arabic-labse-Matryoshka* enhances LaBSE’s cross-lingual embeddings for Arabic, while *MARBERT-all-nli-triplet-Matryoshka* adapts MARBERT for both MSA and dialectal Arabic. Finally, *E5-all-nli-triplet-Matryoshka*, built upon multilingual-E5-small, serves as a comparative benchmark for triplet-based learning in Arabic.

Matryoshka models provide a cost-effective alternative to large-scale models like OpenAI’s embeddings, which face scalability and computational challenges. While larger models excel in multilingual tasks, they struggle with fine-grained Arabic semantics. By adapting Arabic and multilingual base models within the Matryoshka framework and leveraging triplet-based training, these models achieve enhanced semantic understanding, improving similarity and NLI tasks while maintaining a balance between cross-lingual adaptability and Arabic linguistic precision (Nacar and Koubaa, 2024).

3.2.1 Matryoshka Embedding Training Approach

Matryoshka Embedding Models (Kusupati et al., 2022) introduce an advanced technique for generating adaptable and multi-granular embeddings in natural language processing tasks. These models

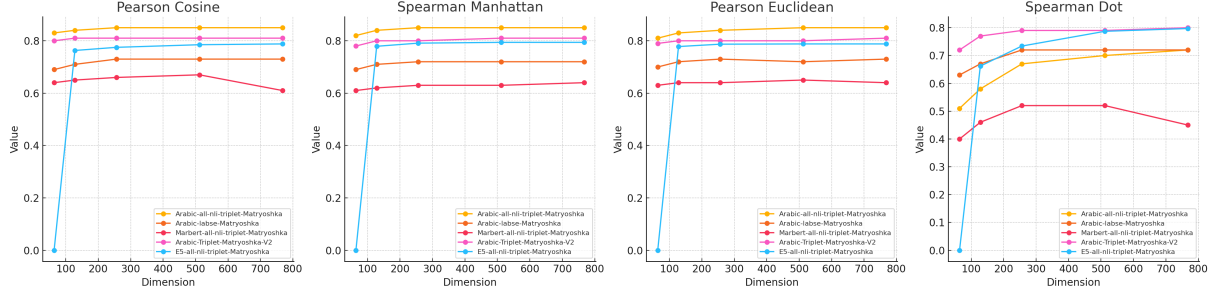


Figure 1: Results of Correlation-based Similarity Metrics on our proposed models.

are designed to capture varying levels of granularity within the embedding vectors, which allows for nuanced representation and efficient computational resource management. This is particularly beneficial in large-scale and resource-constrained scenarios, such as Arabic NLP.

MRL process involves generating a high-dimensional vector $z \in R^d$ for each data point x using a deep neural network $F(\cdot; \theta_F)$ parameterized by learnable weights θ_F . The key objective of MRL is to ensure that each subset of the first m dimensions of this vector, denoted $z_{1:m} \in R^m$ can independently represent the data point effectively. The granularity of the embeddings is controlled through a set of dimensions M , which are selected by progressively halving the vector size until reaching a minimal informative state. This approach guarantees that representations remain useful even when truncated to smaller dimensions.

Given a labeled dataset $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$ where $x_i \in \chi$ is an input point and $y_i \in [L]$ is its label, MRL optimizes the multi-class classification loss for each dimension subset $m \in M$. The overall optimization objective is expressed in equation 1:

$$\mathcal{L}_{\text{MRL}} = \sum_{m \in M} c_m \mathcal{L}_{\text{CE}}(\mathbf{W}^{(m)} \mathbf{z}_{1:m}, y) \quad (1)$$

where \mathcal{L}_{MRL} is the MRL loss. c_m represents the relative importance of each dimension m . \mathcal{L}_{CE} denotes the multi-class softmax cross-entropy loss function. $\mathbf{W}^{(m)} \in \mathbb{R}^{L \times m}$ are the weights of the linear classifier for dimension m . $\mathbf{z}_{1:m} \in \mathbb{R}^m$ is the truncated embedding vector up to dimension m . y is the true label corresponding to the input x .

To optimize memory usage, we implement weight-tying across all linear classifiers, setting $\mathbf{W}^{(m)} = \mathbf{W}_{1:m}$ for a set of common weights \mathbf{W} . This variant, known as Efficient MRL, helps man-

age the memory footprint, which is crucial for handling extensive output spaces.

For the training of Matryoshka models, we utilized the *arabic-nli-triplet* dataset, consisting of 558k triplets, and configured the models to use embeddings at varying dimensions [768, 512, 256, 128, 64]. The training involved using *MultipleNegativesRankingLoss* combined with *MatryoshkaLoss* to handle multiple dimensions effectively. Models were trained on an A100 GPU with a batch size of 128 and a maximum sequence length of 512 tokens. Training configurations and results are managed using the *SentenceTransformerTrainer*.

3.2.2 Hybrid Loss Training Approach

A multi-task hybrid loss method has been employed to address limitations in traditional training approaches for embedding models. The training process for our hybrid loss approach was implemented using a multi-dataset strategy that simultaneously leverages both classification and similarity-based objectives. To accommodate the distinct nature of the tasks, we defined two specialized loss functions. For the pair classification task, which involves labeling premise-hypothesis pairs into one of three classes (entailment, neutral, or contradiction), we use a SoftmaxLoss. This loss operates on the sentence embedding dimension extracted from our model and is parameterized by the number of labels (set to 3 in our case). For each premise x , its corresponding hypothesis y^+ with the correct label (entailment, contradiction, or neutral) is treated as a positive pair, while hypotheses with incorrect labels y^- are treated as negative pairs. The classification loss function is defined in equation 2:

$$L_{\text{cls}} = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{s(x_i, y^+)/\tau}}{e^{s(x_i, y^+)/\tau} + \sum_{j=1}^k e^{s(x_i, y_j^-)/\tau}} \quad (2)$$

where $s(x, y)$ denotes the similarity between the

premise x and the hypothesis y , and τ is the temperature scaling parameter. In this case, label-based negatives are applied rather than in-batch negatives.

For the STS task, which requires capturing subtle semantic differences between sentence pairs, we adopt a cosine similarity-based loss (CoSENT-Loss) that effectively penalizes deviations in the computed cosine similarity. The losses are mapped to their respective datasets in a dictionary, ensuring that the appropriate loss function is applied during each training iteration. The cosine similarity loss function is shown in equation 3:

$$L_{\text{sts}} = \log \left(1 + \sum_{s(x_i, x_j) > s(x_m, x_n)} \exp \left(\frac{\cos(x_m, x_n) - \cos(x_i, x_j)}{\tau} \right) \right) \quad (3)$$

where τ is the temperature scaling parameter, and $\cos(\cdot)$ represents the cosine similarity function. x_m, x_n, x_i, x_j are embeddings of the text pairs.

Training is carried out using a *SentenceTransformerTrainer* configured with meticulously tuned hyperparameters to ensure robust and efficient convergence. In our setup, the training is executed for five epochs with a per-device batch size of 64 and a learning rate of $2e-5$, complemented by a warmup ratio of 0.1 to gradually ramp up the learning rate at the onset of training. Frequent logging, evaluation, and checkpointing—executed every 200 steps—enable real-time monitoring and allow for prompt adjustments during training. The final multi-task loss function is formulated in equation 4:

$$L = \begin{cases} L_{\text{cls}} & \text{if the task is classification,} \\ L_{\text{sts}} & \text{if the task is STS.} \end{cases} \quad (4)$$

This hybrid loss approach ensures that our embedding models are optimally tuned for both classification and STS tasks, thereby enhancing their capability to capture the intricate semantic nuances of Arabic.

4 Results and Discussion

4.1 Results of Correlation Similarity Metrics

In order to assess the robustness of Matryoshka embeddings across different dimensions, we evaluated our Matryoshka models across multiple embedding sizes (768, 512, 256, 128, and 64). We employ correlation-based similarity metrics, commonly used in text embedding evaluations, to measure the consistency of embeddings across different dimensions. Figure 1 presents the results using

Pearson and Spearman correlation metrics, computed with different distance functions: Cosine, Manhattan, Euclidean, and Dot Product.

As shown in Figure 1, higher-dimensional embeddings (768, 512) consistently achieve superior performance, while lower-dimensional embeddings (128, 64) exhibit a noticeable decline, particularly in dot product-based similarity measures. *Arabic-all-nli-triplet-Matryoshka* achieves the highest scores across Pearson Cosine, Spearman Manhattan, and Pearson Euclidean, maintaining values around 0.85 for larger dimensions. *Arabic-Triplet-Matryoshka-V2* follows closely with stable performance across all metrics, scoring approximately 0.80 at higher dimensions. *Arabic-labse-Matryoshka* remains robust, averaging 0.72–0.73, while *Marbert-all-nli-triplet-Matryoshka* shows slightly lower results, particularly in Spearman Dot and Pearson Cosine (0.61–0.67). *E5-all-nli-triplet-Matryoshka* demonstrates a declining trend, especially in Spearman Dot at lower dimensions. These findings reinforce the trade-off between STS accuracy and embedding efficiency, emphasizing the importance of selecting an optimal embedding size based on computational constraints and task requirements.

4.2 Performance Evaluation on Arabic STS MTEB Benchmarks

To evaluate the effectiveness of Matryoshka and Multi-Task Hybrid Loss methods, we conduct experiments on GATE models, and their base counterparts using the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022) for Arabic. MTEB provides a large-scale evaluation across various NLP tasks, including Semantic Textual Similarity (STS), with key Arabic metrics: STS17, STS22, and STS22-v2 (Cer et al., 2017). These metrics assess STS on a scale from 0 to 5, focusing on Arabic-Arabic sentence pairs. Table 3 presents the comparative performance of Matryoshka embeddings against their base models.

As shown in Table 3, Matryoshka-based models consistently outperform their base counterparts. *Arabic-Triplet-Matryoshka-V2* achieves the highest performance (69.99 avg.), excelling in STS17 (85.31), while *GATE-AraBERT-V1* follows closely with 68.54. Interestingly, *GATE-AraBERT-V1*—which incorporates multi-task hybrid loss training—scores slightly lower than *Arabic-Triplet-Matryoshka-V2*, likely due to trade-offs in optimizing multiple objectives (STS and classification).

Model	Dim	# Params.	STS17	STS22	STS22-v2	Average
Arabic-Triplet-Matryoshka-V2	768	135M	85.31	60.7	63.96	69.99
GATE-AraBert-v1	768	135M	82.78	59.75	63.09	68.54
bert-base-arabertv02	768	135M	54.53	46.86	49.95	50.45
Marbert-all-nli-triplet-Matryoshka	768	163M	82.18	58.08	61.32	67.19
MARBERTv2	768	163M	60.98	49.92	53.75	54.88
Arabic-labse-Matryoshka	768	471M	82.46	57.25	60.58	66.76
LaBSE	768	471M	69.07	57.66	60.98	62.57
E5-all-nli-triplet-Matryoshka	384	278M	80.37	56.34	59.64	65.45
multilingual-e5-small	384	278M	74.62	58.13	61.4	64.72
Arabic-all-nli-triplet-Matryoshka	768	135M	82.4	51.38	54.45	62.74
paraphrase-multilingual-mpnet-base-v2	768	135M	79.09	52.18	55.37	62.21

Table 3: Performance comparison of Matryoshka models vs. their base counterparts on MTEB benchmarks.

While hybrid loss improves generalizability, Matryoshka loss preserves fine-grained sentence embedding alignment better, explaining this marginal gap.

Among other Matryoshka adaptations, *Marbert-all-nli-triplet-Matryoshka* scores 67.19, showcasing robust performance across STS22 and STS22-v2, while *Arabic-labse-Matryoshka* follows closely with 66.76. The *E5-all-nli-triplet-Matryoshka*, despite using a smaller 384-dimensional embedding space, maintains competitive results with 65.45, demonstrating an effective balance between efficiency and performance.

In contrast, base models significantly underperform, with *bert-base-arabertv02* achieving the lowest score at 50.45 and *paraphrase-multilingual-mpnet-base-v2* reaching 62.21. These findings underscore the effectiveness of Matryoshka Representation Learning (MRL) and hybrid loss strategies in refining Arabic embedding models, enhancing STS understanding, and optimizing performance across Arabic NLP benchmarks.

Loss	STS17	STS22	STS22-v2	Average
L_{CE}	54.53	46.86	49.95	50.45
L_{MRL}	85.31	60.7	63.96	69.99
$L_{sts} + L_{cls}$	82.78	59.75	63.09	68.54

Table 4: Effect of Matryoshka and hybrid loss functions on Arabic STS benchmarks

Table 4 highlights the impact of different loss functions on the best-performing models, *Arabic-Triplet-Matryoshka-V2* and *GATE-AraBERT-V1*, across the three Arabic STS benchmarks in MTEB. The results demonstrate the crucial role of loss selection in optimizing model performance for STS

tasks.

As shown in Table 4, the baseline cross-entropy loss L_{CE} yields the lowest average score of 50.45, reinforcing its limitations in learning high-quality embeddings for fine-grained STS. In contrast, *Arabic-Triplet-Matryoshka-V2*, trained with Matryoshka loss L_{MRL} , achieves the highest performance with an average of 69.99, significantly improving on STS17 equal to 85.31. Similarly, the hybrid loss approach ($L_{sts} + L_{cls}$), applied to *GATE-AraBERT-V1*, achieves a strong performance with an average of 68.54. While slightly lower than MRL, this result highlights the trade-off between generalization and fine-tuned similarity alignment. Hybrid loss optimizes embeddings for both STS and classification tasks, making it more versatile across different NLP applications.

Moreover, the effectiveness of MRL extends beyond performance gains. It enables models to retain their high-level semantic understanding even when embeddings are trained at progressively smaller dimensions, reducing computational and memory costs without significant degradation in performance. This characteristic is particularly beneficial in resource-constrained settings, where maintaining efficiency without sacrificing accuracy is critical. Table 5 shows the performance of the best-performing model, *Arabic-Triplet-Matryoshka-V2*, across various embedding dimensions (768, 512, 256, 128, and 64) on STS MTEB metrics.

As shown in Table 5, results demonstrate that the model maintains robust performance across all dimensions. At the full 768-dimensional embedding, the model achieves an average score of 69.99, with 85.31 on STS17. Even when reduced to 512 and 256 dimensions, the performance remains nearly

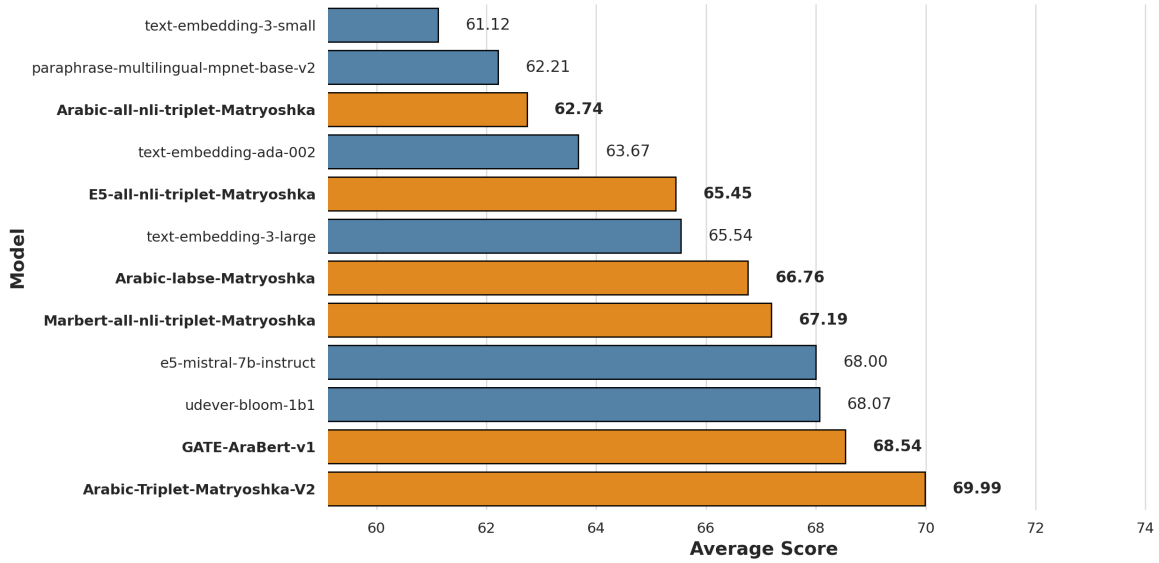


Figure 2: Performance comparison between Matryoshka models and larger models on MTEB Arabic benchmarks.

Evaluation Dim.	STS17	STS22	STS22-v2	Average
768	85.31	60.7	63.96	69.99
512	85.17	60.62	63.98	69.92
256	85.39	60.41	63.77	69.86
128	84.67	60.27	63.62	69.52
64	84.04	60.44	63.8	69.43

Table 5: Impact of embedding dimensions on the performance of *Arabic-Triplet-Matryoshka-V2*.

unchanged, with average scores of 69.92 and 69.86, respectively. Even at the lowest dimension of 64, the model still delivers a strong average score of 69.43, confirming that MRL allows for significant compression without substantial loss in accuracy.

4.3 Comparison of GATE Models with LLMs

To assess the efficiency of GATE models, we conducted a comparative evaluation against larger models, including *e5-mistral-7b-instruct* (7B parameters), *udever-bloom-1b1* (1B parameters), and OpenAI’s *text-embedding-3-small/large* and *text-embedding-ada-002*. Figure 2 highlights how Matryoshka models, despite smaller sizes, outperform or match billion-parameter LLMs in Arabic STS tasks.

As shown in Figure 2, the *Arabic-Triplet-Matryoshka-V2* model and *GATE-Arabert-V1*, with only 135M parameters, achieved the highest scores of 69.99 and 68.54 respectively, surpassing both *e5-mistral-7b-instruct* (68.00) and *udever-bloom-1b1* (68.07), despite their significantly larger parameter sizes. Similarly, OpenAI’s *text-embedding-*

ada-002 achieved a lower average score of 63.67, while the larger *text-embedding-3-large* model reached 65.54. Other Matryoshka models, such as *Marbert-all-nli-triplet-Matryoshka* and *Arabic-labse-Matryoshka*, demonstrated competitive performance, achieving 67.19 and 66.76, respectively. These results underscore the efficiency of the Matryoshka framework, demonstrating that smaller, well-optimized models can achieve state-of-the-art performance in STS tasks without the need for billions of parameters.

4.4 Error Analysis

We conducted an error analysis on Arabic-trained Matryoshka models by comparing their predictions against ground truth labels across high, moderate, and low similarity categories. This evaluation highlights patterns of overestimation and underestimation, particularly in distinguishing semantically unrelated pairs, as shown in Tables 7, 6, and 8.

As observed in the no similarity case in Table 6, most models assigned considerably higher similarity scores than the ground truth of 0.1, with some

exceeding 0.4, indicating a false positive bias. This suggests that while models effectively recognize shared words, they may struggle to distinguish true semantic relationships when there is lexical overlap. Notably, *GATE-AraBERT-V1* achieved the most accurate prediction with a score of 0.04, indicating that its hybrid loss training aids in learning better distinctions between semantically unrelated sentences.

For moderate similarity pairs in Table 7, models exhibit better alignment with ground truth, with scores ranging between 0.66 and 0.83, reinforcing their robustness in handling nuanced semantic relationships. *GATE-AraBERT-V1* slightly overestimates the similarity with a score of 0.81, while *Marbert-all-nli-triplet-Matryoshka* and *Arabic-labse-Matryoshka* reach the highest scores at 0.836 and 0.835, respectively.

For high similarity cases shown in Table 8, all models perform well, scoring above 0.84, closely mirroring the ground truth of 1.0. However, *GATE-AraBERT-V1* achieves a slightly lower score of 0.73, suggesting that hybrid loss training may introduce more conservative similarity estimations compared to Matryoshka loss models.

Model	Score	Sentence1	Sentence2
Ground Truth	0.1		
Arabic-all-nli-triplet-Matryoshka	0.48	رجل يعرف على الجيتار (A man playing the guitar)	رجل يقود سيارة (A man driving a car)
Arabic-Triplet-Matryoshka-V2	0.48		
GATE-AraBert-v1	0.04		
Arabic-labse-Matryoshka	0.32		
Marbert-all-nli-triplet-Matryoshka	0.38		

Table 6: Model scores for a no similarity sample.

Model	Score	Sentence1	Sentence2
Ground Truth	0.72		
Arabic-all-nli-triplet-Matryoshka	0.685	الرجال يلعبون كرة القدم (men are playing football)	الأولاد يلعبون كرة القدم (boys are playing football)
Arabic-Triplet-Matryoshka-V2	0.661		
GATE-AraBert-v1	0.81		
Arabic-labse-Matryoshka	0.835		
Marbert-all-nli-triplet-Matryoshka	0.836		

Table 7: Model scores for a moderate similarity sample.

Model	Score	Sentence1	Sentence2
Ground Truth	1		
Arabic-all-nli-triplet-Matryoshka	0.91	رجل يقوم بخدعة بالبطاقات (A man doing a card trick)	رجل يقوم بخدعة ورق (A man performing a card trick)
Arabic-Triplet-Matryoshka-V2	0.87		
Arabic-labse-Matryoshka	0.84		
Marbert-all-nli-triplet-Matryoshka	0.85		
GATE-AraBert-v1	0.73		

Table 8: Model scores for a high similarity sample.

5 Limitations

This work presents certain limitations. The lack of comprehensive Arabic NLP benchmarks restricts a

broader evaluation beyond STS tasks. Additionally, error analysis reveals a tendency to overestimate similarity in unrelated sentence pairs, often due to shared lexical elements, leading to false positives. Enhancing negative pair handling could further refine model accuracy. While our approach is optimized for Arabic, the methodology holds the potential for multilingual adaptation, expanding its applicability.

6 Conclusion

In this work, we introduced GATE, a General Arabic Text Embedding model leveraging MRL and hybrid loss training to enhance STS tasks. Evaluations on MTEB benchmarks confirmed strong performance retention across reduced dimensions and improved generalization over larger models. GATE fills key gaps in Arabic NLP by optimizing embeddings for fine-grained Arabic semantics. Future work will extend Arabic NLP benchmarks, diversify datasets, and explore multilingual generalization for broader real-world impact.

Acknowledgments

The authors thank Prince Sultan University for their support.

References

- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.
- Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46.
- Ghizlane Bourahouat, Manar Abourezq, and Najima Daoudi. 2024. Word embedding as a semantic feature extraction technique in arabic natural language processing: an overview. *Int. Arab J. Inf. Technol.*, 21(2):313–325.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

- Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
- Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.
- Junqin Huang, Zhongjie Hu, Zihao Jing, Mengya Gao, and Yichao Wu. 2024. Piccolo2: General text embedding with multi-task hybrid loss training. *arXiv preprint arXiv:2405.06932*.
- Infgrad. 2024. *Stella-mrl-large-zh-v3.5-1792d*. Accessed: 2024-08-28.
- Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6586–6593.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.
- Darius Koenig, Sean Lee, and Aamir Shakir. 2024. *Open source strikes bread - new fluffy embeddings model*. Accessed: 2024-08-28.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.
- Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. 2024. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*.
- Yang Li and Tao Yang. 2018. Word embedding for understanding natural language: a survey. *Guide to big data applications*, pages 83–104.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.
- Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen. 2022. Key phrase aware transformer for abstractive summarization. *Information Processing & Management*, 59(3):102913.
- Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528.
- Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.
- Omer Nacar and Anis Koubaa. 2024. Enhancing semantic similarity understanding in arabic nlp with nested embedding learning. *arXiv preprint arXiv:2407.21139*.
- OpenAI. 2023. Openai embeddings documentation. <https://platform.openai.com/docs/guides/embeddings>.
- OpenAI. 2024. *New embedding models and api updates*. Accessed: 2024-08-28.
- Amarnath Pathak, Partha Pakray, and Jereemi Bentham. 2019. English–mizo machine translation using neural and statistical approaches. *Neural Computing and Applications*, 31(11):7615–7631.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Nils Reimers and Iryna Gurevych. 2019. *Sentence-bert: Sentence embeddings using siamese bert-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.
- 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*.
- Yongliang Wu, Shuliang Zhao, and Ruiqiang Guo. 2021. A novel community answer matching approach based on phrase fusion heterogeneous information network. *Information Processing & Management*, 58(1):102408.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.
- Xin Zhang, Zehan Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. 2023. Language models are universal embedders. *arXiv preprint arXiv:2310.08232*.
- Ying Zhao, Tingyu Xia, Yunqi Jiang, and Yuan Tian. 2024. Enhancing inter-sentence attention for semantic textual similarity. *Information Processing & Management*, 61(1):103535.