Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models

Yanzhao Zhang* Mingxin Li* Dingkun Long* Xin Zhang*
Huan Lin Baosong Yang Pengjun Xie An Yang
Dayiheng Liu Junyang Lin Fei Huang Jingren Zhou
Tongyi Lab Alibaba Group

https://huggingface.co/Qwen
https://modelscope.cn/organization/qwen
https://github.com/QwenLM/Qwen3-Embedding

Abstract

In this work, we introduce the Qwen3 Embedding series, a significant advancement over its predecessor, the GTE-Qwen series, in text embedding and reranking capabilities, built upon the Qwen3 foundation models. Leveraging the Qwen3 LLMs' robust capabilities in multilingual text understanding and generation, our innovative multi-stage training pipeline combines large-scale unsupervised pre-training with supervised fine-tuning on high-quality datasets. Effective model merging strategies further ensure the robustness and adaptability of the Qwen3 Embedding series. During the training process, the Qwen3 LLMs serve not only as backbone models but also play a crucial role in synthesizing high-quality, rich, and diverse training data across multiple domains and languages, thus enhancing the training pipeline. The Qwen3 Embedding series offers a spectrum of model sizes (0.6B, 4B, 8B) for both embedding and reranking tasks, addressing diverse deployment scenarios where users can optimize for either efficiency or effectiveness. Empirical evaluations demonstrate that the Qwen3 Embedding series achieves state-of-the-art results across diverse benchmarks. Notably, it excels on the multilingual evaluation benchmark MTEB for text embedding, as well as in various retrieval tasks, including code retrieval, cross-lingual retrieval and multilingual retrieval. To facilitate reproducibility and promote community-driven research and development, the Qwen3 Embedding models are publicly available under the Apache 2.0 license.

1 Introduction

Text embedding and reranking are fundamental components in numerous natural language processing and information retrieval applications, including web search, question answering, recommendation systems, and beyond (Karpukhin et al., 2020; Huang et al., 2020; Zhao et al., 2023; 2024). High-quality embeddings enable models to capture semantic relationships between texts, while effective reranking mechanisms ensure that the most relevant results are prioritized. Recently, emerging application paradigms such as Retrieval-Augmented Generation (RAG) and agent systems, driven by the advancement of large language models (e.g., Qwen3 (Yang et al., 2025), GPT-4o (Hurst et al., 2024)), have introduced new requirements and challenges for text embedding and reranking, both in terms of model training paradigms and application scenarios. Despite significant advancements, training embedding and reranking models that perform well in scalability, contextual understanding, and alignment with specific downstream tasks remains challenging.

The emergence of large language models (LLMs) has significantly advanced the development of text embedding and reranking models. Prior to the introduction of LLMs, the predominant approach

* Equal contribution

involved using encoder-only pretrained language models like BERT as the foundational model for training (Reimers & Gurevych, 2019). The richer world knowledge, text understanding, and reasoning abilities inherent in LLMs have led to further enhancements in models trained on these architectures. Additionally, there has been considerable research facilitating the integration of LLMs into processes such as training data synthesis and quality data filtering (Wang et al., 2024; Lee et al., 2024; 2025b). The fundamental characteristics of LLMs have also inspired the introduction of new training paradigms. For instance, during the embedding model training process, incorporating differentiated tasks across aspects such as instruction type, domain, and language has yielded improved performance in downstream tasks (Su et al., 2023). Similarly, for reranking model training, advancements have been realized through both zero-shot methods based on user prompts and approaches combining supervised fine-tuning (Ma et al., 2023; Pradeep et al., 2023; Zhang et al., 2024a; Zhuang et al., 2024).

In this work, we introduce the Qwen3 Embedding series models, which are constructed on top of the Qwen3 foundation models. The Qwen3 foundation has simultaneously released base and instruct model versions, and we exploit the robust multilingual text understanding and generation capabilities of these models to fully realize their potential in training embedding and reranking models. To train the embedding models, we implement a multi-stage training pipeline that involves large-scale unsupervised pre-training followed by supervised fine tuning on high-quality datasets. We also employ model merging with various model checkpoints to enhance robustness and generalization. The Qwen3 instruct model allows for efficient synthesis of a vast, high-quality, multilingual, and multi-task text relevance dataset. This synthetic data is utilized in the initial unsupervised training stage, while a subset of high-quality, small-scale data is selected for the second stage of supervised training. For the reranking models, we adopt a two-stage training scheme in a similar manner, consisting of high-quality supervised fine tuning and a model merging stage. Based on different sizes of the Qwen3 backbone models (including 0.6B, 4B, and 8B), we ultimately trained three text embedding models and three text reranking models. To facilitate their application in downstream tasks, the Qwen3 Embedding series supports several practical features, such as flexible dimension representation for embedding models and customizable instructions for both embedding and reranking models.

We evaluate the Qwen3 Embedding series across a comprehensive set of benchmarks spanning multiple tasks and domains. Experimental results demonstrate that our embedding and reranking models achieve state-of-the-art performance, performing competitively against leading proprietary models in several retrieval tasks. For example, the flagship model Qwen3-8B-Embedding attains a score of 70.58 on the MTEB Multilingual benchmark (Enevoldsen et al., 2025) and 80.68 on the MTEB Code benchmark (Enevoldsen et al., 2025), surpassing the previous state-of-the-art proprietary embedding model, Gemini-Embedding (Lee et al., 2025b). Moreover, our reranking model delivers competitive results across a range of retrieval tasks. The Qwen3-Reranker-0.6B model exceeds previously top-performing models in numerous retrieval tasks, while the larger Qwen3-Reranker-8B model demonstrates even superior performance, improving ranking results by 3.0 points over the 0.6B model across multiple tasks. Furthermore, we include a constructive ablation study to elucidate the key factors contributing to the superior performance of the Qwen3 Embedding series, providing insights into its effectiveness.

In the following sections, we describe the design of the model architecture, detail the training procedures, present the experimental results for both the embedding and reranking models of the Qwen3 Embedding Series, and conclude this technical report by summarizing the key findings and outlining potential directions for future research.

2 Model Architecture

The core idea behind embedding and reranking models is to evaluate relevance in a task-aware manner. Given a query q and a document d, embedding and reranking models assess their relevance based on a similarity criterion defined by instruction I. To enable the models for task-aware relevance estimation, training data is often organized as $\{I_i, q_i, d_i^+, d_{i,1}^-, \cdots, d_{i,n}^-\}$, where d_i^+ represents a

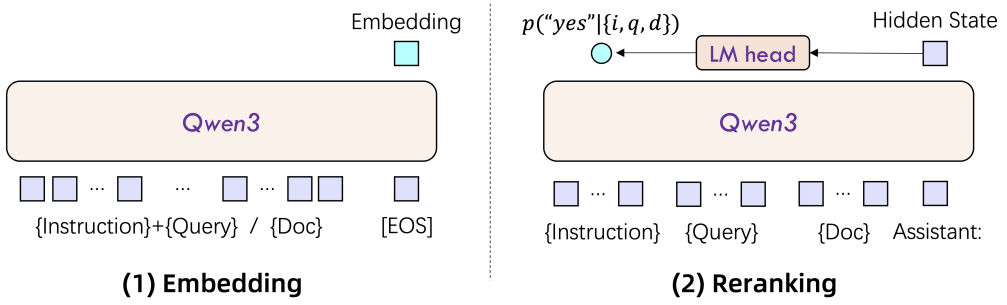

Figure 1: Model architecture of Qwen3-Embedding (left) and Qwen3-Reranker (right).

positive (relevant) document for query q_i, and $d_{i,j}^-$ are negative (irrelevant) documents. Training the model on diverse text pairs broadens its applicability to a range of downstream tasks, including retrieval, semantic textual similarity, classification, and clustering.

Architecture The Qwen3 embedding and reranking models are built on the dense version of Qwen3 foundation models and are available in three sizes: 0.6B, 4B, and 8B parameters. We initialize these models using the Qwen3 foundation models to leverage their capabilities in text modeling and instruction following. The model layers, hidden size, and context length for each model configuration are detailed in Table 1.

Embedding Models For text embeddings, we utilize LLMs with causal attention, appending an `[EOS]` token at the end of the input sequence. The final embedding is derived from the hidden state of the last layer corresponding to this `[EOS]` token.

To ensure embeddings follow instructions during downstream tasks, we concatenate the instruction and the query into a single input context, while leaving the document unchanged before processing with LLMs. The input format for queries is as follows:

```
{Instruction} {Query}<|endoftext|>
```

Reranking Models To more accurately evaluate text similarity, we employ LLMs for point-wise reranking within a single context. Similar to the embedding model, to enable instruction-following capability, we include the instruction in the input context. We use the LLM chat template and frame the similarity assessment task as a binary classification problem. The input to LLMs adheres to the template shown below:

```
<|im_start|>system
Judge whether the Document meets the requirements based on the Query and the
↪  Instruct provided. Note that the answer can only be "yes" or
↪  "no".<|im_end|>
<|im_start|>user
<Instruct>: {Instruction}
<Query>: {Query}
<Document>: {Document}<|im_end|>
<|im_start|>assistant
<think>\n\n</think>\n\n
```

Model Type	Models	Size	Layers	Sequence Length	Embedding Dimension	MRL Support	Instruction Aware
Text Embedding	Qwen3-Embedding-0.6B	0.6B	28	32K	1024	Yes	Yes
	Qwen3-Embedding-4B	4B	36	32K	2560	Yes	Yes
	Qwen3-Embedding-8B	8B	36	32K	4096	Yes	Yes
Text Reranking	Qwen3-Reranker-0.6B	0.6B	28	32K	-	-	Yes
	Qwen3-Reranker-4B	4B	36	32K	-	-	Yes
	Qwen3-Reranker-8B	8B	36	32K	-	-	Yes

Table 1: Model architecture of Qwen3 Embedding models. "MRL Support" indicates whether the embedding model supports custom dimensions for the final embedding. "Instruction Aware" notes whether the embedding or reranker model supports customizing the input instruction according to different tasks.

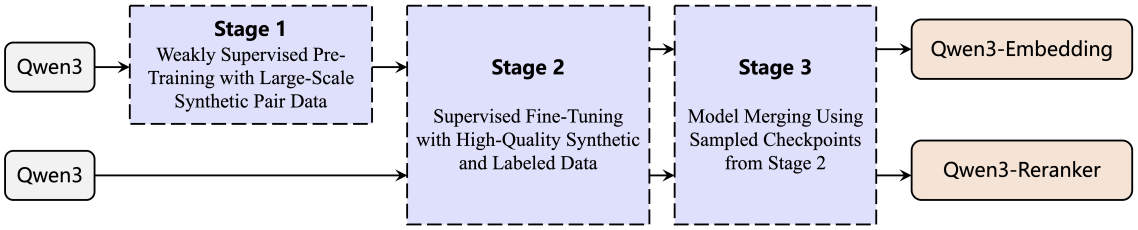

Figure 2: Training pipeline of Qwen3 Embedding and Reranking models.

To calculate the relevance score based on the given input, we assess the likelihood of the next token being "yes" or "no." This is expressed mathematically as follows:

$$\text{score}(q,d) = \frac{e^{P(\text{yes}|I,q,d)}}{e^{P(\text{yes}|I,q,d)} + e^{P(\text{no}|I,q,d)}}$$

3 Models Training

In this section, we describe the multi-stage training pipeline adopted and present the key elements of this training recipe, including training objective, training data synthesis, and filtering of high-quality training data.

3.1 Training Objective

Before introducing our training pipeline, we first outline the optimized loss functions used for the embedding and reranking models during the training process. For the embedding model, we utilize an improved contrastive loss based on the InfoNCE framework (Oord et al., 2018). Given a batch of N training instances, the loss is defined as:

$$L_{\text{embedding}} = -\frac{1}{N}\sum_{i}^{N} \log \frac{e^{(s(q_i,d_i^+)/\tau)}}{Z_i}, \tag{1}$$

where $s(\cdot,\cdot)$ is a similarity function (we use cosine similarity), τ is a temperature parameter, and Z_i is the normalization factor that aggregates the similarity scores of the positive pair against various negative pairs:

$$Z_i = e^{(s(q_i,d_i^+)/\tau)} + \sum_{k}^{K} m_{ik} e^{(s(q_i,d_{i,k}^-)/\tau)} + \sum_{j\neq i} m_{ij} e^{(s(q_i,q_j)/\tau)} + \sum_{j\neq i} m_{ij} e^{(s(d_i^+,d_j)/\tau)} + \sum_{j\neq i} m_{ij} e^{(s(q_i,d_j)/\tau)}$$

where these terms represent similarities with: (1) the positive document d_i^+, (2) K hard negatives $d_{i,k}^-$, (3) other in-batch queries q_j, (4) other in-batch documents d_j compared against the positive document d_i^+. (5) other in-batch documents d_j compared against the query q_i. The mask factor m_{ij} is designed to mitigate the impact of false negatives and is defined as:

$$m_{ij} = \begin{cases} 0 & \text{if } s_{ij} > s(q_i, d_i^+) + 0.1 \text{ or } d_j == d_i^+, \\ 1 & \text{otherwise,} \end{cases}$$

among which s_{ij} is the corresponding score of q_i, d_j or q_i, q_j.

For the reranking model, we optimize the Supervised Fine-Tuning (SFT) loss defined as:

$$L_{\text{reranking}} = -\log p(l|\mathcal{P}(q,d)), \tag{2}$$

where $p(\cdot|*)$ denotes the probability assigned by LLM. The label l is "yes" for positive documents and "no" for negatives. This loss function encourages the model to assign higher probabilities to correct labels, thereby improving the ranking performance.

3.2 Multi-stage Training

The multi-stage training approach is a common practice for training text embedding models (Li et al., 2023; Wang et al., 2022; Chen et al., 2024). This strategy typically begins with initial training on large-scale, semi-supervised data that includes noise, followed by fine-tuning using smaller, high-quality supervised datasets. This two-step process enhances the performance and generalization capabilities of embedding models. Large-scale weakly supervised training data contribute significantly to the model's generalization, while fine-tuning with high-quality data in subsequent stages further improves model performance. Both stages of training for embedding models utilize the optimization objective defined in Equation 1, whereas the reranking model training employs the loss function defined in Equation 2 as the optimization target.

Building upon the existing multi-stage training framework, the Qwen3 Embedding series introduces the following key innovations:

- Large-Scale Synthetic Data-Driven Weak Supervision Training: Unlike previous works (e.g., GTE, E5, BGE models), where weakly supervised training data are primarily collected from open-source communities such as Q&A forums or academic papers, we propose leveraging the text understanding and generation capabilities of foundation models to synthesize pair data directly. This approach allows for arbitrary definition of various dimensions of the desired pair data, such as task, language, length, and difficulty within the synthesis prompts. Compared to data collection from open-domain sources, foundation model-driven data synthesis offers greater controllability, enabling precise management of the quality and diversity of the generated data, particularly in low-resource scenarios and languages.
- High-Quality Synthetic Data Utilization in Supervised Fine Tuning: Due to the exceptional performance of the Qwen3 Foundation model, the synthesized data is of notably high quality. Therefore, in the second stage of supervised training, selective incorporation of this high-quality synthetic data further enhances the overall model performance and generalization capabilities.
- Model Merging: Inspired by previous work (Li et al., 2024), after completing the supervised fine-tuning, we applied a model merging technique based on spherical linear interpolation (slerp). This technique involves merging multiple model checkpoints saved during the fine-tuning process. This step aims to boost the model's robustness and generalization performance across various data distributions.

It is important to note that the reranking model's training process does not include a first-stage weakly supervised training phase.

3.3 Synthetic Dataset

To create a robust synthetic dataset for training models on various similarity tasks, we generate diverse text pairs spanning categories such as retrieval, bitext mining, classification, and semantic textual similarity (STS). The quality of these synthetic data pairs is ensured by utilizing the Qwen3-32B model as the foundational model for data synthesis. We have designed a diverse prompting strategy to improve the variety and authenticity of the generated data. For instance, in the text retrieval task, we synthesize data using the multilingual pre-training corpus from Qwen3. During the data synthesis process, specific roles are assigned to each document to simulate potential users querying that document. This injection of user perspectives enhances the diversity and realism of the synthetic queries. Specifically, we utilize a retrieval model to identify the top five role candidates for each document from a role library and present these documents along with their role candidates to the prompt. This guides the model in outputting the most suitable role configuration for query generation. Moreover, the prompt incorporates various dimensions such as query type (e.g., keyword, factual, summary, judgment), query length, difficulty, and language. This multidimensional approach ensures the quality and diversity of the synthetic data.

Finally, we create a total of approximately 150 million pairs of multi-task weak supervision training data. Our experiments reveal that the embedding model trained with these synthetic data performs exceptionally well in downstream evaluations, particularly surpassing many previously supervised models in the MTEB Multilingual benchmarks. This motivates us to filter the synthetic data to identify high-quality pairs for inclusion in a second stage of supervised training. We employ a simple cosine similarity calculation to select data pairs, retaining those with a cosine similarity greater than 0.7 from randomly sampled data. Ultimately, approximately 12 million high-quality supervised training data pairs are selected for further training.

Model	**Size**	**Mean (Task)**	**Mean (Type)**	Bitext Mining	Class-ification	Clus-tering	Inst. Retrieval	Multilabel Class.	Pair Class.	Rerank	Retrieval	STS
Selected Open-Source Models												
NV-Embed-v2	7B	56.29	49.58	57.84	57.29	40.80	1.04	18.63	78.94	63.82	56.72	71.10
GritLM-7B	7B	60.92	53.74	70.53	61.83	49.75	3.45	22.77	79.94	63.78	58.31	73.33
BGE-M3	0.6B	59.56	52.18	79.11	60.35	40.88	-3.11	20.1	80.76	62.79	54.60	74.12
multilingual-e5-large-instruct	0.6B	63.22	55.08	80.13	64.94	50.75	-0.40	22.91	80.86	62.61	57.12	76.81
gte-Qwen2-1.5B-instruct	1.5B	59.45	52.69	62.51	58.32	52.05	0.74	24.02	81.58	62.58	60.78	71.61
gte-Qwen2-7b-Instruct	7B	62.51	55.93	73.92	61.55	52.77	4.94	25.48	85.13	65.55	60.08	73.98
Commercial APIs												
text-embedding-3-large	-	58.93	51.41	62.17	60.27	46.89	-2.68	22.03	79.17	63.89	59.27	71.68
Cohere-embed-multilingual-v3.0	-	61.12	53.23	70.50	62.95	46.89	-1.89	22.74	79.88	64.07	59.16	74.80
Gemini Embedding	-	68.37	59.59	79.28	71.82	54.59	5.18	**29.16**	83.63	65.58	67.71	79.40
Qwen3 Embedding Models												
Qwen3-Embedding-0.6B	0.6B	64.33	56.00	72.22	66.83	52.33	5.09	24.59	80.83	61.41	64.64	76.17
Qwen3-Embedding-4B	4B	69.45	60.86	79.36	72.33	57.15	**11.56**	26.77	85.05	65.08	69.60	80.86
Qwen3-Embedding-8B	8B	**70.58**	**61.69**	**80.89**	**74.00**	**57.65**	10.06	28.66	**86.40**	**65.63**	**70.88**	**81.08**

Table 2: Performance on MTEB Multilingual (Enevoldsen et al., 2025). For compared models, the scores are retrieved from MTEB online leaderboard on June 4th, 2025.

4 Evaluation

We conduct comprehensive and fair evaluations across multiple benchmarks to assess the capabilities of Qwen3 Embedding models.

4.1 Settings

For the text embedding models, we utilize the Massive Multilingual Text Embedding Benchmark (MMTEB) (Enevoldsen et al., 2025) for evaluation. MMTEB is a large-scale, community-driven expansion of MTEB (Muennighoff et al., 2023), covering over 500 quality-controlled evaluation tasks

Model	Size	Dim	MTEB (Eng, v2)		CMTEB		MTEB (Code)
			Mean (Task)	Mean (Type)	Mean (Task)	Mean (Type)	
Selected Open-Source Models							
NV-Embed-v2	7B	4096	69.81	65.00	63.0	62.0	-
GritLM-7B	7B	4096	67.07	63.22	-	-	73.6[α]
multilingual-e5-large-instruct	0.6B	1024	65.53	61.21	-	-	65.0[α]
gte-Qwen2-1.5b-instruct	1.5B	1536	67.20	63.26	67.12	67.79	-
gte-Qwen2-7b-instruct	7B	3584	70.72	65.77	71.62	72.19	56.41[γ]
Commercial APIs							
text-embedding-3-large	-	3072	66.43	62.15	-	-	58.95[γ]
cohere-embed-multilingual-v3.0	-	1024	66.01	61.43	-	-	51.94[γ]
Gemini Embedding	-	3072	73.30	67.67	-	-	74.66[γ]
Qwen3 Embedding Models							
Qwen3-Embedding-0.6B	0.6B	1024	70.70	64.88	66.33	67.44	75.41
Qwen3-Embedding-4B	4B	2560	74.60	68.09	72.26	73.50	80.06
Qwen3-Embedding-8B	8B	4096	**75.22**	**68.70**	**73.83**	**75.00**	**80.68**

Table 3: Performance on MTEB Engilish, MTEB Chinese, MTEB Code. [α]Taken from (Enevoldsen et al., 2025). [γ]Taken from (Lee et al., 2025b). For other compared models, the scores are retrieved from MTEB online leaderboard on June 4th, 2025.

across more than 250 languages. In addition to classic text tasks such as as a variety of retrieval, classification, and semantic textual similarity, MMTEB includes a diverse set of challenging and novel tasks, such as instruction following, long-document retrieval, and code retrieval, representing the largest multilingual collection of evaluation tasks for embedding models to date. Our MMTEB evaluations encompass 216 individual evaluation tasks, consisting of 131 tasks for MTEB (Multilingual) (Enevoldsen et al., 2025), 41 tasks for MTEB (English, v2) (Muennighoff et al., 2023), 32 tasks for CMTEB (Xiao et al., 2024), and 12 code retrieval tasks for MTEB (Code) (Enevoldsen et al., 2025).

Moreover, we select a series of text retrieval tasks to assess the text reranking capabilities of our models. We explore three types of retrieval tasks: (1) Basic Relevance Retrieval, categorized into English, Chinese, and Multilingual, evaluated on MTEB (Muennighoff et al., 2023), CMTEB (Xiao et al., 2024), MMTEB (Enevoldsen et al., 2025), and MLDR (Chen et al., 2024), respectively; (2) Code Retrieval, evaluated on MTEB-Code (Enevoldsen et al., 2025), which comprises only code-related retrieval data.; and (3) Complex Instruction Retrieval, evaluated on FollowIR (Weller et al., 2024).

Compared Methods We compare our models with the most prominent open-source text embedding models and commercial API services. The open-source models include the GTE (Li et al., 2023; Zhang et al., 2024b), E5 (Wang et al., 2022), and BGE (Xiao et al., 2024) series, as well as NV-Embed-v2 (Lee et al., 2025a), GritLM-7B Muennighoff et al. (2025). The commercial APIs evaluated are text-embedding-3-large from OpenAI, Gemini-embedding from Google, and Cohere-embed-multilingual-v3.0. For reranking, we compare with the rerankers of jina[1], mGTE (Zhang et al., 2024b) and BGE-m3 (Chen et al., 2024).

4.2 Main Results

Embedding In Table 2, we present the evaluation results on MMTEB (Enevoldsen et al., 2025), which comprehensively covers a wide range of embedding tasks across multiple languages. Our Qwen3-Embedding-4B/8B models achieve the best performance, and our smallest model, Qwen3-Embedding-0.6B, only lags behind the best-performing baseline method (Gemini-Embedding), despite having only 0.6B parameters. In Table 3, we present the evaluation results on MTEB (English, v2) (Muennighoff et al., 2023), CMTEB (Xiao et al., 2024), and MTEB (Code) (Enevoldsen et al., 2025). The scores reflect similar trends as MMTEB, with our Qwen3-Embedding-4B/8B models

[1] https://hf.co/jinaai/jina-reranker-v2-base-multilingual

Model	Param	Basic Relevance Retrieval					
		MTEB-R	CMTEB-R	MMTEB-R	MLDR	MTEB-Code	FollowIR
Qwen3-Embedding-0.6B	0.6B	61.82	71.02	64.64	50.26	75.41	5.09
Jina-multilingual-reranker-v2-base	0.3B	58.22	63.37	63.73	39.66	58.98	-0.68
gte-multilingual-reranker-base	0.3B	59.51	74.08	59.44	66.33	54.18	-1.64
BGE-reranker-v2-m3	0.6B	57.03	72.16	58.36	59.51	41.38	-0.01
Qwen3-Reranker-0.6B	0.6B	65.80	71.31	66.36	67.28	73.42	5.41
Qwen3-Reranker-4B	4B	**69.76**	75.94	72.74	69.97	81.20	**14.84**
Qwen3-Reranker-8B	8B	69.02	**77.45**	**72.94**	**70.19**	**81.22**	8.05

Table 4: Evaluation results for reranking models. We use the retrieval subsets of MTEB(eng, v2), MTEB(cmn, v1) and MMTEB, which are MTEB-R, CMTEB-R and MMTEM-R. The rest are all retrieval tasks. All scores are our runs based on the retrieval top-100 results from the first row.

Model	MMTEB	MTEB (Eng, v2)	CMTEB	MTEB (Code, v1)
Qwen3-Embedding-0.6B w/ only synthetic data	58.49	60.63	59.78	66.79
Qwen3-Embedding-0.6B w/o synthetic data	61.21	65.59	63.37	74.58
Qwen3-Embedding-0.6B w/o model merge	62.56	68.18	64.76	74.89
Qwen3-Embedding-0.6B	64.33	70.70	66.33	75.41

Table 5: Performance (mean task) on MMTEB, MTEB(eng, v2), CMTEB and MTEB(code, v1) for Qwen3-Embedding-0.6B model with different training setting.

consistently outperforming others. Notably, the Qwen3-Embedding-0.6B model ranks just behind the Gemini-Embedding, while being competitive with the gte-Qwen2-7B-instruct.

Reranking In Table 4, we present the evaluation results on various reranking tasks (§4.1). We utilize the Qwen3-Embedding-0.6B model to retrieve the top-100 candidates and then apply different reranking models for further refinement. This approach ensures a fair evaluation of the reranking models. Our results indicate that all three Qwen3-Reranker models enhance performance compared to the embedding model and surpass all baseline reranking methods, with Qwen3-Reranker-8B achieving the highest performance across most tasks.

4.3 Analysis

To further analyze and explore the key elements of the Qwen3 Embedding model training framework, we conduct an analysis from the following dimensions:

Effectiveness of Large-Scale Weakly Supervised Pre-Training We first analyze the effectiveness of the large-scale weak supervised training stage for the embedding models. As shown in Table 5, the Qwen3-Embedding-0.6B model trained solely on synthetic data (without subsequent training stages, as indicated in the first row) achieves reasonable and strong performance compared to the final Qwen3-Embedding-0.6B model (as shown in the last row). If we further remove the weak supervised training stage (i.e., without synthetic data training, as seen in the second row), the final performance shows a clear decline. This indicates that the large-scale weak supervised training stage is crucial for achieving superior performance.

Effectiveness of Model Merging Next, we compare the performance differences arising from the model merging stage. As shown in Table 5, the model trained without model merging techniques (the third row, which uses data sampling to balance various tasks) performs considerably worse than the final Qwen3-Embedding-0.6B model (which employs model merging, as shown in the last row). This indicates that the model merging stage is also critical for developing strong models.

5 Conclusion

In this technical report, we present the Qwen3-Embedding series, a comprehensive suite of text embedding and reranking models based on the Qwen3 foundation models. These models are designed to excel in a wide range of text embedding and reranking tasks, including multilingual retrieval, code retrieval, and complex instruction following. The Qwen3-Embedding models are built upon a robust multi-stage training pipeline that combines large-scale weakly supervised pre-training on synthetic data with supervised fine-tuning and model merging on high-quality datasets. The Qwen3 LLMs play a crucial role in synthesizing diverse training data across multiple languages and tasks, thereby enhancing the models' capabilities. Our comprehensive evaluations demonstrate that the Qwen3-Embedding models achieve state-of-the-art performance across various benchmarks, including MTEB, CMTEB, MMTEB, and several retrieval benchmarks. We are pleased to open-source the Qwen3-Embedding and Qwen3-Reranker models (0.6B, 4B, and 8B), making them available for the community to use and build upon.

References

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.137/.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, et al. MMTEB: Massive multilingual text embedding benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zl3pfz4VCV.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2553–2561, 2020.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pp. 6769–6781, 2020.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=lgsyLSsDRe.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*, 2025b.

Mingxin Li, Zhijie Nie, Yanzhao Zhang, Dingkun Long, Richong Zhang, and Pengjun Xie. Improving general text embedding model: Tackling task conflict and data imbalance through model merging. *arXiv preprint arXiv:2410.15035*, 2024.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023. URL https://arxiv.org/abs/2308.03281.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*, 2023.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.eacl-main.148/.

Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BC4lIvfSzv.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. URL https://aclanthology.org/D19-1410/.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1102–1121, 2023.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022. URL https://arxiv.org/abs/2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11897–11916, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.642/.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. Followir: Evaluating and teaching information retrieval models to follow instructions. *arXiv preprint arXiv:2403.15246*, 2024.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 641–649, New York, NY, USA, 2024. Association for Computing Machinery. URL https://doi.org/10.1145/3626772.3657878.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Longhui Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. A two-stage adaptation of large language models for text ranking. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 11880–11891, 2024a.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1393–1412, Miami, Florida, US, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.103. URL https://aclanthology.org/2024.emnlp-industry.103/.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60, 2024.

Xiangyu Zhao, Maolin Wang, Xinjian Zhao, Jiansheng Li, Shucheng Zhou, Dawei Yin, Qing Li, Jiliang Tang, and Ruocheng Guo. Embedding in recommender systems: A survey. *arXiv preprint arXiv:2310.18608*, 2023.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 38–47, 2024.

A Appendix

A.1 Synthetic Data

We construct four types of synthetic data—retrieval, bitext mining, semantic textual similarity, and classification to enable the model to adapt to various similarity tasks during pre-training. To ensure both multilingual and cross-lingual diversity, the data is generated using Qwen3 32B. Below is an example of a synthetic retrieval text pair. The retrieval data is synthesized using a document-to-query approach. We collect a multilingual corpus from the pre-training corpus of the Qwen3 base model to serve as the document source. A two-stage generation pipeline is then applied, consisting of: (1) configuration and (2) query generation. In the configuration stage, we use large language models (LLMs) to determine the "Question Type", "Difficulty", and "Character" for the synthetic query. The candidate characters are retrieved from Persona Hub (Ge et al., 2024), selecting the top five most relevant to the given document. This step aims to enhance the diversity of the generated queries. The template used is as follows:

```
Given a **Passage** and **Character**, select the appropriate option from
 ↪  three fields: Character, Question_Type, Difficulty, and return the output
 ↪  in JSON format.
First, select the Character who are likely to be interested in the Passage
 ↪  from the candidates. Then select the Question_Type that the Character
 ↪  might ask about the Passage; Finally, choose the Difficulty of the
 ↪  possible question based on the Passage, the Character, and the
 ↪  Question_Type.
Character: Given by input **Character**

Question_Type:
- keywords: ...
- acquire_knowledge: ...
- summary: ...
- yes_or_no: ...
- background: ...

Difficulty:
- high_school: ...
- university: ...
- phd: ...

Here are some examples
<Example1> <Example2> <Example3>

Now, generate the **output** based on the **Passage** and **Character** from
 ↪  user, the **Passage** will be in {language} language and the **Character**
 ↪  will be in English.
Ensure to generate only the JSON output with content in English.

**Passage**:
{passage}
**Character**:
{character}
```

In the query generation stage, we use the configuration selected in the first stage to guide the generation of queries. Additionally, we explicitly specify the desired length and language of the generated query. The template used is as follows:

```
Given a **Character**, **Passage**, and **Requirement**, generate a query from
 ↪  the **Character**'s perspective that satisfies the **Requirement** and can
 ↪  be used to retrieve the **Passage**. Please return the result in JSON
 ↪  format.

Here is an example:
<example>

Now, generate the **output** based on the **Character**, **Passage** and
 ↪  **Requirement** from user, the **Passage** will be in {corpus_language}
 ↪  language, the **Character** and **Requirement** will be in English.
Ensure to generate only the JSON output, with the key in English and the value
 ↪  in {queries_language} language.

**Character**
{character}
**Passage**
{passage}
**Requirment**
- Type: {type};
- Difficulty: {difficulty};
- Length: the length of the generated sentences should be {length} words;
- Languange: the language in which the results are generated should be
 ↪  {language} language;
```

Stage	Dataset	Size
Weakly Supervised Pre-Training	Synthetic Data	$\sim$ 150M
Supervised Fine Tuning	MS MARCO, NQ, HotpotQA, NLI, Dureader, T^2-Ranking, SimCLUE, MIRACL, MLDR, Mr.TyDi, Multi-CPR, CodeSearchNet .etc + High-quality Synthetic Data	Labeled Data: $\sim$ 7M Synthetic Data: $\sim$ 12M

Table 6: Statistics of training data utilized at each stage.

A.2 Detail Results

MTEB(eng, v2)	Param	Mean (Task)	Mean (Type)	Class-ification	Clus-tering	Pair Class.	Rerank	Retrieval	STS	Summ.
multilingual-e5-large-instruct	0.6B	65.53	61.21	75.54	49.89	86.24	48.74	53.47	84.72	29.89
NV-Embed-v2	7.8B	69.81	65.00	87.19	47.66	88.69	49.61	62.84	83.82	35.21
GritLM-7B	7.2B	67.07	63.22	81.25	50.82	87.29	49.59	54.95	83.03	35.65
gte-Qwen2-1.5B-instruct	1.5B	67.20	63.26	85.84	53.54	87.52	49.25	50.25	82.51	33.94
stella_en_1.5B_v5	1.5B	69.43	65.32	89.38	57.06	88.02	50.19	52.42	83.27	36.91
gte-Qwen2-7B-instruct	7.6B	70.72	65.77	88.52	58.97	85.9	50.47	58.09	82.69	35.74
gemini-embedding-exp-03-07	-	73.3	67.67	90.05	59.39	87.7	48.59	64.35	85.29	38.28
Qwen3-Embedding-0.6B	0.6B	70.70	64.88	85.76	54.05	84.37	48.18	61.83	86.57	33.43
Qwen3-Embedding-4B	4B	74.60	68.09	89.84	57.51	87.01	50.76	68.46	88.72	34.39
Qwen3-Embedding-8B	8B	75.22	68.70	90.43	58.57	87.52	51.56	69.44	88.58	34.83

Table 7: Results on MTEB(eng, v2) (Muennighoff et al., 2023). We compare models from the online leaderboard.

MTEB(cmn, v1)	**Param**	**Mean (Task)**	**Mean (Type)**	Class-ification	Clus-tering	Pair Class.	Rerank	Retrieval	STS
multilingual-e5-large-instruct	0.6B	58.08	58.24	69.80	48.23	64.52	57.45	63.65	45.81
gte-Qwen2-7B-instruct	7.6B	71.62	72.19	75.77	66.06	81.16	69.24	75.70	65.20
gte-Qwen2-1.5B-instruct	1.5B	67.12	67.79	72.53	54.61	79.5	68.21	71.86	60.05
Qwen3-Embedding-0.6B	0.6B	66.33	67.44	71.40	68.74	76.42	62.58	71.03	54.52
Qwen3-Embedding-4B	4B	72.26	73.50	75.46	77.89	83.34	66.05	77.03	61.26
Qwen3-Embedding-8B	8B	73.84	75.00	76.97	80.08	84.23	66.99	78.21	63.53

Table 8: Results on C-MTEB (Xiao et al., 2024) (MTEB(cmn, v1).

MTEB(Code, v1)	Avg.	Apps	COIR-CodeSearch-Net	Code-Edit-Search	Code-Feedback-MT	Code-Feedback-ST	Code-SearchNet-CCR	Code-SearchNet	Code-Trans-Ocean-Contest	Code-Trans-Ocean-DL	CosQA	Stack-Overflow-QA	Synthetic-Text2SQL
$BGE_{multilingual}$	62.04	22.93	68.14	60.48	60.52	76.70	73.23	83.43	86.84	32.64	27.93	92.93	58.67
NV-Embed-v2	63.74	29.72	61.85	73.96	60.27	81.72	68.82	86.61	89.14	33.40	34.82	92.36	60.90
gte-Qwen2-7B-instruct	62.17	28.39	71.79	67.06	57.66	85.15	66.24	86.96	81.83	32.17	31.26	84.34	53.22
gte-Qwen2-1.5B-instruct	61.98	28.91	71.56	59.60	49.92	81.92	72.08	91.08	79.02	32.73	32.23	90.27	54.49
BGE-M3 (Dense)	58.22	14.77	58.07	59.83	47.86	69.27	53.55	61.98	86.22	29.37	27.36	80.71	49.65
Jina-v3	58.85	28.99	67.83	57.24	59.66	78.13	54.17	85.50	77.37	30.91	35.15	90.79	41.49
Qwen3-Embedding-0.6B	75.41	75.34	84.69	64.42	90.82	86.39	91.72	91.01	86.05	31.36	36.48	89.99	76.74
Qwen3-Embedding-4B	80.06	89.18	87.93	76.49	93.21	89.51	95.59	92.34	90.99	35.04	37.98	94.32	78.21
Qwen3-Embedding-8B	80.68	91.07	89.51	76.97	93.70	89.93	96.35	92.66	93.73	32.81	38.04	94.75	78.75
Qwen3-Reranker-0.6B	73.42	69.43	85.09	72.37	83.83	78.05	94.76	88.8	84.69	33.94	36.83	93.24	62.48
Qwen3-Reranker-4B	81.20	94.25	90.91	82.53	95.25	88.54	97.58	92.48	93.66	36.78	35.14	97.11	75.06
Qwen3-Reranker-8B	81.22	94.55	91.88	84.58	95.64	88.43	95.67	92.78	90.83	34.89	37.43	97.3	73.4

Table 9: Performance on `MTEB(Code, v1)` (Enevoldsen et al., 2025). We report nDCG@10 scores.