Gemini Embedding: Generalizable Embeddings from Gemini

Jinhyuk Lee[*], Feiyang Chen[*], Sahil Dua[*], Daniel Cer[*], Madhuri Shanbhogue[*], Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, Feng Han, Andreas Doumanoglou, Nithi Gupta, Fedor Moiseev, Cathy Yip, Aashi Jain, Simon Baumgartner, Shahrokh Shahi, Frank Palma Gomez, Sandeep Mariserla, Min Choi, Parashar Shah, Sonam Goenka, Ke Chen, Ye Xia, Koert Chen, Sai Meher Karthik Duddu, Yichang Chen, Trevor Walker, Wenlei Zhou, Rakesh Ghiya, Zach Gleicher, Karan Gill, Zhe Dong, Mojtaba Seyedhosseini, Yunhsuan Sung, Raphael Hoffmann and Tom Duerig
Gemini Embedding Team, Google[1]

In this report, we introduce Gemini Embedding, a state-of-the-art embedding model leveraging the power of Gemini, Google's most capable large language model. Capitalizing on Gemini's inherent multilingual and code understanding capabilities, Gemini Embedding produces highly generalizable embeddings for text spanning numerous languages and textual modalities. The representations generated by Gemini Embedding can be precomputed and applied to a variety of downstream tasks including classification, similarity, clustering, ranking, and retrieval. Evaluated on the Massive Multilingual Text Embedding Benchmark (MMTEB), which includes over one hundred tasks across 250+ languages, Gemini Embedding substantially outperforms prior state-of-the-art models, demonstrating considerable improvements in embedding quality. Achieving state-of-the-art performance across MMTEB's multilingual, English, and code benchmarks, our unified model demonstrates strong capabilities across a broad selection of tasks and surpasses specialized domain-specific models.

1. Introduction

Embedding models, which transform inputs into dense vector representations, are pivotal for capturing semantic information across various domains and modalities. Text embedding models represent words and sentences as vectors, strategically positioning semantically similar texts in close proximity within the embedding space (Gao et al., 2021; Le and Mikolov, 2014; Reimers and Gurevych, 2019). Recent research has focused on developing general-purpose embedding models capable of excelling in diverse downstream tasks, including information retrieval, clustering, and classification (Cer et al., 2018; Muennighoff et al., 2023). Leveraging their vast pre-training knowledge, large language models (LLMs) have emerged as a promising avenue for constructing such general-purpose embedding models, with the potential to significantly enhance performance across a broad spectrum of applications (Anil et al., 2023a,b; Brown et al., 2020).

The integration of LLMs has revolutionized the development of high-quality embedding models through two primary approaches. Firstly, LLMs have been employed to refine training datasets by generating higher quality examples. Techniques such as hard negative mining (Lee et al., 2024) and synthetic data generation (Dai et al., 2022; Wang et al., 2023) enable the distillation of LLM knowledge into smaller, more efficient embedding models, leading to substantial performance gains. Secondly, recognizing that the embedding model parameters are frequently initialized from language models (Devlin et al., 2019; Karpukhin et al., 2020), researchers have explored leveraging LLM parameters directly for initialization (Ni et al., 2021). While this approach introduces increased

---

[1]See Contributions and Acknowledgments section. *Equal contributions.

		Gemini Embedding	Gecko Embedding[†]	gte-Qwen2-7B-instruct	multilingual-e5-large-instruct	Cohere-embed-multilingual-v3.0	text-embedding-3-large
MTEB(Multilingual) (Enevoldsen et al., 2025)	Mean (Task)	**68.32**	62.13	62.51	63.23	61.10	58.92
	Mean (Type)	**59.64**	54.32	56.00	55.17	53.31	51.48
	- Bitext Mining	79.32	70.73	73.92	**80.13**	70.50	62.17
	- Classification	**71.84**	64.64	61.55	64.94	62.95	60.27
	- Clustering	**54.99**	48.47	53.36	51.54	47.61	47.49
	- Inst. Retrieval	**5.18**	4.08	4.94	-0.40	-1.89	-2.68
	- Multilabel Class.	**29.16**	22.80	25.48	22.91	22.74	22.03
	- Pair Class.	83.64	81.14	**85.13**	80.86	79.88	79.17
	- Reranking	**65.72**	61.22	65.55	62.61	64.07	63.89
	- Retrieval	**67.71**	59.68	60.08	57.12	59.16	59.27
	- STS	**79.40**	76.11	73.98	76.81	74.80	71.68
MTEB(Eng, v2) (Enevoldsen et al., 2025)	Mean (Task)	**73.30**	69.53	70.72	65.53	66.01	66.43
	Mean (Type)	**67.67**	64.82	65.77	61.21	61.43	62.15
MTEB(Code)[*] (Enevoldsen et al., 2025)		**74.66**	65.40	56.41	57.94	51.94	58.95
XOR-Retrieve (Asai et al., 2021)		**90.42**	65.67	N/A	N/A	N/A	68.76
XTREME-UP (Ruder et al., 2023)		**64.33**	34.97	17.39	18.68	N/A	18.80
Commercial Use		✓	✓			✓	✓

Table 1 | Comparison of embedding models on Massive Multilingual Embedding Benchmark: MTEB(Multilingual), MTEB(Eng, v2), and MTEB(Code). We also show results on XOR-Retrieve and XTREME-UP. For MTEBs, we report task and type mean performances. We report MRR@10 for XTREME-UP and Recall@5kt for XOR-Retrieve. [*]: Averaged over seven code tasks available for all models. [†]: For Gecko Embedding (Lee et al., 2024), we evaluate text-embedding-004 on MTEB(Eng, v2), text-embedding-005 on MTEB(Code), and text-multilingual-embedding-002 on others.

computational demands compared to traditional embedding models, empirical evidence suggests that utilizing strong LLMs for initialization can yield significantly superior performance (Lee et al., 2025; Neelakantan et al., 2022; Wang et al., 2023).

In this work, we introduce Gemini Embedding,[2] a novel embedding model initialized from the powerful Gemini large language model (Anil et al., 2023a; Team, 2024). Leveraging Gemini's diverse capabilities, we train Gemini Embedding on a comprehensive suite of embedding tasks. To construct a high-quality, heterogeneous training dataset, we employ Gemini for several critical data curation steps: filtering low-quality examples, determining relevant positive and negative passages for retrieval, and generating rich synthetic datasets. This curated dataset facilitates training with a contrastive learning objective, enabling Gemini Embedding to learn robust semantic representations. Building upon the success of Gecko (Lee et al., 2024), we incorporate task prompts and a pre-finetuning stage to enhance performance. Finally, we utilize Model Soup (Wortsman et al., 2022), a simple yet effective parameter averaging technique, to combine multiple fine-tuned checkpoints, yielding a superior final embedding model.

To rigorously assess the capabilities of Gemini Embedding, we conduct extensive evaluations across a diverse spectrum of tasks and languages. We primarily utilize the Massive Multilingual Text Embedding Benchmark (MMTEB) (Enevoldsen et al., 2025), a comprehensive test suite encompassing over 100 embedding evaluation tasks across more than 250 languages, to provide a thorough evaluation. Notably, Gemini Embedding achieves state-of-the-art performance on MTEB(Multilingual), significantly surpassing the previous best models. Gemini Embedding achieves a first-place ranking

---

[2]Our model is available at `https://ai.google.dev/gemini-api/docs/embeddings`.

on the public leaderboard based on Borda rank,[3] as well as on mean score averaged over tasks where it attains a score of 68.32, a substantial +5.09 improvement over the second-best model, multilingual-e5-large-instruct. Furthermore, it achieves the highest task-type mean of 59.64, a +3.64 improvement over gte-Qwen2-7B-instruct. As summarized in Table 1, Gemini Embedding establishes a new state-of-the-art on multiple other benchmarks such as XOR-Retrieve (Asai et al., 2021) for cross-lingual retrieval. Remarkably, our findings demonstrate that Gemini Embedding exhibits exceptional performance not only in high-resource languages like English but also in numerous low-resource languages, such as Macedonian. We provide a detailed ablation study to elucidate the key factors contributing to Gemini Embedding's superior performance, offering insights into its effectiveness.

2. Related Work

Text Embedding Models Text embeddings are fundamental for a wide array of downstream natural language processing tasks, including semantic similarity, information retrieval, clustering, and classification. Prior models, such as Universal Sentence Encoder (Cer et al., 2018) and Sentence T5 (Ni et al., 2022), have aimed to provide general-purpose embeddings capable of handling diverse applications. However, empirical studies have revealed limitations in their ability to generalize effectively across varied tasks and domains, highlighting the need for more robust and adaptable embedding models. This has motivated the creation of comprehensive benchmarks like MTEB (Enevoldsen et al., 2025; Muennighoff et al., 2023), which emphasize novel task and domain generalization.

LLMs for Embedding Data Generation Synthetic query generation (Bonifacio et al., 2022; Dai et al., 2022; Jeronymo et al., 2023; Nogueira et al., 2019) for given documents or passages has proven highly effective for creating diverse training data for embedding models. Lee et al. (2024) showed that the seed passage from which a synthetic query was generated may not be the best positive passage for that query and proposed an LLM-based approach to find better positive and negative passages. Wang et al. (2023) scaled up synthetic data generation over nearly one hundred languages and hundreds of thousands of tasks by prompting LLMs to first generate a diverse pool of candidate tasks and then generate data as (query, positive, hard negative) triplets conditioned on specific tasks in the pool.

LLMs as Embedding Models Pre-trained LLM encoders with bidirectional attention, such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), have been very popular as backbones for embedding models. DPR (Karpukhin et al., 2020), Contriever (Izacard et al., 2022), Sentence-BERT (Reimers and Gurevych, 2019), Language-agnostic BERT Sentence Embedding (LaBSE) (Feng et al., 2022), Sentence-T5 (Ni et al., 2021), GTR (Ni et al., 2021), and E5 (Wang et al., 2022) are some of the notable ones. Neelakantan et al. (2022) initialized embedding models from decoder-only GPT-3 (Brown et al., 2020) and adapted it for embeddings via continued contrastive pre-training. They have drastically scaled their embedding model up to 175 billion parameters, demonstrating scaling gains from pre-trained LLM backbones.

Several recent embedding models such as E5-Mistral (Wang et al., 2023), SFR-Mistral (Meng et al., 2024), BGE-ICL (Li et al., 2024), and NV-Embed (Lee et al., 2025) have been initialized from the Mistral-7B (Jiang et al., 2023) backbone and then further adapted as embedding models. These models generally outperform the BERT or T5 based models, showing the benefits of initializing from pre-trained LLMs. However, their reliance on extensive in-domain training datasets has resulted in overfitting to specific benchmarks (Enevoldsen et al., 2025).

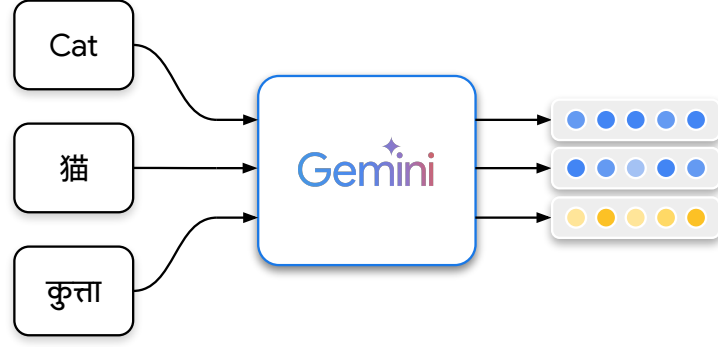[3] https://huggingface.co/spaces/mteb/leaderboard; March 10th, 2025.

Figure 1 | Gemini Embedding represents text as dense vectors where semantically similar text inputs are mapped to vectors near one another in the vector space. Currently it supports more than 100+ languages, and its embeddings can be used for various tasks such as retrieval and classification.

3. Gemini Embedding

In this section we provide technical details of the Gemini Embedding model in terms of the model architecture, the objective function, and the training recipe.

3.1. Model Architecture

The Gemini Embedding model is built to create holistic representations of inputs for diverse downstream tasks, including retrieval, clustering, classification, and ranking by leveraging the power of Gemini. The embedding model is initialized from Gemini and further refined. This allows Gemini Embedding to build representations on top of the vast knowledge already present in Gemini's parameters. In this sense, initializing the embedding model from Gemini can be seen as the "pre-training" of the Gemini Embedding model.

An input sequence $\mathbf{T}$ of L tokens is processed by $\mathcal{M}$, a transformer with bidirectional attention initialized from Gemini, producing a sequence of token embeddings $\mathbf{T}_{\text{embed}} = \mathcal{M}(\mathbf{T}) \in \mathbb{R}^{L \times d_{\mathcal{M}}}$, where $d_{\mathcal{M}}$ is the model dimension. To generate a single embedding representing all the information in the input, a pooler $\mathcal{P}$ is applied, $\mathbf{P}_{\text{embed}} = \mathcal{P}(\mathbf{T}_{\text{embed}}) \in \mathbb{R}^{d_{\mathcal{M}}}$. Prior research ([Suganthan et al., 2025](#)) has demonstrated that simple pooling strategies can be effective in model adaptation. Therefore we have chosen mean pooling, and simply average the token embeddings along the sequence axis. Finally, a randomly initialized linear projection f is applied to scale the embedding to the target dimension, $\mathbf{E} = f(\mathbf{P}_{\text{embed}}) \in \mathbb{R}^d$, where d is the output embedding dimension.

3.2. Training Objective

The Gemini Embedding model was trained with a noise-contrastive estimation (NCE) loss with in-batch negatives. The exact loss differs slightly depending on the stage of training. In general, a training example includes a query q_i, a positive target p_i^+ and (optionally) a hard negative target p_i^-. Each example also has a prescribed task string t, for example "question answering" or "fact checking", describing the nature of the task. The query and passages are embedded as vectors in $\mathbb{R}^d$:

$$\mathbf{q}_i = f(\texttt{mean\_pool}(\mathcal{M}(t \oplus q_i))), \quad \mathbf{p}_i^{\pm} = f(\texttt{mean\_pool}(\mathcal{M}(p_i^{\pm}))). \tag{1}$$

Given a batch of size B the loss applied to these embeddings is as follows:

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B}\left[-\log\frac{e^{\text{sim}(\mathbf{q}_i,\mathbf{p}_i^+)/\tau}}{e^{\text{sim}(\mathbf{q}_i,\mathbf{p}_j^-)/\tau}+\sum_{j=1}^{B}\texttt{mask}(i,j)e^{\text{sim}(\mathbf{q}_i,\mathbf{p}_j^+)/\tau}}\right] \tag{2}$$

where $\text{sim}(\mathbf{x},\mathbf{y}) = \mathbf{x}^\top\mathbf{y}/\|\mathbf{x}\|\|\mathbf{y}\|$ is cosine similarity, and

$$\texttt{mask}(i,j) = \begin{cases} 0 & \text{if } q_i = q_j \text{ or } p_i^+ = p_j^+, \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

This masking term is particularly relevant for classification tasks, where the number of targets (labels) is small. The first term in the denominator is omitted if no hard negative is provided. In contrast with Gecko (Lee et al., 2024), we omit the same-tower negatives (Moiseev et al., 2023) from the loss, as we find this decreases performance for most tasks due to the potential of false negatives.

In order to support different dimensions of embeddings with a single model, we adapt the above loss using MRL (Kusupati et al., 2022), which adapts the loss above into k separate losses across k overlapping sub-dimensions of the embedding (e.g. multi-loss training with one loss for the first 768 embedding dimensions, another for the first 1,536 dimensions, and so on). Gemini Embedding provides $d = 3,072$ dimensional embeddings, with the MRL support on 768 and 1,536 dimensions.

3.3. Recipe

Initializing the embedding model from the Gemini parameters is a good starting point that leverages the language model power. This initialization can be considered a "pre-training" of the embedding model. However, in order to truly capture the generalization capabilities of initialization, we found it beneficial to leverage a two-stage training pipeline.

Pre-finetuning First, the model is "pre-finetuned" on a large number of potentially noisy (query, target) pairs, omitting the hard-negative term from the loss function. We find it beneficial to use a large batch size, as the primary objective is to adapt the parameters from autoregressive generation to encoding. The larger batch size also provides a more stable gradient, mitigating the impact of noise in this phase of training. Due to the larger size of the pre-finetuning dataset, pre-finetuning is performed for a substantially greater number of steps compared to fine-tuning.

Finetuning Next, the model is fine-tuned on a large mixture of task-specific datasets which contain (query, target, hard negative target) triples. For this phase of training we found it beneficial to use smaller batch sizes (e.g., less than 1024), and furthermore limit each batch to a single dataset, as distinguishing a given positive target from in-batch targets from the same task provides greater signal than discerning (say) a retrieval target from a classification label. We perform a grid search of various training hyperparameters, including the inclusion and exclusion of components of the mixture, to obtain candidate checkpoints.

Model Soup To obtain additional generalization performance, we averaged the parameters obtained from individual fine-tuning runs. We experimented with different combinations of parameters, including averaging checkpoints from the same training run (Izmailov et al., 2018), from different training runs (Wortsman et al., 2022), as well as various weighted averages. The final set of ingredient checkpoints were obtained through a combination of intentional data variation as well as manual checkpoint selection and experimentation.

4. Datasets

Our training data mixture contains diverse multilingual embedding tasks as well as code retrieval tasks. Gemini is used in three different ways to improve the quality of our data: synthetic data generation, data filtering, and hard negative mining.

4.1. Training Data Mixture

Pre-finetuning   Our pre-finetuning stage aims to maximize the exposure of diverse training datasets to Gemini Embedding models. We leverage a billion-scale web corpus and used title and passage pairs as input and positive target pairs, similar to some prior work (Lee et al., 2024; Neelakantan et al., 2022). Despite being very simple, this technique is consistently found to be effective even when the embedding model is initialized from an LLM.

Fine-tuning   For fine-tuning, we prepare three different mixtures aiming for task diversity, language diversity, and coding capability. For task diversity, we use a subset of academic datasets used by Gecko (Lee et al., 2024) as well as several synthetic datasets introduced in §4.2. Unlike existing models on the classic MTEB (Muennighoff et al., 2023), we excluded many in-domain MTEB datasets, which improved the performance only on their own test split mainly due to train-test leakage or dataset bias. The training mixture rate was decided based on a fine-grained grid search, initialized from the optimal number of training steps to converge on each training dataset.

4.2. Improving Data Quality with Gemini

Synthetic Data Generation   Recent embedding evaluation benchmarks such as MMTEB (Enevoldsen et al., 2025) contain many different tasks other than retrieval. We diversify and improve our training mixture by adding synthetically generated datasets for two task types: retrieval and classification. For retrieval, we extended our prior work on synthetic data generation using Gemini enhanced adaptations of FRet (Lee et al., 2024) and SWIM-IR (Thakur et al., 2024). Using few-shot prompting, we first use Gemini to generate synthetic queries for web passages followed by a Gemini auto-rater to filter lower-quality examples (e.g., unrealistic search queries). For classification, we generate synthetic counterfactual, sentiment, and review classification datasets in English. To increase the quality of these synthetic datasets we developed multi-stage prompting strategies, such as conditioning on synthetic user, product, or movie generations in a hierarchical manner and sampling from the tail of longer lists of generations, as diversity naturally increases with generation length.

Data Filtering   Our training data mixture includes many human-annotated datasets. We noticed that many retrieval datasets have quality issues of incorrect positive or negative targets for a query. We use Gemini to filter such bad examples. Based on our few-shot prompting for data quality assessment, we remove low quality examples.

Hard Negative Mining   A standard technique when training embedding models is to mine "hard negatives," i.e. targets which are semantically similar to a true positive target but do not answer the query (Reddi et al., 2019). We mine hard negatives for our retrieval datasets using Gemini. We first train a Gemini-initialized embedding model without using any hard negatives. Based on this initial embedding model, we retrieve top k nearest neighbors for each query. Each nearest neighbor is then scored by Gemini along with the query. We follow Lee et al. (2024) and employ two different prompting strategies—graded classification and query likelihood—combining the scores with Reciprocal Rank Fusion (RRF) (Cormack et al., 2009). We found that the lowest-scoring nearest neighbors, (the k-th neighbor after being sorted by Gemini scores) serve as the best hard negatives.

Model Name	Rank	Mean (Task)	Mean (Type)	Bitext Mining	Class.	Clus.	Inst. Retrieval	Multi. Class.	Pair. Class.	Rerank.	Retrieval	STS
Gemini Embedding	**1**	**68.3**	**59.6**	79.3	**71.8**	**55.0**	**5.2**	**29.2**	83.6	**65.6**	**67.7**	**79.4**
Linq-Embed-Mistral	2	61.5	54.2	70.3	62.2	51.3	0.9	24.8	80.4	64.4	58.7	74.9
gte-Qwen2-7B-instruct	3	62.5	56.0	73.9	61.6	53.4	4.9	25.5	**85.1**	65.6	60.1	74.0
multilingual-e5-large-instruct	4	63.2	55.2	**80.1**	64.9	51.5	-0.4	22.9	80.9	62.6	57.1	76.8
SFR-Embedding-Mistral	5	60.9	54.0	70.0	60.0	52.6	0.2	24.6	80.3	64.2	59.4	74.8
GritLM-7B	6	60.9	53.8	70.5	61.8	50.5	3.5	22.8	79.9	63.8	58.3	73.3
text-multilingual-embedding-002	7	62.1	54.3	70.7	64.6	48.5	4.1	22.8	81.1	61.2	59.7	76.1
GritLM-8x7B	8	60.5	53.4	68.2	61.6	50.9	2.4	24.4	79.7	62.6	57.5	73.2
e5-mistral-7b-instruct	9	60.3	53.2	70.6	60.3	51.4	-0.6	22.2	81.1	63.8	55.8	74.0
Cohere-embed-multilingual-v3.0	10	61.1	53.3	70.5	63.0	47.6	-1.9	22.7	79.9	64.1	59.2	74.8
gte-Qwen2-1.5B-instruct	11	59.5	52.8	62.5	58.3	52.6	0.7	24.0	81.6	62.6	60.8	71.6
bilingual-embedding-large	12	60.9	53.0	73.6	62.8	47.2	-3.0	22.4	79.8	61.4	55.1	77.8

Table 2 | Performance of top leaderboard models on MTEB(Multilingual).

5. Evaluation

Gemini Embedding is assessed on a comprehensive collection of task types, domains, languages, and language pairs (e.g., Hindi queries retrieving English content) using benchmark evaluations from the Massive Multilingual Text Embedding Benchmark, MMTEB (Enevoldsen et al., 2025), and the cross-lingual benchmarks XTREME-UP (Ruder et al., 2023) and XOR-Retrieve (Asai et al., 2021).

5.1. Benchmarks and Tasks

MMTEB consists of a large collection of individual evaluation tasks covering 250+ languages and 10 task types: Bitext Mining, Classification, Clustering, Instruction Retrieval, Multilabel Classification, Pair Classification, Reranking, Retrieval, STS, and Summarization. Our MMTEB evaluations include 164 individual evaluation tasks consisting of 132 evaluation tasks for MTEB(Multilingual), 41 tasks for MTEB(Eng, v2), and 12 code retrieval tasks for MTEB(Code). Notably, MTEB(Multilingual) contains 250+ languages. XOR-Retrieve and XTREME-UP provide cross-lingual retrieval evaluations, with XOR-Retrieve pairing English passages with retrieval queries in 7 different languages and XTREME-UP similarly pairing English passages with queries in 20 underrepresented Indo-European languages.

5.2. Overall Performance

Gemini Embedding's overall performance along with that of other top performing models is presented in Table 1 on the following evaluations: three benchmarks from MMTEB, MTEB(Multilingual), MTEB(Eng, v2), MTEB(Code); and the two cross-lingual benchmarks XOR-Retrieve and XTREME-UP.

Gemini Embedding establishes a new state-of-the-art in performance, achieving the highest overall performance on the MTEB(Multilingual) leaderboard (March 10th, 2025) with a substantial performance lead over all previous top performing models on each of the overall metrics summarizing aggregate performance across tasks: Task Mean (equal weighting of all tasks): 68.32, Task Type Mean (equal weighting of all task types): 59.64, and Borda rank #1 (official leaderboard ranking metric). Gemini Embedding's performance advantage is not limited to just MTEB(Multilingual). Within a single unified model and shared embedding space, Gemini Embedding's capabilities allow it to achieve: (i) **#1 ranking on MTEB(Multilingual)**, (ii) **#1 ranking on MTEB(Eng, v2)**, (iii) **#1 ranking on MTEB(Code)**, and (iv) **excellent cross-lingual retrieval on XOR-Retrieve and XTERME-UP**, advancing the state-of-the-art for general-purpose embeddings as cross-lingual representations.

Model Name	Rank	Mean (Task)	Mean (Type)	Class.	Clus.	Pair. Class.	Rerank.	Retrieval	STS	Summ.
Gemini Embedding	**1**	**73.3**	**67.7**	90.1	59.4	87.7	48.6	**64.4**	**85.3**	**38.3**
Linq-Embed-Mistral	2	69.8	65.3	83.0	54.1	88.4	49.4	60.1	84.7	37.3
jasper_en_vision_language_v1	3	71.4	66.7	**90.3**	**60.5**	88.1	50.0	56.1	84.4	37.2
SFR-Embedding-Mistral	4	69.3	64.9	80.5	54.9	88.6	50.2	59.3	84.8	36.3
NV-Embed-v2	5	69.8	65.0	87.2	47.7	**88.7**	49.6	62.8	83.8	35.2
text-embedding-005 (Gecko)	6	69.6	64.8	86.0	51.9	87.6	48.8	58.8	85.2	35.1
text-embedding-004 (Gecko)	7	69.5	64.8	86.0	51.5	87.7	48.5	59.1	84.8	36.1
gte-Qwen2-7B-instruct	8	70.7	65.8	88.5	59.0	85.9	**50.5**	58.1	82.7	35.7
e5-mistral-7b-instruct	9	68.0	64.0	79.9	51.4	88.4	49.8	57.6	84.3	36.6
stella_en_400M_v5	10	69.4	64.8	88.3	57.7	87.2	49.6	52.7	83.9	34.5
stella_en_1.5B_v5	11	69.4	65.3	89.4	57.1	88.0	50.2	52.4	83.3	36.9
gte-Qwen2-1.5B-instruct	12	67.2	63.3	85.8	53.5	87.5	49.3	50.3	82.5	33.9

Table 3 | Performance of top leaderboard models on MTEB(Eng, v2).

Model Name	Rank	Mean All	Mean -COIR	AppsR.	COIR	CESR	CSNCCR	CSNR	CTOC	CTODL	CQA
Gemini Embedding	**1**	**75.5**	**74.7**	**93.8**	81.1	**81.6**	84.7	91.3	89.5	31.5	50.2
inf-retriever-v1-1.5b	2	62.9	60.6	38.9	78.6	67.2	75.5	90.9	85.0	33.8	33.1
text-embedding-005 (Gecko)	3	63.3	65.4	91.3	48.4	54.4	55.7	87.2	82.8	34.4	**52.2**
voyage-code-3	4	-	-	93.6	**89.4**	-	**90.1**	**94.0**	**95.0**	**38.6**	34.5
NV-Embed-v2	5	-	59.4	29.1	-	74.0	68.8	86.6	89.1	33.4	34.8
voyage-3	6	-	67.3	73.0	-	75.6	77.9	92.3	89.9	33.9	28.7
GritLM-7B	7	-	62.4	35.1	-	74.6	86.7	86.7	89.2	33.0	31.2
KaLM-emb.-mling.-mini-v1	8	-	57.4	46.8	-	60.0	59.5	88.0	79.9	34.0	33.6
text-embedding-3-large	9	-	59.0	28.4	-	71.1	73.2	90.5	84.3	34.2	31.0
NV-Embed-v1	10	-	57.7	30.3	-	70.8	65.1	85.8	85.1	33.1	33.4
SFR-Embedding-Mistral	11	-	56.7	26.1	-	68.8	64.5	86.7	83.5	32.9	34.3
Linq-Embed-Mistral	12	-	57.5	30.2	-	70.6	64.5	87.1	84.9	32.8	32.6

Table 4 | Performance of top leaderboard models on MTEB(Code).

MTEB(Multilingual) leaderboard   In Table 2, Gemini Embedding is compared with top-ranked models from MTEB(Multilingual). Achieving the highest Borda rank and excellent overall performance across task types, Gemini Embedding particularly excels at Classification (+9.6), Clustering (+3.7) and Retrieval (+9.0) compared to the second-best model.

MTEB(Eng, v2) leaderboard   Comparing with top-ranked MTEB(Eng, v2) leaderboard models in Table 3, Gemini Embedding achieves the highest Borda rank and great overall performance across task types, with particularly striking performance improvements on Classification (+7.1), Clustering (+5.3), and Retrieval (+4.3) compared to the second-best model.

MTEB(Code) leaderboard   The eight tasks present on the MTEB(code) leaderboard, which excludes the four additional MTEB(code) tasks CodeFeedbackMT, CodeFeedbackST, StackOverflowQA, and SyntheticText2SQL, are shown in Table 4. Only a few models, including both Gemini Embedding and Google's Gecko model, have been submitted to the MTEB(Code) leaderboard with evaluations over all tasks. On the MTEB(Code) leaderboard, Gemini Embedding once again achieves the highest Borda rank and mean performance across all eight evaluation tasks. Since the majority of other top models on MTEB(Code) are missing COIRCodeSearchNetRetrieval (COIR), we also report the mean performance over the seven remaining tasks, **Mean -COIR**. Gemini Embedding still achieves the best mean performance over the seven **Mean -COIR** evaluation tasks.

	Average	as	bho	brx	gbm	gom	gu	hi	hne	kn	mai	ml	mni	mr	mwr	or	pa	ps	sa	ta	ur
Gemini Embedding	**64.3**	**69.2**	**66.4**	**25.7**	**64.9**	**65.5**	**70.3**	**69.1**	**68.3**	**69.5**	**68.4**	**70.8**	**44.4**	**68.8**	**66.5**	**65.8**	**69.5**	**61.9**	**68.1**	**68.6**	**64.8**
Gecko i18n Embedding	35.0	31.9	39.7	3.8	37.4	26.0	42.9	46.3	42.0	41.6	44.1	45.5	9.4	41.5	40.7	19.4	40.9	33.0	35.9	40.5	37.0
voyage-3-large	39.2	34.3	44.8	7.9	46.6	27.1	46.7	54.3	45.3	41.5	48.3	45.3	19.2	45.5	47.9	32.3	48.4	26.8	40.0	36.0	45.6
Linq-Embed-Mistral	24.6	23.8	38.1	8.6	37.0	21.7	11.6	44.2	39.7	21.7	38.5	10.2	14.7	31.4	36.2	10.7	8.3	13.8	37.7	14.3	29.3
multiling.-e5-large-instr.	18.7	21.2	21.9	1.5	19.3	8.7	13.9	30.6	22.6	24.2	24.0	8.6	6.3	23.0	19.8	17.3	24.5	15.9	19.1	22.9	28.2
gte-Qwen2-7B-instruct	17.4	14.7	22.7	5.4	23.0	7.0	19.1	30.4	19.1	16.2	25.9	21.7	7.2	23.8	24.0	11.3	19.2	11.0	21.1	9.7	15.5
text-embedding-3-large	18.8	18.2	28.8	3.3	28.4	11.1	14.6	40.4	29.3	17.1	31.1	15.6	2.9	25.5	28.7	8.3	11.3	6.8	26.6	6.0	22.0

Table 5 | Performance of top multilingual models on XTREME-UP (MRR@10).

Input: প্রশ্ন: মহেঞ্জো-দাৰো সভ্যতা কোন নদীৰ দ্বাৰা আৱদ্ধ আছিল? (language: <u>Assamese</u>)

Translation: The Mohenjo-Daro civilization was surrounded by which river?

text-embedding-3-large

Retrieved Passage: The city of Mymensingh stands on the bank of the Old Brahmaputra, as the 1897 Assam earthquake changed the main flow from Brahmaputra to the Jamuna River which flows west of the greater Mymensingh region. After about hundred years, a port city of sea vessels from England lost its status as the river became a seasonal flow. Some former residences of colonial officials along the side of the river in the city are nowadays government buildings.

Gemini Embedding

Retrieved Passage: Mohenjo-daro is located west of the Indus River in Larkana District, Sindh, Pakistan, in a central position between the Indus River and the Ghaggar-Hakra River. It is situated on a Pleistocene ridge in the middle of the flood plain of the Indus River Valley, around from the town of Larkana. The ridge was prominent during the time of the Indus Valley Civilization, allowing the city to stand above the surrounding flood, but subsequent flooding has since buried most of the ridge in silt deposits. The Indus still flows east of the site, but the Ghaggar-Hakra riverbed on the western side is now dry.

Input: आंध्र प्रदेश की राजकीय पक्षी कोसी है? (language: <u>Hindi</u>)

Translation: Which is the official bird of Andhra Pradesh?

text-embedding-3-large

Retrieved Passage: The Indian National Congress (INC), the Praja Socialist Party and the Krishi Lok Party were the major parties in the 1950s. Later the Communist Party of India (CPI) became the dominant opposition party. In the 1967 state assembly elections, all socialist parties were eliminated and the CPI lost opposition party status.

Gemini Embedding

Retrieved Passage: The Indian state of Andhra Pradesh has about 484 bird species within its political boundary. This following list of birds is based on the taxonomic treatment of Clements's 6th edition but includes recent revisions. The Indian roller is the state bird of Andhra Pradesh.

Figure 2 | Gemini Embedding supports cross-lingual retrieval where different languages can be used for queries and passages. We show two examples from XTREME-UP showing the strong cross-lingual retrieval capability of Gemini Embedding. Despite Assamese being a relatively low-resource language and the Hindi query having a typo, the Gemini Embedding model correctly understood the key entities and the contexts in the queries and retrieved the correct passages.

XTREME UP The performance of Gemini Embedding along with the top-performing multilingual models on XTREME-UP cross-lingual retrieval is presented in Table 5. XTREME-UP requires mapping queries in 20 underrepresented languages to English passages. Gemini Embedding demonstrates a remarkable improvement in cross-lingual retrieval with its general-purpose embeddings.

5.3. Qualitative Examples

In Figure 2, we show examples from XTREME-UP that show the cross-lingual retrieval capability of Gemini Embedding. The two queries are given in Assamese and Hindi, and the task is to retrieve relevant English passages that contain the answers. Each query without any translation is encoded and the highest-scoring English passages are retrieved using cosine similarity. Gemini Embedding found the right passages showcasing its strong capability on multilingual and cross-lingual tasks.

	MTEB(Multilingual)	MTEB(Eng, v2)	MTEB(Code)	XOR-Retrieve	XTREME-UP
Gemini Embedding	**68.32**	**73.28**	**74.66**	**90.42**	64.33
Pre-Finetuning					
Pre-finetuning Only	**48.89**	**50.99**	**46.18**	76.64	21.22
No Training	30.55	28.17	9.86	-	-
Fine-tuning Mixtures					
English Only (Diverse Task)	**66.75**	**72.77**	58.68	85.70	49.34
Multilingual Only (Retrieval)	58.24	61.88	58.75	**89.00**	**65.06**
Code Only (Retrieval)	60.20	62.25	**72.08**	82.16	34.74

Table 6 | Results using different training mixtures for MTEBs (task mean), XTREME-UP (MRR@10), and XOR-Retrieve (Recall@5kt). Using a Gemini foundation, the English Only mixture is able to achieve good performance on MTEB(Multilingual), MTEB(Eng, v2) and XOR-Retrieve. Multilingual fine-tuning helps the most on the long-tail languages in XTREME-UP. Ablations exclude model souping.

	Average	AmazonCounterfactual	AmazonPolarity	AmazonReviews	Emotion
w/o Synthetic	57.57	65.43	67.29	48.84	48.70
w/ Synthetic	**75.17** (+17.6)	**91.30**	**96.51**	**57.00**	**55.90**
Gecko Embedding	66.78	66.52	97.28[*]	51.24	52.09
Gemini Embedding	**76.09**	**92.70**	96.10	**59.30**	**56.27**

Table 7 | Results on MTEB classification using synthetic datasets. Self-training on Gemini generated training data dramatically improves model performance, **+17.6**. Ablation models exclude souping. [*] Gecko training mixtures include training sets provided by several classification tasks from Huggingface.

6. Ablation Study

To better understand how Gemini Embedding achieves great performance across many different tasks and languages, we provide a systematic analysis of our training recipe.

6.1. Does Gemini Embedding Generalize to Multilingual Tasks?

In Table 6, we show how Gemini Embedding can generalize over different languages and tasks. In the middle rows, we show our model's performance before fine-tuning: no training and pre-finetuning only. Pre-finetuning greatly improves the performance across multiple benchmarks. The bottom rows show the effect of further fine-tuning the pre-finetuned checkpoints. We find that training on the English-only mixture still achieves very strong performance on MTEB(Multilingual) where the evaluations are mostly zero-shot. Remarkably, even when training our model on the English-only mixture, we are able to outperform the top embedding models on XTREME-UP.[4] This shows Gemini Embedding can generalize over different languages even if its training mixture contains only a single language. On the other hand, our multilingual-only mixture consists of only retrieval datasets but not other task types such as classification. Its lower score indicates that task diversity matters more than language diversity for fine-tuning in Gemini Embedding.

---

[4] +10.1 MMR@10 for English-only fine-tuning in Table 6 vs. the top performing non-Gemini model in Table 5

	Average	ar	bn	de	en	es	fa	fi	fr	hi	id	ja	ko	ru	sw	te	th	yo	zh
w/o Filtering	59.8	**74.8**	71.5	46.5	54.6	44.5	51.3	66.8	46.0	**59.4**	34.9	56.9	55.1	62.6	69.7	73.6	71.9	**85.0**	51.7
w/ Filtering	**63.7** (+3.9)	74.2	**74.7**	**52.9**	**54.7**	**47.3**	**55.5**	**74.7**	**49.5**	59.3	**47.1**	**61.9**	**63.3**	**64.8**	**76.0**	**75.0**	**75.0**	83.3	**57.0**
Gecko Embedding	56.2	64.3	66.7	49.1	45.3	48.5	49.2	65.2	45.1	55.0	44.7	52.6	57.5	55.1	67.4	74.5	66.5	54.0	50.7
Gemini Embedding	**70.1**	**78.3**	**79.0**	**59.8**	**58.7**	**57.0**	**60.9**	**78.0**	**55.6**	**65.4**	**54.3**	**75.1**	**68.9**	**73.4**	**81.0**	**80.5**	**80.8**	**88.8**	**65.7**

Table 8 | Results on filtering the MIRACL datasets. We show that proper filtering of retrieval datasets using LLMs can greatly improve the performance.

Figure 3 | Results on retrieval datasets with different number of hard negatives. We show that our hard negatives are mostly useful.

6.2. How Does Gemini Improve Data Quality?

Synthetic Data Generation   We show the effectiveness of our multi-stage prompting strategy to create diverse, realistic synthetic classification datasets in Table 7. Note that these are zero-shot synthetic datasets, so no actual examples from the original datasets were used when prompting Gemini. Training on our synthetic classification datasets greatly improves the performance on all datasets. We find that the performance with synthetic datasets can match the performance of in-domain datasets (e.g. Gecko on AmazonPolarity), and our multi-stage prompting strategy even allows for controllable generation, raising the possibility of reducing bias compared to real data.

Data Filtering   We use Gemini to filter retrieval datasets. We test filtering the MIRACL (Zhang et al., 2023) training datasets, which contain retrieval datasets in 18 different languages, and measure the impact of training on the filtered dataset. Table 8 shows that filtered results consistently show better results across different languages showing only minor drops for some languages. As demonstrated in Table 6, our English mixture helps to improve the quality on multilingual tasks, making Gemini Embedding the best in Table 8 as well.

Hard Negative Mining   We examine the quality of our hard negatives selected by Gemini. As demonstrated in Figure 3, incorporating hard negatives generally enhances our model's retrieval performance across the four datasets. However, excessive hard negatives often led to overfitting, causing performance degradation for retrieval tasks. Future work will explore regularization techniques and better hard negative sampling strategies to address overfitting.

7. Future Work

Beyond the text embedding capabilities described here, we will explore extending the embedding capabilities for other modalities like image, video, and audio. We want to leverage the powerful multi-modal capabilities of Gemini to make the Gemini Embedding model comprehensive (Jiang et al., 2024) in terms of representing different combinations of modalities together in a single embedding space. This will require curating multi-modal data tasks suitable for learning generalizable representations. We will also explore training recipes that will balance the performance of a single model across different uni-modal and multi-modal capabilities.

8. Conclusion

Gemini Embedding is a unified, general-purpose, and highly-capable embedding model that capitalizes on the strong capabilities of Gemini to advance the state-of-the-art in representation learning. Building on an excellent foundation provided by Gemini's multilingual and code understanding capabilities, Gemini Embedding generates a versatile encoding of model inputs into representations with a wide range of capabilities over many languages, domains, and task types including: classification, similarity search, clustering, ranking, and retrieval. Gemini Embedding both adapts the capabilities of Gemini to representation learning and uses Gemini itself to generate many of the training sets for this adaptation. The resulting representations benefit from the underlying capabilities of Gemini itself while also being efficient to precompute, cache, and re-use them. Efficiently cacheable and reusable representations unlock the ability to apply the power of Gemini in new compute and latency-sensitive settings.

Rigorous evaluations provided by the Massive Multilingual Text Embedding Benchmark (MMTEB) reveal substantial gains over previous top-performing models advancing the state-of-the-art in performance on multilingual, English, and code evaluations. Beyond strong overall performance, Gemini Embedding particularly excels at classification, clustering and retrieval tasks. The advanced versatile and unified capabilities provided by Gemini Embedding and the ability to precompute representations enables the power of Gemini to be leveraged more broadly by both researchers and developers alike.

References

R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023a.

R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023b.

A. Asai, J. Kasai, J. H. Clark, K. Lee, E. Choi, and H. Hajishirzi. Xor qa: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, 2021.

L. Bonifacio, H. Abonizio, M. Fadaee, and R. Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392, 2022.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174, 2018.

G. V. Cormack, C. L. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759, 2009.

Z. Dai, V. Y. Zhao, J. Ma, Y. Luan, J. Ni, J. Lu, A. Bakalov, K. Guu, K. B. Hall, and M.-W. Chang. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*, 2022.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

K. Enevoldsen, I. Chung, I. Kerboua, M. Kardos, A. Mathur, D. Stap, J. Gala, W. Siblini, D. Krzemiński, G. I. Winata, et al. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*, 2025.

F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic BERT sentence embedding. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, May 2022. URL https://aclanthology.org/2022.acl-long.62/.

T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, 2021.

G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

V. Jeronymo, L. Bonifacio, H. Abonizio, M. Fadaee, R. Lotufo, J. Zavrel, and R. Nogueira. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*, 2023.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Z. Jiang, R. Meng, X. Yang, S. Yavuz, Y. Zhou, and W. Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024.

V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Y. Wu, S. Edunov, D. Chen, and W. tau Yih. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906, 2020.

A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

C. Lee, R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, and W. Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *ArXiv*, 2025. URL https://arxiv.org/abs/2405.17428.

J. Lee, Z. Dai, X. Ren, B. Chen, D. Cer, J. R. Cole, K. Hui, M. Boratko, R. Kapadia, W. Ding, Y. Luan, S. M. K. Duddu, G. H. Abrego, W. Shi, N. Gupta, A. Kusupati, P. Jain, S. R. Jonnalagadda, M.-W. Chang, and I. Naim. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024.

C. Li, M. Qin, S. Xiao, J. Chen, K. Luo, Y. Shao, D. Lian, and Z. Liu. Making text embedders few-shot learners. *ArXiv*, 2024. URL https://arxiv.org/abs/2409.15700.

R. Meng, Y. Liu, S. R. Joty, C. Xiong, Y. Zhou, and S. Yavuz. Sfrembedding-mistral: enhance text retrieval with transfer learning. *Salesforce AI Research Blog*, 3:6, 2024.

F. Moiseev, G. H. Abrego, P. Dornbach, I. Zitouni, E. Alfonseca, and Z. Dong. Samtone: Improving contrastive loss for dual encoder retrieval models with same tower negatives. *arXiv preprint arXiv:2306.02516*, 2023.

N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2006–2029, 2023.

A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

J. Ni, C. Qu, J. Lu, Z. Dai, G. H. 'Abrego, J. Ma, V. Zhao, Y. Luan, K. B. Hall, M.-W. Chang, and Y. Yang. Large dual encoders are generalizable retrievers. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

J. Ni, G. H. Abrego, N. Constant, J. Ma, K. Hall, D. Cer, and Y. Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, 2022.

R. Nogueira, W. Yang, J. Lin, and K. Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

S. J. Reddi, S. Kale, F. Yu, D. Holtmann-Rice, J. Chen, and S. Kumar. Stochastic negative mining for learning with large output spaces. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1940–1949. PMLR, 2019.

N. Reimers and I. 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)*, pages 3982–3992, 2019.

S. Ruder, J. H. Clark, A. Gutkin, M. Kale, M. Ma, M. Nicosia, S. Rijhwani, P. Riley, J.-M. Sarr, X. Wang, et al. Xtreme-up: A user-centric scarce-data benchmark for under-represented languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1856–1884, 2023.

P. Suganthan, F. Moiseev, L. Yan, J. Wu, J. Ni, J. Han, I. Zitouni, E. Alfonseca, X. Wang, and Z. Dong. Adapting decoder-based language models for diverse encoder downstream tasks, 2025. URL https://arxiv.org/abs/2503.02656.

G. Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

N. Thakur, J. Ni, G. Hernandez Abrego, J. Wieting, J. Lin, and D. Cer. Leveraging LLMs for synthesizing training data across many languages in multilingual dense retrieval. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, June 2024. URL https://aclanthology.org/2024.naacl-long.426/.

L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

X. Zhang, N. Thakur, O. Ogundepo, E. Kamalloo, D. Alfonso-Hermelo, X. Li, Q. Liu, M. Rezagholizadeh, and J. Lin. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023.

9. Full Results

Task Name	Performance	Task Name	Performance
AILAStatutes	48.77	NollySentiBitextMining	68.71
AfriSentiClassification	53.56	NordicLangClassification	85.97
AlloProfClusteringS2S.v2	56.36	NorwegianCourtsBitextMining	93.42
AloprofReranking	81.77	NusaParagraphEmotionClassification	56.38
AmazonCounterfactualClassification	88.20	NusaTranslationBitextMining	77.52
ArXivHierarchicalClusteringP2P	64.92	NusaX-senti	80.31
ArXivHierarchicalClusteringS2S	63.84	NusaXBitextMining	82.52
ArguAna	86.44	OdiaNewsClassification	91.84
ArmenianParaphrasePC	96.89	OpusparcusPC	96.62
BUCC.v2	98.99	PAC	71.68
BelebeleRetrieval	90.73	PawsXPairClassification	59.99
BibleNLPBitextMining	20.72	PlscClusteringP2P.v2	74.31
BigPatentClustering.v2	38.06	PoemSentimentClassification	59.66
BiorxivClusteringP2P.v2	53.86	PolEmo2.0-OUT	77.53
BornholmBitextMining	51.69	PpcPC	95.50
BrazilianToxicTweetsClassification	28.02	PunjabiNewsClassification	82.61
BulgarianStoreReviewSentimentClassfication	78.13	RTE3	89.55
CEDRClassification	57.42	Robust04InstructionRetrieval	-2.41
CLSClusteringP2P.v2	42.68	RomaniBibleClustering	43.22
CSFDSKMovieReviewSentimentClassification	49.38	RuBQReranking	73.84
CTKFactsNLI	87.59	SCIDOCS	25.15
CataloniaTweetClassification	54.51	SIB200ClusteringS2S	41.74
Core17InstructionRetrieval	7.69	SICK-R	82.75
CovidRetrieval	79.13	SNLHierarchicalClusteringP2P	61.41
CyrillicTurkicLangClassification	95.30	STS12	81.55
CzechProductReviewSentimentClassification	68.16	STS13	89.89
DBpediaClassification	94.76	STS14	85.41
DalajClassification	50.47	STS15	90.44
DiaBlaBitextMining	87.23	STS17	88.58
EstonianValenceClassification	53.52	STS22.v2	71.69
FaroeseSTS	86.12	STSB	85.50
FilipinoShopeeReviewsClassification	48.45	STSBenchmark	89.08
FinParaSTS	28.60	STSES	81.75
FinancialPhrasebankClassification	88.64	ScalaClassification	51.85
FloresBitextMining	83.71	SemRel24STS	73.14
GermanSTSBenchmark	88.09	SentimentAnalysisHindi	76.06
GreekLegalCodeClassification	43.76	SinhalaNewsClassification	82.29
GujaratiNewsClassification	92.05	SiswatiNewsClassification	62.38
HALClusteringS2S.v2	32.00	SlovakMovieReviewSentimentClassification	90.35
HagridRetrieval	99.31	SpartQA	10.30
IN22GenBitextMining	93.75	SprintDuplicateQuestions	96.90
IndicCrosslingualSTS	62.87	StackExchangeClustering.v2	92.07
IndicGenBenchFloresBitextMining	96.77	StackOverflowQA	96.71
IndicLangClassification	87.69	StatcanDialogueDatasetRetrieRetrieval	51.11
IndonesianIdClickbaitClassification	67.00	SwahiliNewsClassification	66.05
IsiZuluNewsClassification	40.53	SwednClusteringP2P	45.84
ItaCaseholdClassification	73.30	SwissJudgementClassification	57.86
JSICK	84.99	T2Reranking	67.95
KorHateSpeechMLClassification	17.69	TERRa	63.92
KorSarcasmClassification	60.51	TRECCOVID	86.32
KurdishSentimentClassification	86.39	Tatoeba	81.97
LEMBPasskeyRetrieval	38.50	TempReasonL1	2.96
LegalBenchCorporateLobbying	95.98	ToxicConversationsClassification	88.75
MIRACLRetrievalHardNegatives	70.42	TswanaNewsClassification	53.37
MLQARetrieval	84.16	TweetTopicSingleClassification	71.11
MacedonianTweetSentimentClassification	71.83	TwitterHjerneRetrieval	98.02
MalteseNewsClassification	37.38	TwitterURLCorpus	87.05
MasakhaNEWSClassification	83.55	VoyageMMarcoReranking	66.73
MasakhaNEWSClusteringS2S	57.45	WebLINXCandidatesReranking	10.97
MassiveIntentClassification	81.92	WikiCitiesClustering	91.63
MedrxivClusteringP2P.v2	47.16	WikiClusteringP2P.v2	28.23
MultiEURLEXMultilabelClassification	5.28	WikipediaRerankingMultilingual	92.24
MultiHateClassification	72.47	WikipediaRetrievalMultilingual	94.20
NTREXBitextMining	93.64	WinoGrande	60.52
NepaliNewsClassification	98.14	XNLI	85.26
News21InstructionRetrieval	10.26	indonli	60.69

Table 9 | Full results of Gemini Embedding on MTEB(Multilingual).

Task Name	Performance
AmazonCounterfactualClassification	92.69
ArXivHierarchicalClusteringP2P	64.92
ArXivHierarchicalClusteringS2S	63.84
ArguAna	86.44
AskUbuntuDupQuestions	64.24
BIOSSES	88.97
Banking77Classification	94.27
BiorxivClusteringP2P.v2	53.86
CQADupstackGamingRetrieval	70.68
CQADupstackUnixRetrieval	53.69
ClimateFEVERHardNegatives	31.06
FEVERHardNegatives	88.98
FiQA2018	61.78
HotpotQAHardNegatives	87.01
ImdbClassification	94.98
MTOPDomainClassification	99.27
MassiveIntentClassification	88.46
MassiveScenarioClassification	92.08
MedrxivClusteringP2P.v2	47.16
MedrxivClusteringS2S.v2	45.01
MindSmallReranking	32.95
SCIDOCS	24.04
SICK-R	82.75
STS12	81.55
STS13	89.89
STS14	85.41
STS15	90.44
STS17	91.61
STS22.v2	68.37
STSBenchmark	89.08
SprintDuplicateQuestions	96.90
StackExchangeClustering.v2	92.07
StackExchangeClusteringP2P.v2	50.91
SummEvalSummarization.v2	38.28
TRECCOVID	86.32
Touche2020Retrieval.v3	52.39
ToxicConversationsClassification	88.75
TweetSentimentExtractionClassification	69.88
TwentyNewsgroupsClustering.v2	57.37
TwitterSemEval2015	79.17
TwitterURLCorpus	87.05

Task Name	Performance
AppsRetrieval	93.75
COIRCodeSearchNetRetrieval	81.06
CodeEditSearchRetrieval	81.61
CodeFeedbackMT	56.28
CodeFeedbackST	85.33
CodeSearchNetCCRetrieval	84.69
CodeSearchNetRetrieval	91.33
CodeTransOceanContest	89.53
CodeTransOceanDL	31.47
CosQA	50.24
StackOverflowQA	95.92
SyntheticText2SQL	69.96

Table 10 | Full results of Gemini Embedding on MTEB(Eng, v2) (left) and MTEB(Code) (right).

Language	Performance
ar	91.26
bn	94.08
fi	89.17
ja	86.31
ko	89.82
ru	88.61
te	93.70

Language	Performance
as	69.25
bho	66.38
brx	25.66
gbm	64.87
gom	65.54
gu	70.26
hi	69.06
hne	68.33
kn	69.54
mai	68.39
ml	70.82
mni	44.44
mr	68.77
mwr	66.49
or	65.77
pa	69.55
ps	61.90
sa	68.09
ta	68.57
ur	64.85

Table 11 | Full results of Gemini Embedding on XOR-Retrieve (left) and XTREME-UP (right).

10. Contributions and Acknowledgments

Acknowledgement