Llama-Embed-Nemotron-8B: A Universal Text Embedding Model for Multilingual and Cross-Lingual Tasks

Yauhen Babakhin, Radek Osmulski, Ronay Ak, Gabriel Moreira, Mengyao Xu,
Benedikt Schifferer, Bo Liu, Even Oldridge

NVIDIA[1]

Abstract

We introduce llama-embed-nemotron-8b, an open-weights text embedding model[2] that achieves state-of-the-art performance on the Multilingual Massive Text Embedding Benchmark (MMTEB) leaderboard as of October 21, 2025. While recent models show strong performance, their training data or methodologies are often not fully disclosed. We aim to address this by developing a fully open-source model, publicly releasing its weights and detailed ablation studies, and planning to share the curated training datasets. Our model demonstrates superior performance across all major embedding tasks – including retrieval, classification and semantic textual similarity (STS) – and excels in challenging multilingual scenarios, such as low-resource languages and cross-lingual setups. This state-of-the-art performance is driven by a novel data mix of 16.1 million query-document pairs, split between 7.7 million samples from public datasets and 8.4 million synthetically generated examples from various open-weight LLMs. One of our key contributions is a detailed ablation study analyzing core design choices, including a comparison of contrastive loss implementations, an evaluation of synthetic data generation (SDG) strategies, and the impact of model merging. The llama-embed-nemotron-8b is an instruction-aware model, supporting user-defined instructions to enhance performance for specific use-cases. This combination of top-tier performance, broad applicability, and user-driven flexibility enables it to serve as a universal text embedding solution.

1 Introduction

Dense text embedding models are a fundamental component of modern information retrieval. They are critical for a wide range of applications, including web search, question answering, semantic textual similarity, and recommendation engines. Their importance has been further amplified by the widespread adoption of Retrieval-Augmented Generation (RAG), which grounds Large Language Models (LLMs) in external context. Recent notable text embedding models include NV-Embed [1], NV-Retriever [2], Qwen3-Embedding [3], and Gemini Embedding [4]. These models achieve strong results on benchmarks like the Massive Text Embedding Benchmark (MTEB) [5, 6], which comprehensively evaluate models across a broad range of tasks.

In parallel, the field has seen a significant shift towards multi-modal [7, 8, 9] and omni-modal [10] embedding models. This trend is driven by the need to handle real-world documents, such as PDFs or slides, which contain a mix of text, tables, and charts, as well as other rich modalities like audio and video. While multi-modal models address this trend, high-performance, text-only embedding models remain a relevant and efficient solution for a wide range of text-centric use cases. This includes a large volume of inherently text-native data, such as news articles, support tickets, and

[1]Correspondence to Yauhen Babakhin (ybabakhin@nvidia.com)

[2]We released the model at https://huggingface.co/nvidia/llama-embed-nemotron-8b

Table 1: Aggregated results for the MTEB(Multilingual, v2) split of the MTEB Leaderboard (as of October 21, 2025). Ranking on the Leaderboard is performed based on the Borda rank. Each task is treated as a preference voter, which gives votes to the models based on their relative performance on the task. The model with the highest number of votes across all tasks obtains the best rank.

Model	Borda Rank	Borda Votes	Mean (Task) 131 tasks	Mean (Task Type) 9 task types
llama-embed-nemotron-8b	**1.**	**39,573**	69.46	61.09
gemini-embedding-001	2.	39,368	68.37	59.59
Qwen3-Embedding-8B	3.	39,364	**70.58**	**61.69**
Qwen3-Embedding-4B	4.	39,099	69.45	60.86
Qwen3-Embedding-0.6B	5.	37,419	64.34	56.01
gte-Qwen2-7B-instruct	6.	37,167	62.51	55.93
Linq-Embed-Mistral	7.	37,149	61.47	54.14

legal documents, as well as popular pipelines where documents like scanned PDFs or invoices are first converted to text via Optical Character Recognition (OCR).

The central challenge in this domain remains the development of a truly "universal" text model that performs robustly across diverse tasks, domains, and, critically, multiple languages. To address this challenge, we introduce llama-embed-nemotron-8b, a new universal text embedding model. Our model establishes a new state-of-the-art, achieving the 1st place rank on the comprehensive Multilingual Massive Text Embedding Benchmark (MMTEB) leaderboard [6] (as of October 21, 2025). See aggregated results in Table 1.

This paper details the architecture, training methodology, and data mixing strategies that enable llama-embed-nemotron-8b to effectively unify text representation across a wide spectrum of languages and text embedding tasks.

2 Model

Our model, llama-embed-nemotron-8b, is a universal, instruction-tuned text embedding model designed to generate specialized embeddings for a wide range of tasks, including retrieval, classification, and STS. It has the ability to adapt its embedding outputs based on a task-specific instructional prefix.

We initialize the model using the weights and architecture of the Llama-3.1-8B model [11]. The base Llama-3.1 model is a decoder-only transformer that employs a causal attention mask, where each token can only attend to itself and previous tokens. We replace the causal attention mask in all transformer layers with a standard bi-directional attention (i.e., no masking). This allows every token in the input sequence to freely attend to all other tokens in the sequence, effectively converting the model into a bi-directional encoder. We unfreeze all the model weights and fine-tune Llama-3.1-8B end-to-end.

To produce a single, fixed-size embedding, the model processes the tokenized input sequence S and produces a sequence of hidden states $H \in \mathbb{R}^{L \times d_{model}}$ from its final transformer layer, where L is the sequence length and d_{model} is the model's hidden dimension (4096). We then apply global average pooling over the sequence dimension of these final hidden states to obtain the final embedding vector v.

The model's specialization for diverse task types is guided by a textual instruction provided in the input. All inputs T are formatted using a specific template:

```
Input = f"Instruct: {task_instruction}\nQuery: {T}"
```

where $task\_instruction$ is a string that tells the model to produce an embedding suitable for the target task or use-case.

While the core encoder model is shared, its application architecture varies depending on the task family:

- **For Retrieval Tasks:** We employ a bi-encoder architecture. The query and corpus (documents) are processed independently by the shared encoder. This projects them into a common embedding space. Query is using the appropriate instruction template, while no specific formatting is required for documents. At inference time, relevance is computed using cosine similarity, enabling efficient and scalable search across large corpora.
- **For STS & Classification Tasks:** The model functions as a uni-encoder. Each text is passed through the model with the appropriate instruction to generate its embedding. These embeddings are then directly used for the task, such as computing cosine similarity for STS or serving as features for a classifier.

This flexible, instruction-driven approach allows a single model to effectively handle the varied demands of all the MMTEB task types.

3 Training

3.1 Training Objective

We leverage contrastive learning to train the model, mapping inputs to a shared embedding space. The core objective is to maximize the similarity between related items and minimize it for unrelated ones. While the specific definition of the training triplet – an anchor query (q), a positive document (d^+), and a set of negative documents (D_N) – is adapted depending on the specific problem type, the training is governed by the InfoNCE contrastive loss [12]. The formal objective is:

$$\mathcal{L}(q, d^+, D_N) = -\log \frac{\exp(\text{sim}(q, d^+)/\tau)}{\sum_{d_i \in \{d^+\} \cup D_N} \exp(\text{sim}(q, d_i)/\tau)} \tag{1}$$

where q is the embedding of an anchor, d^+ is the embedding of the positive item, and D_N denotes the set of negative items. $sim(\cdot)$ represents the cosine similarity function, and τ is the temperature hyperparameter.

While Equation 1 defines the core objective, the composition of the training triplet (q, d^+, D_N) is adapted for the different task types:

- **For Retrieval Tasks:** The inputs directly map to a <query, positive document, negative documents> triplet. The query q is formatted with the retrieval instruction (as described in Section 2), while documents (d^+ and D_N) do not require any prefix instructions. Some of the popular components which are frequently added to the D_N set are in-batch negatives [4] and same-tower negatives [3, 13]. For our model training, we do not utilize any extra negatives in D_N, apart from the mined hard negatives. Our hard negative mining process is described in Section 4.4.
- **For Classification Tasks:** The input text serves as the anchor q. The positive d^+ is the text of the correct label name. The negatives D_N are a set of random incorrect label names from the given classification task. The anchor is formatted with the specific classification instructions, while label names are processed without any modifications.
- **For STS Tasks:** These tasks are treated as symmetric. Given a positive pair of texts (T_A, T_B), we create a training instance: $q = T_A, d^+ = T_B$. In this case, D_N contains hard negative examples which are mined from the dataset's corpus. All texts (query, positive, and negatives) are processed using the same instruction prefix: "Retrieve semantically similar text.".

3.2 Training Stages

We start training from Llama-3.1-8B foundation model weights [11]. Llama-3.1-8B model is already pre-trained on a corpus of about 15T multilingual tokens. This makes it a strong base model for training multi-lingual text embedding models. We train the model in two stages described below. Detailed hyperparameters for each stage are provided in Appendix A.

Stage 1: Retrieval Pretraining. The goal of the first stage is to adapt Llama-3.1-8B LLM to both bi-directional attention and embedding model setup. In this stage we use only retrieval data where queries and documents are based on the Web corpus. We use only a single hard-negative for each <query, document> pair which is mined from the same Web corpus. This stage constitutes about 70% of the overall data mix.

Stage 2: Fine-Tuning. In the second stage, we fine-tune the model using high-quality datasets for various problem types: retrieval, classification, STS, and bitext mining. The goal of this stage is to train a well-rounded model that works across different tasks. This stage constitutes the other 30% of the overall data mix.

3.3 Model Merging

We train multiple models using the two-stage approach above, and then apply model merging across resulting checkpoints. Model merging involves combining the parameters of multiple models, and is a well-established technique [14, 15]. Recently, this approach has been popularized for improving the robustness and generalization of embedding models, such as Qwen3-Embedding [3], Gemini Embedding [4], and EmbeddingGemma [16].

The core idea of this method is to average the parameters obtained from the individual model runs. There are different strategies for selecting the individual checkpoints. These include averaging checkpoints from different steps within the same training run [14], or combining models from multiple, distinct training runs that may use different hyperparameters or intentional data variations [15].

Our final model is an average of six diverse individual checkpoints. We achieved this diversity by varying the data mixes and model hyperparameters across training runs. This final model produces the best evaluation results compared to individual checkpoints, with no inference time increase. We provide more details about the results in the ablation studies (Section 6.4).

4 Datasets

This section provides details about our data curation process. We plan to release our data mix, it will be available as a part of our HuggingFace collection[3]. The whole data mix contains a total of 16 million <query, document> pairs. Its overview is presented in Table 2.

4.1 Pretraining Data Mix

For pretraining data, we relied on the NVIDIA's Nemotron-CC-v2 dataset [17]. We employed two strategies to create our final pretraining set, which consists of approximately 11.8M <query, document> pairs.

[3]Nemotron RAG collection: https://huggingface.co/collections/nvidia/nemotron-rag

Table 2: Overview of the training data mix, detailing the number of <query, document> pairs for pretraining and fine-tuning, segmented by non-synthetic and synthetic data sources.

Training Stage	Non-Synthetic Data	Synthetic Data
Pretraining	5.0M	6.8M
Fine-tuning	2.7M	1.6M
Total	7.7M	8.4M

Utilize existing questions from Nemotron-CC-v2. In this strategy, we utilized the Diverse-QA split of the Nemotron-CC-v2 dataset. We extracted the existing questions and their corresponding positive documents from this split. We then mined hard negatives from a pool of 1M document chunks sampled from the same Diverse-QA corpus. This approach yielded 5.0M training pairs.

Generate new questions for Nemotron-CC-v2 corpus. For our second strategy, we generated new synthetic queries. We took the existing documents from the Diverse-QA split and generated our own synthetic questions for them, creating a new set of <query, positive document> pairs. Subsequently, we mined hard negatives for these new pairs in the same manner as the first strategy. This approach contributed the remaining 6.8M training pairs.

4.2 Fine-tuning Data Mix

For the fine-tuning data mix, we started with a mix introduced in NV-Embed [1]. The MTEB Leaderboard [6] reports a zero-shot percentage for each model, which indicates whether any of the benchmark's train, validation, or test splits were used during a model's training phase. This metric is designed to ensure that the MTEB evaluation datasets remain out-of-domain for the models being tested. Therefore, to preserve the integrity of our zero-shot evaluation, we removed the majority of data originating from both MTEB(Multilingual, v2) and MTEB(eng, v2) splits of the MTEB.

As reported in Table 2, our fine-tuning mix consists of two parts: non-synthetic and synthetic data. For non-synthetic part, we utilized well-known public datasets, like MIRACL [18], HotpotQA [19], MS MARCO[20], Natural Questions [21], SQuAD [22], and more. The next section describes the synthetic part. The full list of fine-tuning datasets, together with a number of samples is presented in Appendix B.

4.3 Synthetic Data Generation

To enhance the diversity of our datasets, we employed a comprehensive Synthetic Data Generation (SDG) strategy, with a specific focus on multi-lingual and cross-lingual data. We applied SDG for creating datasets across primary task types: retrieval, classification, STS, and bitext mining.

Our methodology relied on two main strategies, inspired by the recent state-of-the-art embedding models. The first strategy, similar to [23] approach, involved the end-to-end generation of complete <query, positive, negatives> text triplets from scratch. The second strategy, inspired by [3] and [4], leveraged a seed corpus. We first sampled a positive document from the corpus, generated a corresponding query, and subsequently mined hard-negatives from the same corpus.

Additionally, we expanded our multi-lingual data by translating several existing high-quality datasets into various target languages. For all SDG and translation tasks, we utilized a diverse mix of powerful, open-weights LLMs. This list includes: gpt-oss-20b and gpt-oss-120b [24], Mixtral-8x22B-Instruct-v0.1 [25], Llama-3.3-70B-Instruct [11], Llama-4-Scout-17B-16E-Instruct and Llama-4-Maverick-17B-128E-Instruct [26]. We further explore the quality of synthetic datasets produced by different LLMs in our ablation studies (Section 6.2).

4.4 Hard Negative Mining

To improve the effectiveness of contrastive learning, we incorporated the *top-k with percentage to positive threshold* strategy from NV-Retriever [2] for hard negative mining. For this process, we set the threshold at 0.95. This means that for a given query, we selected the top K most relevant negative samples whose similarity to the query is less than 95% of the query–positive similarity score. This approach encourages the model to learn from challenging negatives while simultaneously filtering out potential false negatives that have high similarity scores. We sourced hard negatives using a combination of two embedding models: e5-mistral-7b-instruct [23] and Qwen3-Embedding-8B [3].

5 Results

We evaluate our model on the Multilingual split of the MTEB benchmark [5], which was introduced in Massive Multilingual Text Embedding Benchmark (MMTEB) [6]. This is the most extensive benchmark for multilingual and cross-lingual text embedding models, comprising 131 diverse tasks across 9 task types and 250+ languages (both high- and low-resource). The task types include Bitext Mining, Classification, Clustering, Instruction Reranking, Multilabel Classification, Pair Classification, Reranking, Retrieval, and STS.

Our model is instruction-aware, supporting custom instructions to optimize performance for specific use cases. For the MMTEB evaluations, we firstly took task-specific instructions directly from the MMTEB evaluation datasets, and adapted instructions from the Qwen3-Embedding evaluation repository [4] for tasks without default instructions.

We present a detailed comparison of our model against the other top-10 models on the MMTEB Leaderboard (as of October 21, 2025) in Table 3. Other models in the comparison include gemini-embedding-001 [4], Qwen3-Embedding family of models [3], gte-Qwen2-7B-instruct [27], Linq-Embed-Mistral [28], multilingual-e5-large-instruct [29], embeddinggemma-300m [16] and SFR-Embedding-Mistral [30].

Our model achieves state-of-the-art performance, securing the Rank 1 position with 39,573 Borda votes. This represents a significant lead of over 200 votes compared to the 2nd place (gemini-embedding-001) and 3rd place (Qwen3-Embedding-8B) models.

Ranking on the official MMTEB Leaderboard is determined by the Borda count method [6]. Each task is treated as a preference voter, which gives votes to the models based on their relative performance on the task. The best model obtains the highest number of votes. The model with the highest number of votes across all tasks obtains the highest rank. The Borda count method has been shown to be more robust for comparing NLP systems [31]. While Qwen3-Embedding-8B, achieves a higher "Mean (Task)" score (70.58 vs. our 69.46), this mean metric can be sensitive to outlier performance on a small subset of benchmarks. High scores on a few tasks can inflate the overall average without necessarily indicating consistent generalization. In contrast, the Borda rank is designed to reward broad and consistent generalization across the entire spectrum of 131 tasks, rather than strong performance in a limited number of areas.

6 Ablation Study

In this section, we ablate design choices that helped the development of the llama-embed-nemotron-8b model. Due to the computational cost of ablating at the 8B scale, all studies presented below (unless mentioned otherwise) were conducted on a 1B model fine-tuned from Llama-3.2-1B [11] on our fine-tuning data mix. These smaller-scale experiments allowed us to validate our training decisions before scaling up the experiments.

[4] Qwen3-Embedding GitHub repository: https://github.com/QwenLM/Qwen3-Embedding

Table 3: Evaluation results of top leaderboard models on MTEB(Multilingual, v2) Leaderboard (as of October 21, 2025). Ranking on the official Leaderboard is determined by the Borda rank. Mean (Task) column is the average of scores across 131 individual tasks, while Mean (Type) is the average across 9 problem types.

Model	Borda Rank	Borda Votes	Mean (Task)	Mean (Type)	Bitext Mining	Class.	Clust.	Instr. Rerank.	Multi. Class.	Pair Class.	Rerank.	Retrieval	STS
llama-embed-nemotron-8b	**1.**	**39,573**	69.46	61.09	**81.72**	73.21	54.35	10.82	**29.86**	83.97	**67.78**	68.69	79.41
gemini-embedding-001	2.	39,368	68.37	59.59	79.28	71.82	54.59	5.18	29.16	83.63	65.58	67.71	79.40
Qwen3-Embedding-8B	3.	39,364	**70.58**	**61.69**	80.89	**74.00**	**57.65**	10.06	28.66	**86.40**	65.63	**70.88**	**81.08**
Qwen3-Embedding-4B	4.	39,100	69.45	60.86	79.36	72.33	57.15	**11.56**	26.77	85.05	65.08	69.60	80.86
Qwen3-Embedding-0.6B	5.	37,419	64.34	56.01	72.23	66.83	52.33	5.09	24.59	80.83	61.41	64.65	76.17
gte-Qwen2-7B-instruct	6.	37,167	62.51	55.93	73.92	61.55	52.77	4.94	25.48	85.13	65.55	60.08	73.98
Linq-Embed-Mistral	7.	37,149	61.47	54.14	70.34	62.24	50.60	0.94	24.77	80.43	64.37	58.69	74.86
multilingual-e5-large-instruct	8.	36,921	63.22	55.08	80.13	64.94	50.75	-0.40	22.91	80.86	62.61	57.12	76.81
embeddinggemma-300m	9.	36,728	61.15	54.31	64.40	60.90	51.17	5.61	24.82	81.40	63.25	62.49	74.73
SFR-Embedding-Mistral	10.	36,579	60.90	53.92	70.00	60.02	51.84	0.16	24.55	80.29	64.19	59.44	74.79

Table 4: Comparison of different InfoNCE loss implementations on the MMTEB Leaderboard.

Loss	Borda Votes	Mean (Task)	Mean (Type)	Bitext Mining	Class.	Clust.	Instr. Rerank.	Multi. Class.	Pair Class.	Rerank.	Retrieval	STS
Gecko	37,903	63.45	55.86	72.58	65.12	52.29	5.28	25.46	80.68	64.21	**61.20**	75.87
Qwen3-Embedding	36,835	62.14	55.49	**73.50**	60.60	**54.82**	4.97	25.22	80.63	64.11	60.12	75.41
Gemini Embedding	38,135	63.83	55.90	73.13	66.28	53.04	4.76	**24.70**	80.74	64.26	60.28	75.96
Ours (HNs Only)	**38,225**	**64.03**	**56.04**	72.94	**66.99**	52.27	**5.69**	24.60	**80.77**	**64.44**	60.66	**75.99**

6.1 Contrastive Loss Formulations

In this analysis, we compare our InfoNCE loss implementation to other prominent formulations. These approaches primarily differ in the composition of negative samples used in the loss denominator.

- Gecko Model [13] implementation contrasts a <query, positive passage> pair against a comprehensive set of negatives: (1) a single hard negative, (2) other positive passages from different queries in the batch, and (3) other queries in the batch. This third category is termed "same-tower negatives" [32], which are noted as being beneficial for symmetric text embedding tasks (e.g., semantic similarity).
- Qwen3-Embedding family of models [3] uses same-tower negatives not only for the queries, but also for in-batch positive and negative documents.
- Gemini Embedding [4] explicitly omits the same-tower negatives from the loss to avoid potential false negatives. The loss denominator is thus limited to only a single hard negative and other positive passages in the batch.
- **Ours:** In contrast, our approach simplifies the loss denominator to include only hard negative documents (HNs) (one in pretraining, four in fine-tuning). This formulation omits all the in-batch negatives and same-tower negatives.

To have a fair comparison, we have fixed all the hyperaparameters, and only tuned a learning rate for each loss separately. As shown in Table 4, all approaches achieve similar performance. This suggests that the inclusion of in-batch negatives or same-tower negatives provides minimal-to-no significant benefit over our simpler approach. Our implementation, which relies only on hard negatives, achieves the highest number of Borda votes (38,225) and wins the most individual task types.

6.2 Choice of LLM for Synthetic Data Generation

For training llama-embed-nemotron-8b we relied on multiple open-weights LLMs to generate synthetic data for various problem types: retrieval, classification, STS, and bitext mining. This analysis focuses on evaluating LLMs for the task of generating classification datasets. Following [1, 23], we created synthetic examples in 2 steps:

- Step 1. Prompt LLM to generate a list of potential classification tasks. These tasks are also used as instructions in our instruction-aware training.
- Step 2. Given the classification task, prompt LLM to generate (a) a text sample, (b) the correct label, and (c) a list of plausible but incorrect labels (misleading labels). We use the correct label name as a positive and the misleading label names as negatives.

Our baseline is a model that was trained without any synthetic classification datasets. We compare it against generating 100k synthetic samples using each of the LLMs, and training separate embedding models with extra 100k examples in the data mix. For each LLM we follow Step 1 and Step 2 described above, and generate data only in English language. LLMs being evaluated include gpt-oss-20b and gpt-oss-120b [24], Mixtral-8x22B-Instruct-v0.1 [25], Llama-3.3-70B-Instruct [11], Llama-4-Scout-17B-16E-Instruct and Llama-4-Maverick-17B-128E-Instruct [26].

By comparing performance on individual MMTEB evaluation datasets, we observed that different LLMs for SDG excel in different domains/languages. Therefore, we also compare the performance of each individual LLM with another mix of 100k synthetic samples, which mixes examples from all the LLMs with equal weights (i.e., $\approx$16.7k samples from each of the 6 models). See our full results in Table 5.

We compare results on multiple task types including classification, multilabel classification and clustering, which is also closely related to the classification. Results suggest that the largest LLM is not necessarily the best model for SDG, as gpt-oss-20b performs very well, while being the smallest model. But the best results are achieved by using a data mix compiled from all the LLMs. These findings suggest that diversity of synthetic data is more important than single-model quality. One of the reasons for such behavior might be that Mix approach has a more diverse tasks list from the Step 1 compared to individual LLMs.

This mixing approach for SDG is used in both our pretraining and fine-tuning datasets, and across all the problem types. We also extend this principle by using cross-model SDG, where Step 1 classification tasks are generated with one model, while Step 2 actual samples are generated with another model.

To conclude this analysis, we compare a baseline without synthetic classification data to the best SDG approach. We can see how only 100k synthetic classification examples show an improvement of +464 Borda votes (37,812 vs 37,348) and +0.94 in Mean points (62.89 vs 61.95). This demonstrates the efficacy of synthetic data. The next ablation explores how good synthetic data is, compared to in-domain classification datasets.

6.3 Impact of Synthetic vs. In-Domain Data

The MTEB leaderboard [6] tracks whether models use in-domain data in their data mixes (e.g., the training split of an evaluation dataset). This "zero-shot" percentage is tracked because in-domain data can significantly inflate performance on a specific evaluation dataset, making comparisons difficult.

This ablation study quantifies the gap between our synthetic data and in-domain data. The goal is to measure the effectiveness of our synthetic mix in closing the performance gap to a

Table 5: Comparison of different LLMs for generating synthetic classification datasets.

Model for SDG	Number of Parameters	Borda Votes	Mean (Task)	Class.	Clust.	Multi. Class.
No synthetic data	-	37,348	61.95	62.16	49.94	22.40
gpt-oss-20b	21B (3.6B active)	37,732	62.54	63.71	50.45	23.21
gpt-oss-120b	117B (5.1B active)	37,594	62.38	63.12	50.77	22.47
Mixtral-8x22B-Instruct-v0.1	141B (39B active)	37,797	62.64	63.67	50.89	22.81
Llama-3.3-70B-Instruct	70B	37,623	62.49	63.15	50.80	22.85
Llama-4-Scout-17B-16E-Instruct	109B (17B active)	37,595	62.21	63.02	50.57	22.70
Llama-4-Maverick-17B-128E-Instruct	400B (17B active)	37,643	62.36	63.20	50.51	22.41
Mix from all models	-	**37,812**	**62.89**	**64.39**	**50.95**	**23.37**

Table 6: Comparison of synthetic datasets against in-domain data.

Data	Amazon	Czech	Greek	Estonian	TweetTopic
Baseline (no synthetic data)	73.06	64.68	38.21	38.80	72.85
+1M synthetic samples	82.64	66.58	42.60	49.36	78.00
+75k in-domain samples	**90.51**	**71.73**	**60.51**	**57.20**	**80.12**

model trained on in-domain data. For this analysis, we randomly selected five classification evaluation datasets from MTEB(Multilingual, v2), namely AmazonCounterfactualClassification [33], CzechProductReviewSentimentClassification [34], GreekLegalCodeClassification [35], EstonianValenceClassification [36], and TweetTopicSingleClassification [37].

The baseline model is a model trained without any synthetic classification datasets. It is compared to two other models. Crucially, the model trained on in-domain data described below was prepared solely for this ablation study to serve as a comparative benchmark. This in-domain data was not used in our final llama-embed-nemotron-8b model submitted to the MTEB leaderboard.

- First model is trained on about 1M synthetic classification samples generated with the approach described in Section 6.2.
- Second model is trained on train splits of all the five datasets being evaluated: AmazonCounterfactualClassification (17.7k observations), CzechProductReviewSentimentClassification (24.0k), GreekLegalCodeClassification (28.5k), EstonianValenceClassification (3.3k) and TweetTopicSingleClassification (1.5k).

Results are presented in Table 6. The model trained on our synthetic classification data consistently outperforms baseline model across all five tasks. However, model trained on in-domain data achieves the highest scores, substantially outperforming the synthetic data mix. Notably, even a very small amount of in-domain data provides a powerful signal, with 1.5k train samples from TweetTopicSingleClassification dataset surpassing about 1M synthetic samples.

This shows that while our synthetic data allows to improve general performance on classification benchmarks, it is not a complete substitute for acquiring even small amounts of high-quality, in-domain data.

6.4 Model Merging

In Section 3.3, we discussed a model merging technique that enhances model's generalizability at no additional inference costs. This ablation quantifies the impact of this technique by comparing

Table 7: Evaluation results of individual checkpoints and the final llama-embed-nemotron-8b model on the MTEB(Multilingual, v2) Leaderboard. llama-embed-nemotron-8b is an average of six individual models listed in the table.

Model	Borda Votes	Mean (Task)	Mean (Type)	Bitext Mining	Class.	Clust.	Instr. Rerank.	Multi. Class.	Pair Class.	Rerank.	Retrieval	STS
Individual model 1	39,167	67.27	59.36	78.45	70.17	54.11	10.28	27.86	83.25	66.84	64.99	78.27
Individual model 2	39,265	67.99	59.78	78.60	71.83	53.79	10.42	28.59	83.86	66.85	65.77	78.33
Individual model 3	39,336	68.00	59.71	79.33	71.78	54.27	9.14	28.64	83.76	66.83	64.86	78.82
Individual model 4	39,401	68.36	60.18	79.52	71.89	**54.91**	10.32	29.20	83.82	67.04	66.15	78.76
Individual model 5	39,435	68.38	60.05	79.37	71.87	54.12	9.48	28.72	83.91	66.85	67.33	78.83
Individual model 6	39,454	68.62	60.37	79.56	72.37	54.34	10.80	29.71	83.60	66.91	66.99	79.08
llama-embed-nemotron-8b	**39,573**	**69.46**	**61.09**	**81.72**	**73.21**	54.35	**10.82**	**29.86**	**83.97**	**67.78**	**68.69**	**79.41**

the merged model against its individual component checkpoints. Our final llama-embed-nemotron-8b model is an average of six diverse individual models, weighted equally. This diversity stems from varying data mixes and hyperparameter sets used during training.

In Table 7, we present the MTEB(Multilingual, v2) results for the individual checkpoints and the final merged model. Notably, our best individual model ("Model 6") would already achieve SOTA performance on the MMTEB Leaderboard, securing 39,454 Borda votes (as of October 21, 2025). However, model merging yields a significantly stronger model, improving the score by +119 Borda votes (39,573 vs 39,454) and mean by +0.84 (69.46 vs 68.62) over "Model 6".

A key observation is that the individual models specialize in different task types. For instance, "Model 4" specializes in clustering and reranking; "Model 5" – in pair classification and retrieval; "Model 6" – in classification and STS. Merging all six checkpoints together creates a robust cumulative ensemble that aggregates these complementary strengths. This results in the strongest overall model, which achieves the top score in almost all problem types (apart from clustering).

7 Conclusion

In this work, we introduced llama-embed-nemotron-8b, a new open-weights universal text embedding model. Our model achieves state-of-the-art performance, securing the #1 position on the MMTEB leaderboard as of October 21, 2025. It demonstrates superior generalization according to the Borda count method, outperforming other top models across a diverse set of 131 tasks and 250+ languages.

This result is driven by a combination of a strong Llama-3.1-8B foundational model converted to a bi-directional encoder, a novel 16M-pair training data mix, and a robust training methodology. Our ablation studies revealed that using a mix of diverse open-weights LLMs for synthetic data generation yields more robust results than using any single LLM, emphasizing the importance of data diversity.

By releasing the weights of the llama-embed-nemotron-8b model, we provide a powerful, instruction-aware tool for a wide range of applications, including retrieval, classification, and STS. We also plan to release our curated data mix to facilitate future research and development in robust, multilingual text embeddings.

References

[1] 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, 2025.

[2] Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*, 2024.

[3] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[4] 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: Generalizable embeddings from gemini, 2025.

[5] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[6] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. Mmteb: Massive multilingual text embedding benchmark, 2025.

[7] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models, 2024.

[8] Mengyao Xu, Gabriel Moreira, Ronay Ak, Radek Osmulski, Yauhen Babakhin, Zhiding Yu, Benedikt Schifferer, and Even Oldridge. Llama nemoretriever colembed: Top-performing text-image retrieval model. *arXiv preprint arXiv:2507.05513*, 2025.

[9] Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Bo Wang, Sedigheh Eslami, Scott Martens, Maximilian Werk, Nan Wang, and Han Xiao. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval, 2025.

[10] Mengyao Xu, Wenfei Zhou, Yauhen Babakhin, Gabriel Moreira, Ronay Ak, Radek Osmulski, Bo Liu, Even Oldridge, and Benedikt Schifferer. Omni-embed-nemotron: A unified multimodal retrieval model for text, image, audio, and video, 2025.

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani,

Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam,

Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[13] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models, 2024.

[14] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2019.

[15] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022.

[16] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. Embeddinggemma: Powerful and lightweight text representations, 2025.

[17] Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, Alex Kondratenko, Alex Shaposhnikov, Alexander Bukharin, Ali Taghibakhshi, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amy Shen, Andrew Tao, Ann Guan, Anna Shors, Anubhav Mandarwal, Arham Mehta, Arun Venkatesan, Ashton Sharabiani, Ashwath Aithal, Ashwin Poojary, Ayush Dattagupta, Balaram Buddharaju, Banghua Zhu, Barnaby Simkin, Bilal Kartal, Bita Darvish Rouhani, Bobby Chen, Boris Ginsburg, Brandon Norick, Brian Yu, Bryan Catanzaro, Charles Wang, Charlie Truong, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christian Munley, Christopher Parisien, Dan Su, Daniel Afrimi, Daniel Korzekwa, Daniel Rohrer, Daria Gitman, David Mosallanezhad, Deepak Narayanan, Dima Rekesh, Dina Yared, Dmytro Pykhtar, Dong Ahn, Duncan Riach, Eileen Long, Elliott Ning, Eric Chung,

Erick Galinkin, Evelina Bakhturina, Gargi Prasad, Gerald Shen, Haifeng Qian, Haim Elisha, Harsh Sharma, Hayley Ross, Helen Ngo, Herman Sahota, Hexin Wang, Hoo Chang Shin, Hua Huang, Iain Cunningham, Igor Gitman, Ivan Moshkov, Jaehun Jung, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jian Zhang, Jiaqi Zeng, Jimmy Zhang, Jinze Xue, Jocelyn Huang, Joey Conway, John Kamalu, Jonathan Cohen, Joseph Jennings, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kezhi Kong, Krzysztof Pawelec, Kumar Anik, Kunlun Li, Kushan Ahmadian, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Luis Vega, Maer Rodrigues de Melo, Makesh Narsimhan Sreedhar, Marcin Chochowski, Mark Cai, Markus Kliegl, Marta Stepniewska-Dziubinska, Matvei Novikov, Mehrzad Samadi, Meredith Price, Meriem Boubdir, Michael Boone, Michael Evans, Michal Bien, Michal Zawalski, Miguel Martinez, Mike Chrzanowski, Mohammad Shoeybi, Mostofa Patwary, Namit Dhameja, Nave Assaf, Negar Habibi, Nidhi Bhatia, Nikki Pope, Nima Tajbakhsh, Nirmal Kumar Juluru, Oleg Rybakov, Oleksii Hrinchuk, Oleksii Kuchaiev, Oluwatobi Olabiyi, Pablo Ribalta, Padmavathy Subramanian, Parth Chadha, Pavlo Molchanov, Peter Dykas, Peter Jin, Piotr Bialecki, Piotr Januszewski, Pradeep Thalasta, Prashant Gaikwad, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Rabeeh Karimi Mahabadi, Rajen Patel, Ran El-Yaniv, Ranjit Rajan, Ria Cheruvu, Rima Shahbazyan, Ritika Borkar, Ritu Gala, Roger Waleffe, Ruoxi Zhang, Russell J. Hewett, Ryan Prenger, Sahil Jain, Samuel Kriman, Sanjeev Satheesh, Saori Kaji, Sarah Yurick, Saurav Muralidharan, Sean Narenthiran, Seonmyeong Bak, Sepehr Sameni, Seungju Han, Shanmugam Ramasamy, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shizhe Diao, Shreya Gopal, Shrimai Prabhumoye, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Siddhartha Jain, Somshubra Majumdar, Soumye Singhal, Stefania Alborghetti, Syeda Nahida Akter, Terry Kong, Tim Moon, Tomasz Hliwiak, Tomer Asida, Tony Wang, Tugrul Konuk, Twinkle Vashishth, Tyler Poon, Udi Karpas, Vahid Noroozi, Venkat Srinivasan, Vijay Korthikanti, Vikram Fugro, Vineeth Kalluru, Vitaly Kurin, Vitaly Lavrukhin, Wasi Uddin Ahmad, Wei Du, Wonmin Byeon, Ximing Lu, Xin Dong, Yashaswi Karnati, Yejin Choi, Yian Zhang, Ying Lin, Yonggan Fu, Yoshi Suhara, Zhen Dong, Zhiyu Li, Zhongbo Zhu, and Zijia Chen. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model, 2025.

[18] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023.

[19] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

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

[21] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[23] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models, 2024.

[24] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025.

[25] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[26] Meta AI. The llama 4 herd: The beginning of a new era of natively multimodal intelligence. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`, 2025.

[27] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[28] Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Minkyung Cho Yejin Kim, Jy yong Sohn, and Chanyeol Choi. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog, 2024.

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

[30] Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog, 2024.

[31] Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stephan Clemencon. What are the best systems? new perspectives on nlp benchmarking, 2022.

[32] Fedor Moiseev, Gustavo Hernandez Abrego, Peter Dornbach, Imed Zitouni, Enrique Alfonseca, and Zhe Dong. Samtone: Improving contrastive loss for dual encoder retrieval models with same tower negatives, 2023.

[33] James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. I wish I would have loved this one, but I didn't – a multilingual dataset for counterfactual detection in product review. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7092–7108, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[34] Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. Sentiment analysis in Czech social media using supervised machine learning. In Alexandra Balahur, Erik van der Goot, and Andres Montoyo, editors, *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 65–74, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[35] Christos Papaloukas, Ilias Chalkidis, Konstantinos Athinaios, Despina-Athanasia Pantazi, and Manolis Koubarakis. Multi-granular legal topic classification on greek legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 63–75, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[36] Hille Pajupuu, Jaan Pajupuu, Rene Altrov, and Kairi Tamuri. Estonian Valence Corpus / Eesti valentsikorpus. 11 2023.

[37] Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Leonardo Neves, Vitor Silva, and Francesco Barbieri. Twitter Topic Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[38] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA, 2013. Association for Computing Machinery.

[39] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28, 2015.

[40] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 990–998, New York, NY, USA, 2008. Association for Computing Machinery.

[41] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: Contextualized affect representations for emotion recognition. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[42] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[43] Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. Gooaq: Open question answering with diverse answer types, 2021.

[44] Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. Hover: A dataset for many-hop fact extraction and claim verification. *arXiv preprint arXiv:2011.03088*, 2020.

[45] Yuchen Zhuang, Aaron Trinh, Rushi Qiang, Haotian Sun, Chao Zhang, Hanjun Dai, and Bo Dai. Towards better instruction following retrieval models, 2025.

[46] Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. Mammoth2: Scaling instructions from the web. *Advances in Neural Information Processing Systems*, 2024.

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

[48] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. tydi: A multi-lingual benchmark for dense retrieval. *arXiv preprint arXiv:2108.08787*, 2021.

[49] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018.

[50] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. 2016.

[51] Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115, 2021.

[52] DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. Quora question pairs, 2017.

[53] Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. Tweac: Transformer with extendable qa agent classifiers. *arXiv preprint*, abs/2104.07081, 2021.

[54] David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. Scifact-open: Towards open-domain scientific claim verification. *arXiv preprint arXiv:2210.13777*, 2022.

[55] Stack-Exchange-Community. Stack exchange data dump, 2023.

[56] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems, 2020.

[57] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

[58] cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. Toxic comment classification challenge. `https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge`, 2017. Kaggle.

[59] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, page arXiv:1705.03551, 2017.

[60] Silly-Machine. Tupy-dataset (revision de6b18c), 2023.

A Implementation Details

This section details the hyperparameters and implementation specifics for our llama-embed-nemotron-8b model. Pretraining was conducted for 25.0 hours, and fine-tuning for 21.5 hours. Both stages utilized a cluster of 64 NVIDIA A100 80GB GPUs. The hyperparameters for both training stages are summarized in Table 8.

Table 8: Main hyperparameters for llama-embed-nemotron-8b training.

Hyperparameter	Pretraining	Fine-tuning
Peak learning rate	1e-5	2e-6
Batch size	2,048	128
Number of steps	5,773	33,668
Scheduler	Linear decay	Linear decay
Warm-up steps	100	100
Optimizer	AdamW	AdamW
Weight decay	0.01	0.01
Number of hard negatives	1	4
Temperature	0.02	0.02
Query max length	512	512
Document max length	512	512

B Fine-Tuning Data Mix for llama-embed-nemotron-8b

This section details the high-quality, curated data mix used in the fine-tuning stage of the llama-embed-nemotron-8b training. The complete dataset consists of 4.3 million samples, sourced from a diverse range of corpora, including multi-lingual and cross-lingual data.

The mix is composed of approximately 2.7 million non-synthetic samples from public sources and 1.6 million synthetic samples generated to target specific model capabilities. A detailed breakdown of the component datasets and their respective sample sizes is provided in Table 9.

Table 9: Component datasets and sample counts for the llama-embed-nemotron-8b fine-tuning data mix.

Dataset	Number of samples
AmazonReviews [38]	60,000
BioASQ [39]	2,495
DBLP-Citation-network V17 [40]	25,000
EmotionClassification [41]	13,046
FEVER [42]	140,085
GooAQ [43]	100,000
HotpotQA [19]	170,000
HoVer [44]	29,721
InF-IR [45]	77,518
MAmmoTH2 stackexchange [46]	317,180
MIRACL [18]	79,648
MLDR [47]	9,500
Mr.TyDi [48]	12,610
MS MARCO[20]	500,000
MultiNLI [49]	75,505
Natural Questions [21]	100,231
NFCorpus [50]	3,685
PAQ [51]	500,000
Quora question pairs [52]	101,762
RedditClustering [53]	90,000
SciFact [54]	919
SQuAD [22]	87,599
Stack Exchange [55]	80,001
SuperGLUE Textual Entailment [56]	3,094
Synthetic bitext mining data [57]	169,534
Synthetic classification data	1,044,212
Synthetic retrieval data	182,814
Synthetic STS data	239,997
Toxic Comment Classification [58]	16,800
TriviaQA [59]	73,346
TuPy [60]	3,200
Total	**4,309,502**