Nomic Embed: Training a Reproducible Long Context Text Embedder

Zach Nussbaum
zach@nomic.ai

John X. Morris
jack@nomic.ai
jxm3@cornell.edu

Brandon Duderstadt
brandon@nomic.ai

Andriy Mulyar
andriy@nomic.ai

Abstract

This technical report describes the training of nomic-embed-text-v1, the first fully reproducible, open-source, open-weights, open-data, 8192 context length English text embedding model that outperforms both OpenAI Ada-002 and OpenAI text-embedding-3-small on short and long-context tasks. We release the training code and model weights under an Apache 2 license. In contrast with other open-source models, we release a training data loader with 235 million curated text pairs that allows for the full replication of nomic-embed-text-v1. You can find code and data to replicate the model at https://github.com/nomic-ai/contrastors.

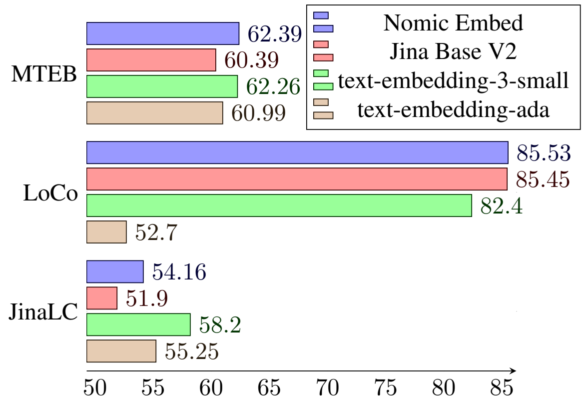

Figure 1: **Text Embedding Model Benchmarks.** Aggregate performance of nomic-embed-text-v1, OpenAI text-embedding-ada, OpenAI text-embedding-3-small and jina-embedding-base-v2 on short and long context benchmarks. Nomic Embed is the only fully auditable long-context model that exceeds OpenAI text-embedding-ada, OpenAI text-embedding-3-small, and Jina performance across both short and long context benchmarks. X-axis units vary per benchmark suite.

1 Introduction

Text embeddings are an integral component of modern NLP applications powering retrieval-augmented-generation (RAG) for LLMs and semantic search (Lewis et al., 2021a; Izacard et al., 2022b; Ram et al., 2023). These embeddings encode semantic information about sentences or documents as low-dimensional vectors that are used in downstream applications, such as clustering for data visualization, classification, and information retrieval.

The majority of the top open-source models on the MTEB benchmark (Muennighoff et al., 2023) are limited to context lengths of 512, such as E5 Wang et al. (2022), GTE Li et al. (2023), and BGE Xiao et al. (2023). This short context length reduces model utility in domains where overall document semantics are not localized to sentences or paragraphs. Most top embedding models with a context length longer than 2048 are closed-source, such as Voyage-lite-01-instruct Voyage (2023) and text-embedding-ada-002 Neelakantan et al. (2022).

The top two performing open-source long context embedding models are jina-embedding-v2-base-en Günther et al. (2024) and E5-Mistral-7b-instruct Wang et al. (2023b).

Unfortunately, jina-embedding-v2-base does not surpass OpenAI's text-embedding-ada-002 Neelakantan et al. (2022) (see Table 1). Further, E5-Mistral Wang et al. (2023b) is not feasible to use in many engineering applications due to the large inference requirements of a 7B parameter transformer, and is not recommended for use beyond 4096 tokens.

This report describes how we trained nomic-embed-text-v1, a 137M parameter, open-source, open-weights, open-data, 8192 sequence length model that surpasses OpenAI text-embedding-ada and text-embedding-3-small performance on both short and long context benchmarks (Table 1). We release the model weights and codebase under an Apache-2 license. We additionally release our curated training dataset to enable end-to-end auditability and replication of the model.

Model	Params	Seq	MTEB	LoCo	Jina LC	Weights	Code	Data
nomic-embed-text-v1	137M	8192	**62.39**	85.53	54.16	**Yes**	**Yes**	**Yes**
nomic-embed-text-v1-ablated	137M	8192	61.36	**86.89**	53.53	**Yes**	**Yes**	**Yes**
jina-embeddings-base-v2-en	137M	8192	60.39	85.45	51.90	**Yes**	No	No
text-embedding-ada-002	N/A	8192	60.99	52.70	55.25	No	No	No
text-embedding-3-small	N/A	8192	62.26	82.4	**58.21**	No	No	No
E5-Mistral-7b-instruct	7B	4096	66.6	87.8	N/A	Yes	No	No
text-embedding-3-large	N/A	8192	64.59	79.4	58.69	No	No	No

Table 1: Benchmarking nomic-embed-text-v1 against OpenAI models and other top long context open-source models. Nomic-embed-text-v1 is the only 100M parameter class open-source model that outperforms OpenAI text-embedding-ada and text-embedding-3-small on both short and long-context tasks. Nomic-embed-text-v1-ablated refers to the training setup described in Section 5.4, which omits the HotpotQA and FEVER data. 'Seq' refers to the context length of the model, and Jina LC is an average over tasks in the Jina Long Context benchmark.

2 Related Work

State-of-the-art text embedding models are trained by initializing a pre-trained transformer and then fine-tuning with a contrastive loss objective. Traditionally, fine-tuning involved leveraging labeled datasets such as MSMarco and SNLI (Bowman et al., 2015) to generate paired training data for the contrastive signal. Examples include SBERT (Reimers and Gurevych, 2019), SimCSE (Gao et al., 2022), and SGPT (Muennighoff, 2022). Recent systems such as E5 (Wang et al., 2022), GTE (Li et al., 2023), BGE (Xiao et al., 2023), InstructOR (Su et al., 2023a), and Jina (Günther et al., 2023, 2024) utilize a multi-stage regime in which a pretrained transformer is first contrastively fine-tuned using a large corpus of weakly paired data (e.g. Quora, Reddit Comments) and then additionally fine-tuned on small, higher quality labeled datasets such as MSMarco. The two-stage paradigm significantly improves model quality as weakly paired data is available in much greater quantity.

Evaluating text embedding models is challenging. The BEIR benchmark Thakur et al. (2021) evaluates dense retrievers on 15 zero-shot retrieval datasets. Early transformer-based text embedding models such as SBERT (Reimers and Gurevych, 2019) were only evaluated on semantic textual similarity (STS) datasets. More recently, MTEB Muennighoff et al. (2023) has become the de facto benchmark for quantitatively evaluating embedding models across many tasks, but has limited evaluations over long context lengths (>512 tokens). Jina Günther et al. (2024) developed a benchmark of four datasets specialized for long context evaluation. Additionally, the LoCo Saad-Falcon et al. (2024) benchmark was recently released to evaluate the performance of long context retrieval models.

As AI applications mature, auditability and compliance of models and their training data will be a critical component of safe model deployments in high-impact domains. For example, recent work by Anthropic on sleeper agents (Hubinger et al., 2024) demonstrates the risk of deploying models without end-to-end auditability. Top-performing text embedding models currently do not have auditable training stacks (i.e. a fully reproducible training pipeline with available weights, data, and code).

3 Training Data

In this section, we describe our data mix across each training stage. You can access the training data of nomic-embed-text-v1 by visiting the nomic-ai/contrastors code repository. You can explore a 5M sample of our contrastive training pairs at https://atlas.nomic.ai/map/nomic-text-embed-v1-5m-sample.

3.1 Masked Language Modeling Pretraining

Following (Devlin et al., 2019), we use BooksCorpus (Zhu et al., 2015) and a Wikipedia dump from 2023 to train a long-context BERT model, hereinafter called nomic-bert-2048. Each document from BooksCorpus and Wikipedia is tokenized using the bert-base-uncased tokenizer from Devlin et al. (2019) and packed to chunks of 2048 tokens. If a document is shorter than 2048 tokens, we append another document until it fits 2048 tokens. If a document is greater than 2048 tokens, we split it across multiple documents.

3.2 Unsupervised Contrastive Pretraining

Similar to Wang et al. (2022); Li et al. (2023); Xiao et al. (2023); Ni et al. (2022), we use large collections of publicly available data to form pairs. These datasets span various objectives and domains, from web retrieval to clustering of scientific articles. In total, we curated 470 million pairs across 29 datasets[1].

However, since these datasets can contain noisy examples, we employ consistency filtering (Günther et al., 2023; Wang et al., 2022).

Instead of using all-MiniLM-L6-v2 model[2], we use the gte-base model[3]. For each pair, described as $(query, document)$, we embed both the queries and documents of a 1 million point sub-sample of the dataset. For each query, we find the top-k (in this case 2) neighbors using cosine similarity. If $document$ is not in the top-k neighbors, we discard the example. After filtering, we end up with $\sim$235M pairs. The full dataset distribution can be seen in Table 5.

As the majority of these datasets are composed of sequences shorter than 2048 tokens we additionally curate long context datasets to allow for the learning of long-range dependencies. Namely, we use full Wikipedia articles paired with their titles as well as abstracts and full paper bodies from a single paper from S2ORC (Lo et al., 2020).

During training, we sample pairs from one data source at a time and fill the entire batch with samples from that single source to discourage the model from learning source-specific shortcuts.

3.3 Supervised Contrastive Fine-tuning

Supervised fine tuning is performed on MSMarco (Bajaj et al., 2018; Wang et al., 2023a), NQ (Karpukhin et al., 2020; Gao and Callan, 2021), NLI (Gao et al., 2022), HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), portions of MEDI (Su et al., 2023a), WikiAnswers (Fader et al., 2014), and Reddit[4]. For the datasets MSMarco, NQ, NLI, FEVER, and HotpotQA, we train over the released training sets from the BEIR benchmark (Thakur et al., 2021). For the retrieval datasets (MSMarco, NQ, HotpotQA, and Fever), we mine negatives, if not already mined using gte-baseLi et al. (2023). For every (q, d) pair, we get the top-k similar documents as hard negatives. For all other datasets, we randomly sample negatives in place of hard negatives as we found that mining negatives did not improve performance.

Similar to the unsupervised contrastive stage, we sample a dataset and fill a batch with all points from that chosen dataset.

4 Experimental Setup

4.1 Model Architecture

One of the main drawbacks of existing text encoders is their limited sequence length, which is predominately capped at 512 tokens. To train a long sequence length model, we first begin by adapting BERT so it can accommodate a long sequence length. In this work, we target an 8192 sequence length. To do so, we apply the following architecture changes and optimizations to BERT base (Devlin et al., 2019):

- Substituting absolute positional embeddings for rotary positional embeddings (Su et al., 2023b)
- Using SwiGLU activation instead of GeLU (Shazeer, 2020)
- Using Flash Attention (Dao et al., 2022)
- Setting Dropout to 0 (Geiping and Goldstein, 2022)
- Vocab size as a multiple of 64 (Portes et al., 2023) (Shoeybi et al., 2020)

resulting in a 137M parameter encoder.

We train all stages with a max sequence length of 2048 and employ Dynamic NTK interpolation at inference to scale to 8192 sequence length (Peng et al., 2023; emozilla, 2023). Additionally, we opt for SwiGLU versus GeGLU like proposed in (Portes et al., 2023) as runtime is roughly 25% faster for SwiGLU using the Flash Attention repository[5].

[1] https://huggingface.co/datasets/sentence-transformers/embedding-training-data

[2] all-MiniLM-L6-v2 model https://huggingface.co/thenlper/gte-base)

[3] gte-base model (https://huggingface.co/thenlper/gte-base)

[4] https://github.com/PolyAI-LDN/conversational-datasets/tree/master/reddit

[5] https://github.com/Dao-AILab/flash-attention/tree/main

Model	Bsz	Steps	Seq	Cola	SST2	MRPC	STSB	QQP	MNLI	QNLI	RTE	Avg
nomic-bert-2048	4k	100k	2k	0.50	0.93	0.88	0.90	0.92	0.86	0.92	0.82	0.84
MosaicBERT	4k	70k	2k	0.54	0.93	0.87	0.90	0.92	0.86	0.92	0.82	0.85
RobertaBase	8k	500k	512	0.64	0.95	0.90	0.91	0.92	0.88	0.93	0.79	0.86
JinaBERTBase	4k	100k	512	0.51	0.95	0.88	0.90	0.81	0.86	0.92	0.79	0.83
MosaicBERT	4k	178k	128	0.59	0.94	0.89	0.90	0.92	0.86	0.91	0.83	0.85

Table 2: GLUE Dev Set Results. Roberta numbers taken from Table 8 in (Liu et al., 2019). MosaicBert numbers taken from Table S1 in Portes et al. (2023) except for the 2048 model which we evaluated in the same manner as nomic-bert-2048. JinaBertBase Glue Test numbers reported in Table 2 from (Günther et al., 2024).

4.2 Masked Language Modeling

During training, we use a 30% masking rate instead of 15% following (Portes et al., 2023) and we remove the Next Sentence Prediction task (Liu et al., 2019). We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 5e-4 with $\beta_1 = 0.9$ $\beta_2 = 0.98$. We employ a linear warmup of 6% of the total training steps and a linear decay to 0. We use a global batch size of 4096 with gradient accumulation over 8 batches. We utilize DeepSpeed (Rajbhandari et al., 2020) stage 2 to fit bigger batches into memory. Additionally, we use bfloat16 precision for matrix multiplication and fp32 for gradient accumulation dtype. We disable gradient clipping (Liu et al., 2019) and set weight decay to 1e-5. We tried training with a learning rate of 1e-3, but found instabilities during training. We call our final model nomic-bert-2048 and also release its weights.

4.3 Unsupervised Contrastive Pretraining

Unsupervised contrastive pretraining aims to teach a model to distinguish the most similar documents from other irrelevant documents. To do so, we employ the InfoNCE contrastive loss (van den Oord et al., 2019). For a given batch $B = (q_0, d_0), (q_1, d_1), ..., (q_n, d_n)$, we minimize the loss function:

$$\mathcal{L}_C = -\frac{1}{n} \sum_i \log \frac{e^{s(q_i,d_i)/\tau}}{e^{s(q_i,d_i)/\tau} + \sum_{j\neq i}^{n} e^{s(q_i,d_j)/\tau}}$$

where $s(q, d)$ is the cosine similarity of (q, d)

We initialize the model for unsupervised contrastive training with the weights of nomic-bert-2048. We use a batch size of 16,384 so each batch has a large number of in-batch negatives. Our optimizations for the encoder architecture and training strategy centered around achieving this batch size. We use AdamW with a learning rate of 2e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 0.01. Gradient clipping is set to 1.0. We use an linear warmup schedule of 700 steps and an inverse square root decay schedule. We train with a max sequence length of 2048 for 1 full epoch over the data.

Due to GPU memory constraints, we could not fit the full model, optimizer, states, and data into memory. As a workaround, we employ GradCache (Luyu Gao and Callan, 2021) as well as mixed precision training (Micikevicius et al., 2018).

Finally, we use task specific prefixes to break the symmetry of the biencoder as in (Wang et al., 2022). Without prefixes, the model receives conflicting reward signal. Consider the case of determining which response is closest to the question ”What is the capital of France?”:

1. “What is the name of the capital city of France?
2. “Paris is the capital of France.”

A semantic similarity task would consider the first closest, while a question answering task would consider the second closest. Prefixes enable the model to distinguish between the behaviors specified by each of these tasks.

We use the following task-specific prefixes:

- `search_query`
- `search_document`
- `classification`
- `clustering`

inspired by Reimers et al. (2023). We first break prefixes into two categories: symmetric, where the query and document have a similar structure, and asymmetric, where the query is usually a single sentence and the document can be many sentences.

(Su et al., 2023a) The first two prefixes are used for retrieval tasks: where `search_query` is typically for the question and `search_document` is for the response. `classification` is used for STS-related tasks like rephrasals. `clustering` is used for tasks where to objective is to group semantically similar texts close together, like Arxiv title-abstract pairs. For symmetric tasks, the same prefix is appended to both the query and document.

4.4 Supervised Contrastive Fine-tuning

The last stage of training aims to boost performance by utilizing human-labeled datasets. Several papers including (Ni et al., 2021a,b; Wang et al., 2022; Li et al., 2023) have shown that fine-tuning on these datasets leads to improvements in downstream performance.

We adapt the paired contrastive loss to include hard negatives in each batch. We train for one epoch using seven hard negatives per pair and a batch size of 256. We employ a learning rate of 2e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 0.01. Gradient clipping is set to 1.0. We use a linear warmup schedule of 400 steps and a linear cooldown to 0 and train with prefixes as described above. We found that increasing the number of negatives above 7 to not meaningfully improve performance. We also found that training for multiple epochs hurts performance.

5 Results

We evaluate nomic-bert-2048 on the GLUE benchmark (Wang et al., 2019) and find that it is competitive with similarly sized and trained models. We evaluate nomic-embed-text-v1 on MTEB (Muennighoff et al., 2023), Jina's Long Context Benchmark (Günther et al., 2024), and LoCo (Saad-Falcon et al., 2024). nomic-embed-text-v1 exceeds text-embedding-ada-002 and jina-embeddings-v2-base-en. On the long context benchmarks, LoCo and Jina Long Context Benchmark, nomic-embed-text-v1 uniformly outperforms jina-embeddings-v2-base-en. nomic-embed-text-v1 outperforms text-embedding-ada-002 on LoCo and on two of four datasets in Jina's Long Context Benchmark.

5.1 nomic-bert-2048 GLUE Results

We evaluate nomic-bert-2048 on the GLUE benchmark (Wang et al., 2019) following the methodolgy presented in (Liu et al., 2019). The GLUE benchmark consists of 9 tasks, but we evaluate on 8 similar to (Liu et al., 2019).

For each task, we train for 10 epochs with batch sizes 16, 32 and learning rate 1e-5, 2e-5, 3e-5 with a linear warmup of 6% across 5 seeds. The median score per task at the end of the 10 epochs is presented in Table 2. Note we report accuracy for MRPC and QQP and Pearson for STSB [6]. We report our results in Table 2. Similar to (Liu et al., 2019), we initialize from an MNLI checkpoint for RTE, STSB, and MRPC.

MosaicBERT (Portes et al., 2023) performs slightly better but is trained for slightly longer and on C4 (Raffel et al., 2019). Across all tasks, nomic-bert-2048 scores similarly to MosaicBERT except on Cola. However, we used a longer sequence length model and in effect have seen more tokens during pretraining. JinaBERT also scores similarly, although they report test scores versus dev scores and is trained similarly to MosaicBERT.

5.2 MTEB Results

MTEB (Muennighoff et al., 2023) has become the standard benchmark for evaluating embedding models due to its diverse coverage of 8 tasks spanning 56 datasets. MTEB evaluated embedding models across Classification, Clustering, Pair Classification, Reranking, Retrieval, Semantic Textual Similarity, and Summarization. The MTEB score is a weighted average of the per-task scores.

5.3 Long Context Results

However, as noted in (Günther et al., 2024), MTEB has very few datasets that include long sequences. To evaluate nomic-embed-text-v1's performance on longer sequences, we consider two additional benchmarks: (Günther et al., 2024) Long Context Dataset as well as the LoCo benchmark from (Saad-Falcon et al., 2024).

5.3.1 JinaAI Long Context Benchmark

The Jina Long Context Benchmark (Günther et al., 2024) evaluates on 4 datasets across Retrieval and Clustering; namely, NarrativeQA (Günther et al., 2024), WikiCites [7], SciFact (Wadden et al., 2020),

[6] https://github.com/facebookresearch/fairseq/issues/1561#issuecomment-571729519

[7] https://huggingface.co/datasets/jinaai/cities_wiki_clustering

Table 3: Results on the MTEB benchmark (Muennighoff et al., 2023). The numbers are averaged for each category. Please refer to `https://huggingface.co/spaces/mteb/leaderboard` for the scores per dataset and the most up to date results.

Category →	Cls.	Clust.	PairCls.	Rerank	Retr.	STS	Summ.	Avg
Number of datasets →	12	11	3	4	15	10	1	56
Unsupervised Models								
Glove (Pennington et al., 2014)	57.3	27.7	70.9	43.3	21.6	61.9	28.9	42.0
SimCSE (Gao et al., 2022)	62.5	29.0	70.3	46.5	20.3	74.3	31.2	45.5
nomic-embed-text-v1$_{\texttt{unsup}}$	71.2	42.5	83.7	55.0	48.0	80.8	30.7	59.9
Supervised Models								
SimCSE$_{\text{bert-sup}}$ (Gao et al., 2022)	67.3	33.4	73.7	47.5	21.8	79.1	23.3	48.7
Contriever (Izacard et al., 2022a)	66.7	41.1	82.5	53.1	41.9	76.5	30.4	56.0
GTR$_{\texttt{xxl}}$ (Ni et al., 2021a)	67.4	42.4	86.1	56.7	48.5	78.4	30.6	59.0
Sentence-T5$_{\texttt{xxl}}$ (Ni et al., 2021b)	73.4	43.7	85.1	56.4	42.2	82.6	30.1	59.5
E5$_{\texttt{large-v2}}$ (Wang et al., 2022)	75.2	44.5	86.0	56.6	50.6	82.1	30.2	62.3
E5$_{\texttt{mistral}}$ (Wang et al., 2023b)	78.5	50.3	88.3	60.2	56.9	84.6	31.4	66.6
GTE$_{\texttt{base}}$ (Li et al., 2023)	73.0	46.2	84.6	58.6	51.1	82.3	31.2	62.4
GTE$_{\texttt{large}}$ (Li et al., 2023)	73.3	46.8	85.0	59.1	52.2	83.4	31.7	63.1
BGE$_{\texttt{base}}$ (Xiao et al., 2023)	75.5	45.8	86.6	58.9	53.3	82.4	31.1	63.6
BGE$_{\texttt{large}}$ (Xiao et al., 2023)	76.0	46.1	87.1	60.0	54.3	83.1	31.6	64.2
Jina$_{\texttt{v2}}$ (Günther et al., 2024)	73.5	41.7	85.4	57.0	47.9	80.7	31.6	60.4
`text-embedding-ada-002`	70.9	45.9	84.9	56.3	49.3	81.0	30.8	61.0
`text-embedding-3-small`	73.2	46.7	85.0	56.7	51.1	81.6	31.1	62.3
`text-embedding-3-large`	75.5	49.0	85.7	59.2	55.4	81.7	29.9	64.6
nomic-embed-text-v1-ablated	73.6	43.7	84.6	53.3	51.4	80.2	31.3	61.4
nomic-embed-text-v1	74.1	43.9	85.2	55.7	52.8	82.1	30.1	62.4

and BigPatent [8] (Sharma et al., 2019). Results are presented in Table 4. Similar to (Günther et al., 2024), we report the V-scores and NDCG@10 for the clustering and retrieval datasets respectively. Across sequence lengths and tasks, nomic-embed-text-v1 beats or ties jina-embeddings-v2-base on all datasets at 8k context length. Additionally, nomic-embed-text-v1 beats text-embedding-ada-002 on two of the four datasets. We also report similar results to (Günther et al., 2024) on WikiCitiesClustering that sequence length hurts performance, suggesting that longer sequence lengths are not necessary to perform well on the test.

5.3.2 LoCo Benchmark

The LoCo Benchmark consists of 5 retrieval datasets, 3 from (Shaham et al., 2022) and 2 from (Dasigi et al., 2021). The benchmark tests retrieval across meeting transcripts, national policy reports, TV episode transcripts, and scientific research papers. We include the QASPER Abstract Articles dataset for completeness, but would like to highlight that many models seem to oversaturate the benchmark and approach 1.0 NDCG@10. Results are presented in Table 6. nomic-embed-text-v1 beats jina-embeddings-v2-base-en across sequence lengths. nomic-embed-text-v1 beats M2-Bert at 2048 and is competitive at 8192. At sequence length 4096, nomic-embed-text-v1 is competitive with E5 Mistral while being significantly smaller.

5.4 Few-Shot Evaluation of BEIR

While the BEIR component of MTEB was originally purposed as a zero-shot benchmark, several top open-source models, including BGE (Xiao et al., 2023), GTE (Li et al., 2023), and E5-Mistral (Wang et al., 2023b) report training on train splits of BEIR benchmark datasets such as FEVER and HotpotQA. To understand the impact of this on our

[8] `https://huggingface.co/datasets/jinaai/big-patent-clustering`

Model	Seq	NarrativeQA	WikiCities	SciFact	BigPatent	Avg
`nomic-embed-text-v1`	128	20.1	90.0	65.4	18.5	48.5
`nomic-embed-text-v1-ablated`	128	20.8	86.8	65.2	17.5	47.6
`jina-embeddings-base-v2`	128	19.6	79.9	62.1	14.4	44.0
`text-embedding-ada-002`	128	25.4	84.9	68.8	16.6	48.9
`text-embedding-3-small`	128	29.5	87.5	68.8	15.0	50.2
`text-embedding-3-large`	128	45.6	87.9	74.8	16.5	56.2
`nomic-embed-text-v1`	512	23.9	88.7	70.5	25.3	52.1
`nomic-embed-text-v1-ablated`	512	25.7	81.9	71.5	23.7	50.7
`jina-embeddings-base-v2`	512	21.3	79.3	66.7	21.9	47.3
`text-embedding-ada-002`	512	25.5	84.8	72.6	23.0	51.5
`text-embedding-3-small`	512	32.2	89.0	73.2	23.6	54.5
`text-embedding-3-large`	512	48.1	89.9	77.6	23.6	59.6
`nomic-embed-text-v1`	8191	37.8	84.3	70.2	24.5	54.2
`nomic-embed-text-v1-ablated`	8191	44.0	77.4	69.1	23.6	53.5
`jina-embeddings-base-v2`	8191	39.4	75.7	69.4	23.1	51.9
`text-embedding-ada-002`	8191	41.1	84.7	72.7	22.5	55.3
`text-embedding-3-small`	8191	47.1	89.9	73.3	22.5	58.3
`text-embedding-3-large`	8191	51.6	86.2	77.7	19.3	58.7

Table 4: Jina Long Context Evaluation Benchmark. Numbers for `text-embedding-ada-002` and `jina-embeddings-base-v2` taken from (Günther et al., 2024).

downstream scores, we also train a nomic-embed-text-v1-ablated model that omits the FEVER, HotpotQA, and MEDI datasets. As reported in Table 1, this decreases our overall MTEB score by about one point. To maintain an apples-to-apples comparison with top open-source models, we opt to train on the FEVER, HotpotQA, and MEDI datasets for the released version of nomic-embed-text-v1. Unfortunately, due to the nature of closed-source models, we have no indication regarding whether closed-source models trained on these datasets.

6 Training Resources

Full training of nomic-embed-text-v1 can be conducted in a single week on one 8xH100 node. Masked language modeling of nomic-bert-2048 takes roughly 4 days. Contrastive pretraining lasts 3 and a half days. Contrastive fine-tuning takes one hour. We encourage the reader to initialize from our nomic-bert-2048 or Unsupervised Contrastive checkpoints, released under the same license as nomic-embed-text-v1.

7 Conclusion

We release the first fully open-source long context text embedding model that surpasses OpenAI Ada-002 performance on both sort and long context benchmarks. We release the model weights and training code under a permissible license as well as the recipe, including data, to reproduce the model.

7.1 Contributions

Zach Nussbaum lead the project, including the majority of the implementation, training and data decisions present in the final version, as well as making several design decisions at all levels of the stack. Jack Morris made several design contributions regarding dataset curation and model architecture. Brandon Duderstadt made several design contributions across the entire stack and wrote the base implementation of the data curation pipeline. Andriy Mulyar set early project direction, reviewed code implementations, and made several model design and dataset curation contributions.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

emozilla. 2023. Dynamically scaled rope further increases performance of long context llama with zero fine-tuning.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open Question Answering Over Curated and Extracted Knowledge Bases. In *KDD*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: long form question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA. Association for Computational Linguistics.

Wikimedia Foundation. Wikimedia downloads.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Jonas Geiping and Tom Goldstein. 2022. Cramming: Training a language model on a single gpu in one day.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task.

Michael Günther, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. 2023. Jina embeddings: A novel set of high-performance sentence embedding models.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2024. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. 2024. Sleeper agents: Training deceptive llms that persist through safety training.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. CodeSearchNet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Atlas: Few-shot learning with retrieval augmented language models.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

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

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021a. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021b. Paq: 65 million probably-asked questions and what you can do with them.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. 2020. S2orc: The semantic scholar open research corpus.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Jiawei Han Luyu Gao, Yunyi Zhang and Jamie Callan. 2021. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. 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, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021a. Large dual encoders are generalizable retrievers.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021b. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jacob Portes, Alex Trott, Sam Havens, Daniel King, Abhinav Venigalla, Moin Nadeem, Nikhil Sardana, Daya Khudia, and Jonathan Frankle. 2023. Mosaicbert: A bidirectional encoder optimized for fast pretraining.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models.

Nils Reimers, Elliot Choi, Amr Kayid, Alekhya Nandula, Manoj Govindassamy, and Abdullah Elkady. 2023. Introducing embed v3.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Jon Saad-Falcon, Dan Fu, and Simran Arora. 2024. Long-context retrieval models with monarch mixer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. *CoRR*, abs/1906.03741.

Noam Shazeer. 2020. Glu variants improve transformer.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-lm: Training multi-billion parameter language models using model parallelism.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023a. One embedder, any task: Instruction-finetuned text embeddings.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023b. Roformer: Enhanced transformer with rotary position embedding.

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

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Voyage. 2023. Excited to announce voyage embeddings!

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023a. Simlm: Pre-training with representation bottleneck for dense passage retrieval.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Improving text embeddings with large language models.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.

Appendix

Table 5: Pretraining Dataset Distribution

Dataset	Datapoints	% Dataset
Reddit[a]	64,978,944	0.28
PAQ (Lewis et al., 2021b)	52,953,088	0.23
Amazon Reviews (Ni et al., 2019)	38,682,624	0.16
S2ORC Title Abstract (Lo et al., 2020)	35438592	0.15
WikiAnswers (Fader et al., 2014)	9,912,320	0.04
S2ORC Citation Titles (Lo et al., 2020)	7,585,792	0.03
S2ORC Abstract Citation (Lo et al., 2020)	7,503,872	0.03
S2ORC Abstract Body (Lo et al., 2020)	6,389,760	0.03
Wikipedia Title Body (Foundation)	6,078,464	0.03
Gooaq (Khashabi et al., 2021)	1,245,184	0.01
Codesearch (Husain et al., 2019)	835,584	<.01
AGNews (Zhang et al., 2016)	409,600	<.01
CCNews (Hamborg et al., 2017)	344,064	<.01
NPR [b]	344,064	<.01
CNN (See et al., 2017)	278,528	<.01
Yahoo Title-Answer [c]	262,144	<.01
AmazonQA (Gupta et al., 2019)	212,992	<.01
Yahoo Title-Question [d]	196,608	<.01
Sentence Compression (Filippova and Altun, 2013)	163,840	<.01
YahooQA [e]	131,072	<.01
ELI5 (Fan et al., 2019)	98,304	<.01
Altlex (Hidey and McKeown, 2016)	98,304	<.01
Wikihow (Koupaee and Wang, 2018)	81,920	<.01
SimpleWiki (Coster and Kauchak, 2011)	81,920	<.01
StackExchange Duplicate Questions [f]	65,536	<.01
StackExchange Title Body [g]	65,536	<.01
StackExchange Body Body [h]	65,536	<.01
Quora Duplicate Questions [i]	32,768	<.01
SQuAD (Rajpurkar et al., 2016)	16,384	<.01
Total	234,553,344	1

[a] https://huggingface.co/datasets/sentence-transformers/reddit-title-body
[b] https://files.pushshift.io/news/
[c] https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset
[d] https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset
[e] https://www.kaggle.com/soumikrakshit/yahoo-answers-dataset
[f] https://data.stackexchange.com/apple/query/fork/1456963
[g] https://data.stackexchange.com/apple/query/fork/1456963
[h] https://data.stackexchange.com/apple/query/fork/1456963
[i] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Model	Seq	Param.	Tau Scr.	Tau Gov.	Tau QMS.	QASP. Tit. Art.	QASP. Abs. Art.	Avg
Unsupervised Models								
Jina$_{\texttt{base-v2}}$ (Günther et al., 2024)	2048	137M	87.2	97.7	35.1	95.3	99.7	83.0
Jina$_{\texttt{base-v2}}$ (Günther et al., 2023)	8192	137M	93.3	98.6	40.8	95.1	99.3	85.5
nomic-embed-text-v1-ablated	2048	137M	83.1	97.3	49.4	97.4	99.9	85.4
nomic-embed-text-v1-ablated	4096	137M	89.1	97.6	49.6	97.5	99.9	86.7
nomic-embed-text-v1-ablated	8192	137M	92.5	97.8	47.6	96.5	99.9	**86.9**
nomic-embed-text-v1	2048	137M	86.1	96.9	47.8	96.1	99.7	85.3
nomic-embed-text-v1	4096	137M	89.0	97.4	45.7	95.8	99.9	85.6
nomic-embed-text-v1	8192	137M	90.9	97.8	44.2	94.9	99.9	85.5
text-embedding-ada-002	8192	N/A	37.3	44.3	7.30	85.1	89.7	52.7
text-embedding-3-small	8192	N/A	92.2	97.7	27.4	95.9	98.9	82.4
text-embedding-3-large	8192	N/A	88.0	93.6	25.5	93.2	96.8	79.4
E5$_{\texttt{mistral}}$ (Wang et al., 2023b)	4096	7B	95.9	98.3	46.8	98.4	99.8	**87.8**
Supervised Models								
M2-Bert (Saad-Falcon et al., 2024)	2048	80M	81.8	94.7	58.5	87.3	95.5	83.6
M2-Bert (Saad-Falcon et al., 2024)	8192	80M	94.7	96.5	64.1	86.8	97.5	**87.9**

Table 6: Results on the LoCo benchmark (Saad-Falcon et al., 2024). NCDG@10 is reported for each dataset. We split evaluations into parameter class and whether the evaluation is performed in a supervised or unsupervised setting. We bold the top-performing model in each split. Nomic-embed-text-v1 is the best-performing 100M parameter class unsupervised model. Nomic-embed-text-v1 is competitive with the top-performing models in both the 7B parameter class and with models trained in a supervised setting specifically for the LoCo benchmark.