NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models

Chankyu Lee [*1] **Rajarshi Roy** [1] **Mengyao Xu** [1] **Jonathan Raiman** [1]

Mohammad Shoeybi [1] **Bryan Catanzaro** [1] **Wei Ping** [* 1]

[1] NVIDIA

Abstract

Decoder-only large language model (LLM)-based embedding models are beginning to outperform BERT or T5-based embedding models in general-purpose text embedding tasks, including dense vector-based retrieval. In this work, we introduce the `NV-Embed` model with a variety of architectural designs and training procedures to significantly enhance the performance of LLM as a versatile embedding model, while maintaining its *simplicity* and *reproducibility*. For model architecture, we propose a *latent attention layer* to obtain pooled embeddings, which consistently improves retrieval and downstream task accuracy compared to mean pooling or using the last `<EOS>` token embedding from LLMs. To enhance representation learning, we remove the causal attention mask of LLMs during contrastive training. For model training, we introduce a two-stage contrastive instruction-tuning method. It first applies contrastive training with instructions on retrieval datasets, utilizing in-batch negatives and curated hard negative examples. At stage-2, it blends various non-retrieval datasets into instruction tuning, which not only enhances non-retrieval task accuracy but also improves retrieval performance. Combining these techniques, our `NV-Embed` model, using only publicly available data, has achieved a record-high score of 69.32, ranking No. 1 on the Massive Text Embedding Benchmark (MTEB) (as of May 24, 2024), with 56 tasks, encompassing retrieval, reranking, classification, clustering, and semantic textual similarity tasks. Notably, our model also attains the highest score of 59.36 on 15 retrieval tasks in the MTEB benchmark (also known as BEIR). We will open-source the model at: `https://huggingface.co/nvidia/NV-Embed-v1`.

1 Introduction

Embedding or dense vector representation of text (Mikolov et al., 2013; Devlin et al., 2018) encodes its semantic information and can be used for many downstream applications, including retrieval, reranking, classification, clustering, and semantic textual similarity tasks. The embedding-based retriever is also a critical component for retrieval-augmented generation (RAG) (Lewis et al., 2020), which allows LLMs to access the most up-to-date external or proprietary knowledge without modifying the model parameters (Liu et al., 2024; Guu et al., 2020; Shi et al., 2023; Wang et al., 2023a).

The embedding models built on bidirectional language models (Devlin et al., 2018; Raffel et al., 2020) have dominated the landscape for years (e.g., Reimers & Gurevych, 2019; Gao et al., 2021; Wang et al., 2022; Izacard et al., 2021; Ni et al., 2021), although one notable exception is Neelakantan et al. (2022). The most recent work by Wang et al. (2023b) demonstrates that decoder-only LLMs can outperform frontier bidirectional embedding models (Wang et al., 2022; Ni et al., 2021; Chen et al., 2023) in

[*]Correspondence to: Chankyu Lee <chankyul@nvidia.com>, Wei Ping <wping@nvidia.com>.

retrieval and general-purpose embedding tasks. However, previous leading efforts (Wang et al., 2023b; Meng et al., 2024) have depended on fine-tuning LLMs using large volumes of proprietary synthetic data from GPT-4, which is not readily available to the community.

In this work, we introduce `NV-Embed`, a generalist embedding model that significantly enhances the performance of decoder-only LLMs for embedding and retrieval tasks. Specifically, we make the following contributions:

1. For model architecture, we propose a novel *latent attention layer* to obtain pooled embeddings for a sequence of tokens. In contrast to the popular average pooling in bidirectional embedding models (e.g., Wang et al., 2022) and the last `<EOS>` token embedding in decoder-only LLMs (Neelakantan et al., 2022; Wang et al., 2023b), our proposed pooling technique consistently improves the accuracy of retrieval and other downstream tasks. To further enhance the representation learning, we remove the causal attention mask during the contrastive training of decoder-only LLM, resulting in solid improvements. Our design is simpler yet more effective compared to recent related work (BehnamGhader et al., 2024; Muennighoff et al., 2024), which involves an additional training phase with masked token prediction or a mixed training objective.

2. For model training, we introduce a two-stage contrastive instruction-tuning method, starting with the pretrained Mistral-7B (Jiang et al., 2023). In the first stage, we apply contrastive training with instructions on retrieval datasets, utilizing in-batch negative and curated hard-negative examples. In the second stage, we blend carefully curated non-retrieval datasets into the stage-one training data. Since in-batch negative samples may be misleading for non-retrieval tasks, we disable in-batch negative training in stage two. This design not only improves the accuracy of classification, clustering, and semantic textual similarity tasks, but also surprisingly enhances retrieval performance. Note that our training data is entirely publicly available and does not include any synthetic data from proprietary models like GPT-4. Our model is also not fine-tuned from existing embedding models. [2]

3. Combining all the techniques, our `NV-Embed` model sets a new record high score of **69.32** and ranks No. 1 (as of May 22, 2024) on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022) across 56 embedding tasks. It significantly outperforms the previous leading embedding models: E5-mistral-7b-instruct (score: 66.63) (Wang et al., 2023b), SFR-Embedding (score: 67.56) (Meng et al., 2024), and Voyage-large-2-instruct (score: 68.28) (Voyage-AI, 2024). Notably, our model also attained the highest score of 59.35 on 15 retrieval tasks within the MTEB, which is drawn from BEIR benchmark (Thakur et al., 2021).

We organize the rest of the paper in the following. In § 2, we discuss the related work. We present the architectural and training details in § 3.

2 Related Work

2.1 Bidirectional Embedding Models

BERT (Devlin et al., 2018) or T5 (Raffel et al., 2020)-based embedding models have long been the dominant approaches for general-purpose embedding tasks. Early examples include Sentence-BERT (Reimers & Gurevych, 2019) and SimCSE (Gao et al., 2021), which finetune BERT on natural language inference (NLI) datasets. In general, these embedding models are first initialized from pre-trained BERT (Wang et al., 2022; Izacard et al., 2021) or T5 encoders (Ni et al., 2021). Then, they are further pre-trained with contrastive learning on curated unsupervised (Izacard et al., 2021) or weakly-supervised text pairs (Wang et al., 2022). Finally, the embedding models (Li et al., 2023; Wang et al., 2022; Ni et al., 2021; Chen et al., 2023) are fine-tuned on a variety of supervised data, including MS MARCO (Nguyen et al., 2016), for retrieval and other downstream tasks. Note that all the state-of-the-art embedding models are trained in this supervised manner. Some of the most recent frontier models in this category include mxbai-embed-large-v1 (Lee et al., 2024b) (MTEB: 64.68), UAE-Large-V1 (Li & Li, 2023) (MTEB: 64.64), and voyage-large-2-instruct (Voyage-AI, 2024) (MTEB: 68.28).

[2]For example, SFR-Embedding is fine-tuned from E5-mistral-7b-instruct.

2.2 Decoder-only LLM-based Embedding Models

Decoder-only LLMs (Brown et al., 2020) were believed to underperform bidirectional models on general-purpose embedding tasks because: *i*) unidirectional attention limits the representation learning capability, and *ii*) the scaling of LLMs leads to very high-dimension embeddings, which may suffer from the *curse of dimensionality*.

Neelakantan et al. (2022) initializes the embedding models with pre-trained GPT-3 models (Brown et al., 2020) and applies continued contrastive training. The hidden state from the last layer corresponding to the special token <*EOS*> at the end of the sequence is taken as the embedding of the input sequence. The latest text-embedding-3-large obtains MTEB score 64.59 (OpenAI, 2024) Most recently, E5-Mistral (Wang et al., 2023b) (METB: 66.63) applies contrastive learning with task-specific instructions on Mistral 7B (Jiang et al., 2023). It begins to outperform the state-of-the-art bidirectional models on comprehensive embedding benchmarks (Muennighoff et al., 2022) by utilizing a massive amount of synthetic data from the proprietary GPT-4 model. LLM2Vec (BehnamGhader et al., 2024) (METB score: 65.01) tries to build the embedding model from LLMs while only using public available data, but it is still worse than E5-Mistral.

Given the notable success of E5-Mistral, SFR-Embedding-Mistral (Meng et al., 2024) (METB: 67.56) further fine-tunes it on the blend of non-retrieval and retrieval datasets for improved accuracy on both tasks, which is closely related to our `NV-Embed`. However, there are the following key differences: 1) `NV-Embed` is trained from scratch on Mistral 7B LLM directly using public available data, and not dependent on other embedding model or proprietary synthetic data. Consequently, we introduce a new architecture that eliminates unnecessary causal attention mask and further improves the sequence pooling mechanism with latent attention layer. 2) SFR-Embedding-Mistral uses task-homogeneous batching, which constructs batches consisting exclusively of samples from a single task. In contrast, our `NV-Embed` uses well-blended batches consisting samples from all tasks to avoid potential "zigzag" gradient updates, which leads to a new record high score on both full MTEB and retrieval tasks compared to SFR-Embedding-Mistral.

There are other recent works. Gecko (Lee et al., 2024a) (METB: 66.31) attempts to distill a smaller bidirectional embedding model from a decoder-only LLM (Gemini et al., 2023) by generating synthetic paired data. It refines the data quality by retrieving a set of candidate passages for each query and relabeling the positive and hard negative passages using the LLM. In addition, GritLM (Muennighoff et al., 2024) (METB: 65.66) unifies text embedding and generation into a single model.

3 Method

In this section, we describe our architecture designs and two-stage instruction-tuning method.

3.1 Bidirectional Attention

The causal attention mask in decoder-only LLMs is introduced for next-token prediction task (Vaswani et al., 2017). In principle, causal mask in decoder blocks prevents information leakage by allowing the decoder to attend only to previous positions during auto-regressive text generation. However, it is observed that unidirectional attention limits the model's representation power, as evidenced by the poor performance of GPT models compared to similarly sized BERT or T5 models on natural language understanding benchmarks (e.g., Wang et al., 2019). In recent, LLM2Vec (BehnamGhader et al., 2024) introduces additional training phase with a specially designed masked token prediction to warm-up the bidirectional attention. GRIT (Muennighoff et al., 2024) utilizes a hybrid objective with both bidirectional representation learning and causal generative training. In contrast, we simply remove the causal attention mask of decoder-only LLM during the contrastive learning and find it works compellingly well as demonstrated by our results. As a result, we go with simple solution.

3.2 Latent Attention Layer

There are two popular methods to obtain the embedding for a sequence of tokens: *i)* mean pooling, and *ii*) the last `<EOS>` token embedding. Previous bidirectional embedding models typically use mean pooling (Wang et al., 2022; Izacard et al., 2021), while the last `<EOS>` token embedding is more popular for decoder-only LLM based embedding models. However, both methods have certain

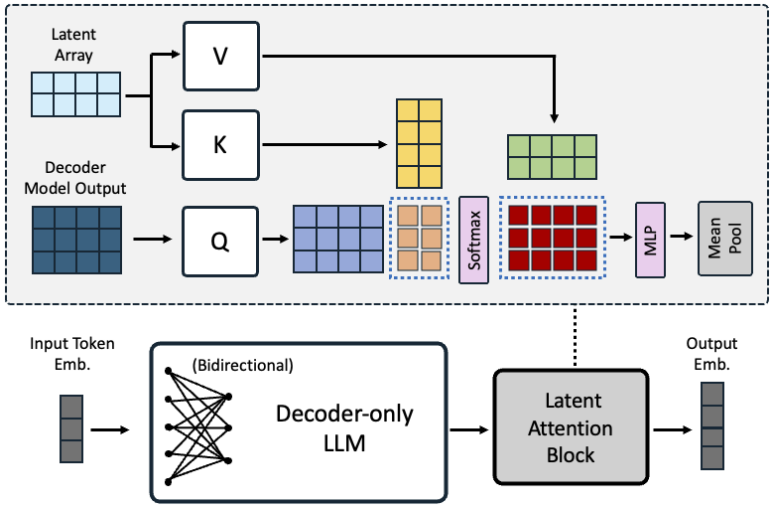

Figure 1: The illustration of proposed architecture design comprising of decoder-only LLM followed by latent attention layer. Latent attention layer functions as a form of cross-attention where the decoder-only LLM output serves as queries (Q) and trainable latent array passes through the key-value inputs, followed by MLP. Blue dotted lines indicate the two matrix multiplications involved in QKV-attentions.

limitations. Mean pooling simply takes the average of token embeddings and may dilute the important information from key phrases, meanwhile the last <EOS> token embedding may suffer from *recency bias*, relying heavily on the output embedding of last token.

In this work, we propose a latent attention layer inspired by Jaegle et al. (2021) to achieve more expressive pooling of the sequences for general-purpose embedding tasks. Specifically, we denote the last layer hidden from decoder as the query $Q \in \mathbb{R}^{l \times d}$, where l is the length of sequence, and d is the hidden dimension. They are sent to attend the latent array $K = V \in \mathbb{R}^{r \times d}$, which are *trainable* "dictionary" used to obtain better representation, where r is the number of latents in the dictionary. The output of this cross-attention is $O \in \mathbb{R}^{l \times d}$,

$$O = \text{softmax}(QK^T)V \tag{1}$$

which is followed by a regular MLP consists of two linear transformations with a GELU activation in between. Our model uses latent attention layer with r of 512 and the number of heads as 8 for multi-head attention. Finally, we apply mean pooling after MLP layers to obtain the embedding of whole sequences. See Figure 1 for an illustration. It is worth mentioning here that our approach follows the spirit of dictionary learning to obtain better representation (e.g., Wang et al., 2018), which is different from the Perceiver IO architecture. We compare the proposed *latent attention layer* with normal self-attention and find consistent improvements in our ablation study.

3.3 Two-stage Instruction-Tuning

Instruction-tuning has been widely applied for training LLM to follow instructions (Wei et al., 2021; Ouyang et al., 2022) and to perform retrieval-augmented generation (Wang et al., 2023a; Liu et al., 2024). It has also been recently applied for training retrievers and general-purpose embedding models that can adapt their output embeddings with different instructions and task types (Asai et al., 2022; Wang et al., 2023b).

To obtain a generalist embedding model that can appropriately perform on retrieval and non-retrieval tasks (e.g., classification, clustering), we need take the characteristics of different tasks into account. For example, the use of in-batch negatives has been demonstrated to be highly efficient for training dense-embedding-based retrievers (e.g., Karpukhin et al., 2020), because it allows to reuse the computation and effectively train on B^2 question/passage pairs for each mini-batch with only B questions and corresponding positive passages. However, applying in-batch negatives trick can

mislead the embedding model for classification or clustering task, as the "passages" in the mini-batch may come from the the class and are not negatives.

Given these considerations, we introduce a two-stage instruction tuning method which first conducts contrastive training with instructions on a variety of retrieval datasets (details are in section 4.1), utilizing in-batch negatives and curated hard-negative examples. In the second stage, we perform contrastive instruction-tuning on a combination of retrieval and non-retrieval datasets (details are in section 4.2) without applying the trick of in-batch negatives. It is worth mentioning here that retrieval task presents greater difficulty compared to the other tasks so that our training strategy focuses on fine-tuning the model for retrieval initially. In second stage, we blend the remaining embedding tasks into the instruction-tuning.

4 Training Data

While recent embedding models (Wang et al., 2023b; Meng et al., 2024; Lee et al., 2024a) have utilized both public supervised datasets and proprietary synthetic data from GPT-4 (OpenAI, 2023) or Gemini (Gemini et al., 2023), we exclusively employ public datasets to demonstrate our model's capability in embedding tasks. Our training procedure incorporates both retrieval and non-retrieval tasks, including classification, clustering, and semantic textual similarity datasets.

Given a relevant query-document pair, the instructed query follows the instruction template as follows:

$$q^+_{\text{inst}} = \texttt{Instruct}: \texttt{task\_definition}\ \texttt{Query}: q^+ \tag{2}$$

The instruction templates for each `task_definition` are provided in Table 6 for training and Table 7 for evaluation. Note that we mask out the instruction tokens in the output embeddings during both training and evaluation, although they still impact the output due to self-attention. We do not add instruction prefixes to documents.

4.1 Public Retrieval Datasets

We adopt the retrieval datasets as follows: MS MARCO (Bajaj et al., 2016), HotpotQA (Yang et al., 2018), Natural Question (Kwiatkowski et al., 2019), PAQ (Lewis et al., 2021), Stackexchange (Stack-Exchange-Community, 2023), Natural language inference (Group et al., 2022), SQuAD (Rajpurkar et al., 2016), ArguAna (Wachsmuth et al., 2018), BioASQ (Tsatsaronis et al., 2015), FiQA (Maia et al., 2018), FEVER (Thorne et al., 2018). Typically, these datasets do not contain its own hardnegatives, necessitating the mining of such examples. To address this, we further finetune another encoder-based embedding model (Wang et al., 2022) to select the hardnegatives on those datasets. Refer to Table 6 for the number of samples used for training.

4.2 Public Non-retrieval Datasets

Besides retrieval datasets, we also utilize non-retrieval datasets from three sub-tasks in MTEB benchmark: classification, clustering and semantic similarity (STS). We pre-process these datasets to use the same format as retrieval datasets for contrastive training: instructed query q^+_{inst} (containing query q^+), positive document d^+ and hard negative documents $d^-_0, ..., d^-_n$.

We utilize the English training splits of various classification datasets from MTEB Huggingface datasets (Muennighoff et al., 2022; Lhoest et al., 2021). The classification datasets that we use are: AmazonReviews-Classification (McAuley & Leskovec, 2013), AmazonCounterfactual-Classification (O'Neill et al., 2021), Banking77-Classification (Casanueva et al., 2020), Emotion-Classification (Saravia et al., 2018), IMDB-Classification (Maas et al., 2011), MTOPIntent-Classification (Li et al., 2021), ToxicConversations-Classification (Adams et al., 2019), TweetSentimentExtraction-Classification (Maggie, 2020).

Because the training splits of Emotion-Classification and AmazonCounterfactual-Classification contain some content similar to their evaluation splits, we use BM25 (Robertson et al., 2009) similarity thresholds to remove similar content from the training splits before subsampling. We use the $text$ field as the q^+, $label\_text$ field as the d^+ and random sample among other $label\_text$ values for d^-_k. Since the AmazonReviewsClassification dataset does not provide the $label\_text$ field, we generate label texts associated with values in the $label$ field. For subsampling the classification datasets, we perform the stratified sampling across d^+.

Table 1: Top MTEB leaderboard models as of 2024-05-22. We use the original model names on the leaderboard for clarity.

Embedding Task Mertric	Retrieval (15) nDCG@10	Rerank (4) MAP	Cluter. (11) V-Meas.	PairClass. (3) AP	Class. (12) Acc.	STS (10) Spear.	Summ.(1) Spear.	Avg. (56)
NV-Embed	**59.36**	60.59	52.80	86.91	87.35	82.84	31.2	**69.32**
NV-Embed (mean pool)	58.71	60.75	52.80	85.85	87.06	82.53	30.49	68.98
Voyage-large-2-instruct	58.28	60.09	53.35	89.24	81.49	84.58	30.84	68.28
SFR-Embedding	59.00	60.64	51.67	88.54	78.33	85.05	31.16	67.56
Gte-Qwen1.5-7B-instruct	56.24	60.13	55.83	87.38	79.6	82.42	31.46	67.34
Voyage-lite-02-instruct	56.6	58.24	52.42	86.87	79.25	85.79	31.01	67.13
GritLM-7B	57.41	60.49	50.61	87.16	79.46	83.35	30.37	66.76
E5-mistral-7b-instruct	56.9	60.21	50.26	88.34	78.47	84.66	31.4	66.63
Google-gecko	55.7	58.9	47.48	87.61	81.17	85.07	32.63	66.31
LLM2Vec-Meta-Llama-3	56.63	59.69	46.45	87.79	75.92	83.58	30.94	65.01
Text-embed-3-large (OpenAI)	55.44	59.16	49.01	85.72	75.45	81.73	29.92	64.59

We approach clustering in a similar manner as classification by employing the cluster labels for positives and negatives. We utilize the raw cluster label datasets raw_arxiv, raw_biorxiv and raw_medrxiv datasets from MTEB Huggingface datasets and filter out common content from the MTEB evaluation set of {Arxiv/Biorxiv/Medrxiv}-Clustering-{S2S/P2P} tasks. We use the $title$ field for q^+ for the S2S datasets and the $abstract$ field for q^+ for the P2P datasets. We use the $category$ field or random sample from the $categories$ field for d^+ and random sample other categories for d_k^-. We also use the raw label dataset for TwentyNewsgroups-Clustering (Lang, 1995) and remove any content that match with the MTEB evaluation set of the TwentyNewsgroups-Clustering task. For subsampling the clustering datasets, we perform stratified sampling across d^+.

We use the training splits of three semantic similarity datasets STS12 (Agirre et al., 2012), STS22 (Chen et al., 2022), STS-Benchmark (Cer et al., 2017) from MTEB Huggingface datasets. For any pair of texts with associated relevance scores $(t_a, t_b, score)$, we create two examples $(q^+ = t_a, d^+ = t_b)$ and $(q^+ = t_b, d^+ = t_a)$ if $score \geq 4$. We mine the hard negatives d_k^- from the pool of all texts using BM25, selecting the highest matching texts with rank >=2 that do not have relevance scores >2.5 with q^+.

5 Experiments

5.1 Experimental Details

In this section, we describe our detailed experimental setups. We use a parameter-efficient finetuning (PEFT) method denoted as low-rank adaptation (LoRA) (Hu et al., 2021) to efficiently finetune our proposed NV-Embed model. We chose Mistral 7B (Jiang et al., 2023) as the base decoder-only LLM. We replace the attention mask from causal to bidirectional, and integrate the latent attention layer with 512 latents, 4096 hidden dimension size, and 8 multi-head attentions.

We train Mistral 7B LLM model end-to-end with a contrastive loss using LoRA with rank 16, alpha 32 and dropout rate of 0.1. We use Adam optimizer with 500 warm-up steps and learning rate 2e-5 for first stage and 1.5e-5 for second stage with linear decay. The model is finetuned with 128 batch size, where each batch is composed of a query paired with 1 positive and 7 hard negative documents. We train using Bfloat16, and set the maximum sequence length as 512 tokens. The special <BOS> and <EOS> tokens are appended at the start and end of given query and documents. The whole training is conducted in two stages where the model is initially trained on retrieval datasets utilizing in-batch negative technique. Subsequently, the model is trained with blended datasets with both retrieval and non-retrieval embedding tasks.

For evaluation, we assess our model using a maximum length of 512 tokens to ensure fair comparisons with prior work (Wang et al., 2023b), which also provides evaluation results based on 512 token limits. Evaluation instructions templates are available in Table 7.

5.2 MTEB Results

We evaluate the proposed NV-Embed model on the full MTEB benchmark (Muennighoff et al., 2022) encompassing 15 retrieval datasets, 4 reranking datasets, 12 classification datasets, 11 clustering

Table 2: Averaged MTEB scores on seven tasks after first stage training

Pool Type	EOS		Mean		Latent-attention		Self-attention	
Mask Type	bidirect	causal	bidirect	causal	bidirect	causal	bidirect	causal
Retrieval(15)	57.70	56.42	58.42	57.55	**59.00**	57.65	57.89	57.21
Rerank (4)	59.76	57.21	60.02	59.35	59.59	59.72	59.73	59.51
Clustering (11)	44.75	40.83	45.97	45.42	45.44	45.61	45.19	45.07
PairClass. (3)	86.17	83.63	87.45	84.46	87.59	82.02	86.51	85.74
Classification (12)	73.17	69.22	74.62	72.48	73.93	72.74	73.54	73.32
STS (10)	74.96	73.45	77.47	73.60	79.07	78.65	76.89	77.55
Summar. (1)	29.28	28.4	29.72	30.89	30.16	30.94	30.22	31.59
Average (56)	62.68	60.06	64.00	62.32	**64.18**	63.39	63.27	63.11

Table 3: Averaged MTEB scores on seven tasks after second stage training

Pool Type	EOS		Mean		Latent-attention		Self-attention	
Mask Type	bidirect	causal	bidirect	causal	bidirect	causal	bidirect	causal
Retrieval (15)	58.39	56.59	58.71	57.88	**59.36**	58.33	58.64	57.71
Rerank (4)	60.37	59.23	60.77	60.27	60.54	60.57	60.5	60.38
Clustering (11)	51.43	49.81	52.80	51.58	52.80	51.7	53.34	51.51
PairClass. (3)	84.06	80.99	87.45	82.89	86.91	83.45	86.12	84.44
Classification (12)	85.85	85.04	87.06	86.08	87.35	86.58	86.76	86.25
STS (10)	79.55	79.12	82.53	81.74	82.84	81.94	82.38	81.52
Summar. (1)	30.36	29.12	30.49	31.82	31.20	31.87	30.105	31.4
Average (56)	67.85	66.50	68.97	68.13	**69.32**	68.47	69.10	68.16

datasets, 3 pair classification datasets, 10 semantic textual similarity datasets, and 1 summarization dataset.

Table 1 shows the averaged MTEB scores for overall performance and seven sub-category tasks compared to all frontier models on the MTEB leaderboard[3]. Our `NV-Embed` model achieves a new record high score of **69.32** on the MTEB benchmark with 56 tasks and also attains the highest score of **59.36** on 15 retrieval tasks originally from the BEIR benchmark (Thakur et al., 2021).

Based on quantitative leaderboard results, we compare our `NV-Embed` with the recent frontier embedding models. The e5-mistral-7b-instruct (Wang et al., 2023b) and google-gecko (Lee et al., 2024a) utilize proprietary synthetic data to train their model in a single stage manner. In contrast, we recognize that retrieval task presents greater difficulty compared to the other embedding tasks and prioritizes our training strategy on fine-tuning the model for retrieval first, followed by blending the remaining sub-tasks into instruction-tuning, leading to substantially improved BEIR and overall METB results.

SFR-Embedding (Meng et al., 2024) demonstrates competitive scores on the MTEB (67.56) and BEIR (59.0) benchmarks by continuing to finetune the e5-mistral-7b-instruct model (Wang et al., 2023b). However, it remains largely constrained by the architectural limitations of its parent model, such as the causal attention mask and the last token pooling method. In contrast, our `NV-Embed` model is trained starting from the Mistral 7B LLM rather than finetuning e5-mistral-7b-instruct. It features a new architecture that removes the unnecessary causal attention mask and further improves the sequence pooling mechanism with a latent attention layer. Table 4 provides a detailed summary of task-wise BEIR and MTEB benchmarks.

5.3 Ablation Study

We perform ablation studies to compare causal and bidirectional attention for contrastive trainng. We also compare the the proposed *latent attention layer* with other pooling methods.

5.3.1 Causal Attention vs. Bidirectional Attention

To examine the impact of self-attention masks in decoder-only LLM models for embedding applications, we conducted experiments comparing bidirectional and causal mask types. As illustrated in Tables 2 and 3, the bidirectional mask consistently outperforms the causal mask based on the average

[3] `https://huggingface.co/spaces/mteb/leaderboard`

Table 4: Full BEIR and MTEB benchmark

Model Name	E5-mistral-7b	SFR-Embedding	Voyage-large2-instruct	EOS		Mean		Latent-attention		Self-attention	
				bidirect	causal	bidirect	causal	bidirect	causal	bidirect	causal
ArguAna	61.88	67.27	64.06	67.06	62.51	63.83	64.14	68.21	64.57	65.56	62.82
ClimateFEVER	38.40	36.41	32.65	33.92	31.05	34.09	31.52	34.72	33.38	34.78	33.05
CQADupStack	42.97	46.54	46.60	48.39	46.14	48.69	47.13	50.51	47.44	50.19	46.79
Dbpedia	48.90	49.06	46.03	48.03	46.28	49.12	48.17	48.29	47.46	47.74	47.96
FEVER	87.80	89.35	91.47	87.11	85.80	87.90	87.46	87.77	87.66	86.89	87.45
FiQA2018	56.62	60.55	59.76	59.72	56.86	60.84	57.73	63.10	60.01	62.81	58.83
HotpotQA	75.70	77.02	70.86	78.40	75.75	78.65	77.90	79.92	78.37	79.33	77.73
MSMARCO	43.10	43.41	40.60	46.10	45.60	46.23	45.80	46.49	46.10	46.80	46.00
NFCorpus	38.59	42.02	40.32	38.48	36.89	39.13	38.31	38.04	38.48	37.68	37.58
Natural Question	63.50	69.92	65.92	70.07	68.75	71.23	69.43	71.22	70.77	71.33	70.22
QuoraRetrieval	89.62	89.81	87.40	88.88	88.57	88.75	88.73	89.21	88.71	88.85	88.76
SCIDOCS	16.27	19.91	24.32	19.81	16.81	21.08	20.38	20.19	18.86	20.86	18.20
SciFact	76.41	78.06	79.99	77.21	75.48	77.53	78.22	78.43	79.17	76.83	77.38
TREC-COVID	87.33	87.10	85.07	85.34	84.42	85.87	84.44	85.88	85.81	83.75	84.64
Touche2020	26.39	29.00	39.16	27.36	28.00	27.66	28.84	28.38	28.08	26.21	28.26
BIOSSES	85.58	86.07	89.12	86.44	83.04	86.19	83.27	85.59	83.37	85.26	82.71
SICK-R	82.64	82.92	83.16	78.59	77.65	82.87	81.06	82.80	81.44	83.21	80.87
STS12	79.65	79.47	76.15	73.30	72.77	74.82	73.54	76.22	75.03	77.10	74.67
STS13	88.43	89.15	88.49	81.84	83.10	85.81	86.48	86.30	85.44	84.54	84.07
STS14	84.54	84.93	86.49	77.84	77.18	81.45	80.73	82.09	80.51	80.29	80.05
STS15	90.42	90.74	91.13	85.50	83.97	87.20	86.69	87.24	86.35	87.16	87.03
STS16	87.68	87.82	85.68	81.99	81.90	84.62	84.42	84.77	84.70	83.76	84.02
STS17	91.75	92.02	90.06	77.89	81.85	88.53	86.67	87.42	87.64	86.69	86.16
STS22	67.28	68.36	66.32	71.14	68.81	68.69	70.06	69.85	70.02	69.91	70.59
STSBenchmark	88.60	89.00	89.22	80.95	80.89	85.15	84.50	86.14	84.94	85.86	85.07
SummEval	31.40	31.16	30.84	30.36	29.12	30.49	31.82	31.20	31.87	30.11	31.40
SprintDuplicateQuestions	95.66	96.31	94.50	94.89	91.46	95.39	94.71	95.94	95.15	95.98	95.12
TwitterSemEval2015	81.62	81.52	86.32	70.94	65.73	75.81	67.80	78.73	69.06	76.31	72.41
TwitterURLCorpus	87.75	87.78	86.90	86.34	85.79	86.36	86.15	86.05	86.12	86.07	85.78
AmazonCounterfactual	78.69	77.93	77.60	94.69	93.87	94.48	94.10	95.12	93.88	94.78	93.64
AmazonPolarity	95.91	95.97	96.58	97.05	96.34	96.92	96.66	97.14	97.08	97.27	97.02
AmazonReviews	55.79	54.35	50.77	53.37	56.09	55.68	55.99	55.47	56.59	55.47	54.81
Banking77	88.23	88.81	86.96	87.93	87.08	89.13	88.55	90.34	89.08	89.76	89.37
Emotion	49.77	50.24	59.81	91.19	91.39	91.01	91.52	91.71	91.54	91.97	91.13
Imdb	94.78	94.79	96.13	97.15	96.34	96.83	96.26	97.06	96.69	97.13	97.03
MassiveIntent	80.57	79.99	81.08	78.89	76.37	80.13	78.91	80.07	80.42	79.88	79.31
MassiveScenario	82.39	82.20	87.95	81.53	78.79	81.80	81.94	81.74	83.24	81.92	81.96
MTOPDomain	96.12	96.36	98.86	96.28	95.34	96.58	96.08	96.51	95.93	96.98	96.33
MTOPIntent	86.11	86.30	86.97	88.16	85.58	88.58	88.19	89.77	88.60	88.97	89.27
ToxicConversations	69.59	69.33	83.58	93.17	93.16	92.86	92.57	92.60	93.41	93.21	93.04
TweetSentimentExtraction	63.72	63.64	71.55	79.62	79.19	80.68	79.86	80.64	80.41	81.10	79.72
Arxiv-P2P	50.45	52.08	51.81	53.60	53.23	53.45	53.23	53.76	53.21	53.51	53.24
Arxiv-S2S	45.50	47.38	44.73	48.23	48.71	49.52	48.79	49.59	49.01	49.61	49.00
Biorxiv-P2P	43.53	43.94	46.07	47.02	45.53	46.97	47.09	48.15	47.56	48.71	47.87
Biorxiv-S2S	40.24	41.14	40.64	43.99	43.52	44.03	43.26	44.74	43.76	45.36	44.58
Medrxiv-P2P	38.19	40.03	42.94	38.64	38.42	38.30	37.90	39.24	38.34	38.88	38.34
Medrxiv-S2S	37.45	39.00	41.44	36.67	38.52	36.94	36.61	36.98	36.88	37.53	36.98
Reddit	57.71	59.90	68.50	62.42	60.88	64.62	62.99	63.20	62.39	64.77	61.33
Reddit-P2P	66.49	67.64	64.86	67.67	63.92	68.01	66.59	68.01	66.85	68.17	65.99
StackExchange	73.10	74.25	74.16	72.44	68.49	77.07	73.35	74.99	72.36	76.58	72.03
StackExchange-P2P	45.91	46.78	45.10	41.32	36.40	40.95	37.79	42.04	38.99	41.87	38.28
TwentyNewsgroups	54.31	56.27	66.62	53.68	50.34	60.97	59.81	60.13	59.33	61.72	58.98
AskUbuntuDupQuestions	66.98	67.58	64.92	66.78	65.53	67.85	68.10	67.50	68.17	67.01	68.24
MindSmallRerank	32.60	32.72	30.97	30.53	28.71	31.10	30.98	30.82	31.50	31.81	31.63
SciDocsRR	86.33	86.58	89.34	88.05	86.33	87.71	87.24	87.26	87.24	87.38	86.74
StackOverflowDupQuestions	54.91	55.68	55.11	55.66	54.06	56.43	54.75	56.58	55.38	56.62	55.54
BEIR Average (15)	56.90	59.03	58.28	58.39	56.59	58.71	57.88	**59.36**	58.33	58.64	57.71
MTEB Average (56)	66.63	67.56	68.28	67.85	66.50	68.97	68.13	**69.32**	68.47	69.10	68.16

MTEB scores across 56 tasks for all pooling types. This indicates that embeddings generated with causal attention masks are significantly less effective than those produced with bidirectional attention masks.

5.3.2 Pooling Methods

To examine the impact of different pooling methods on embedding models, we conducted experiments comparing <EOS>-last, mean, latent-attention, and self-attention pooling types. As depicted in Tables 2 and 3, mean pooling consistently outperforms <EOS>-last token embedding based on the average MTEB scores across 56 tasks. This difference may be due to the last <EOS> token embedding being influenced by *recency bias*, showing an excessive dependence on the output of the final token.

To enhance performance beyond mean pooling, we experimented with adding the proposed latent-attention or self-attention layer (both followed by MLP) before mean pooling to address the issue of important information from key phrases being diluted. According to Table 2 and 3, self-attention

does not provide additional accuracy improvements for the embedding capabilities of decoder-only LLMs (i.e., mean pooling 68.97 vs. self-attention 69.10 on 56 MTEB tasks). It even slightly reduces accuracy on 15 retrieval tasks (i.e., mean pooling 58.71 vs. self-attention 58.64). This is not surprising, as the LLM already has many self-attention layers to learn the representation, and adding an additional one does not bring significant additive value.

In contrast, the latent-attention layer proved beneficial for retrieval, classification, and STS subtasks, as shown in Table 3. Specifically, the nDCG@10 accuracy of the more challenging 15 retrieval tasks improved (i.e., mean pooling 58.71 vs. latent-attention 59.36). We hypothesize that this is due to the "dictionary learning" provided by the latent array, which offers more expressive representation. The latent-attention layer effectively learns output embedding representations from decoder-only LLMs, mitigating the information dilution caused by averaging the output embeddings.

6 Conclusion

In this work, we introduce `NV-Embed` model which presents novel architectural design and two-staged training procedure to substantially enhance the LLM capability as a generalist embedding model. For model architecture, we propose a latent attention layer to obtain expressive pooled embeddings and remove the unnecessary causal attention mask of decoder-only LLMs. For model training, we introduce a two-stage contrastive instruction-tuning scheme to sequentially improve the embedding tasks encompassing retrieval, classification, clustering, and semantic textual similarity. As of May 24, 2024, our `NV-Embed` model obtains a new record high score on the Massive Text Embedding Benchmark (MTEB) with 56 tasks and also attains the highest score on BEIR benchmark (15 retrieval tasks in the MTEB benchmark). Notably, we obtain state-of-the-art results using only publicly available data, without any synthetic data from frontier proprietary LLMs, such as GPT-4.

References

Adams, C., Borkan, D., Sorensen, J., Dixon, L., Vasserman, L., and Thain, N. Jigsaw unintended bias in toxicity classification, 2019. URL https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification.

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. SemEval-2012 task 6: A pilot on semantic textual similarity. In Agirre, E., Bos, J., Diab, M., Manandhar, S., Marton, Y., and Yuret, D. (eds.), **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1051.

Asai, A., Schick, T., Lewis, P., Chen, X., Izacard, G., Riedel, S., Hajishirzi, H., and Yih, W.-t. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260*, 2022.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., and Reddy, S. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Casanueva, I., Temcinas, T., Gerz, D., Henderson, M., and Vulic, I. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*, mar 2020. URL https://arxiv.org/abs/2003.04807. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S. M., Cer, D., and Jurgens, D. (eds.), *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2023.

Chen, X., Zeynali, A., Camargo, C., Flöck, F., Gaffney, D., Grabowicz, P., Hale, S., Jurgens, D., and Samory, M. SemEval-2022 task 8: Multilingual news article similarity. In Emerson, G., Schluter, N., Stanovsky, G., Kumar, R., Palmer, A., Schneider, N., Singh, S., and Ratan, S. (eds.), *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pp. 1094–1106, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.semeval-1.155. URL https://aclanthology.org/2022.semeval-1.155.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

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

Group, S. N. et al. The stanford natural language inference (snli) corpus, 2022.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Lang, K. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pp. 331–339. Elsevier, 1995.

Lee, J., Dai, Z., Ren, X., Chen, B., Cer, D., Cole, J. R., Hui, K., Boratko, M., Kapadia, R., Ding, W., et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024a.

Lee, S., Shakir, A., Koenig, D., and Lipp, J. Open source strikes bread - new fluffy embeddings model, 2024b. URL https://www.mixedbread.ai/blog/mxbai-embed-large-v1.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Lewis, P., Wu, Y., Liu, L., Minervini, P., Küttler, H., Piktus, A., Stenetorp, P., and Riedel, S. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115, 2021.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

Li, H., Arora, A., Chen, S., Gupta, A., Gupta, S., and Mehdad, Y. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In Merlo, P., Tiedemann, J., and Tsarfaty, R. (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2950–2962, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.257. URL https://aclanthology.org/2021.eacl-main.257.

Li, X. and Li, J. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023. URL https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1.

Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., and Zhang, M. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Liu, Z., Ping, W., Roy, R., Xu, P., Shoeybi, M., and Catanzaro, B. ChatQA: Surpassing GPT-4 on conversational QA and RAG. *arXiv preprint arXiv:2401.10225*, 2024.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Maggie, Phil Culliton, W. C. Tweet sentiment extraction, 2020. URL https://kaggle.com/competitions/tweet-sentiment-extraction.

Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., and Balahur, A. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pp. 1941–1942, 2018.

McAuley, J. and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pp. 165–172, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324090. doi: 10.1145/2507157.2507163. URL https://doi.org/10.1145/2507157.2507163.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

Muennighoff, N., Su, H., Wang, L., Yang, N., Wei, F., Yu, T., Singh, A., and Kiela, D. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*, 2024.

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

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. MS MARCO: A human-generated machine reading comprehension dataset. 2016.

Ni, J., Qu, C., Lu, J., Dai, Z., Ábrego, G. H., Ma, J., Zhao, V. Y., Luan, Y., Hall, K. B., Chang, M.-W., et al. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*, 2021.

O'Neill, J., Rozenshtein, P., Kiryo, R., Kubota, M., and Bollegala, D. I wish i would have loved this one, but i didn't–a multilingual dataset for counterfactual detection in product reviews. *arXiv preprint arXiv:2104.06893*, 2021.

OpenAI. GPT-4, 2023.

OpenAI. New embedding models and api updates, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Robertson, S., Zaragoza, H., et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. CARER: Contextualized affect representations for emotion recognition. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL https://aclanthology.org/D18-1404.

Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., and Yih, W.-t. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

Stack-Exchange-Community. Stack exchange data dump, 2023.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M. R., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Voyage-AI. voyage-large-2-instruct: Instruction-tuned and rank 1 on mteb, 2024.

Wachsmuth, H., Syed, S., and Stein, B. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 241–251, 2018.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Wang, B., Ping, W., McAfee, L., Xu, P., Li, B., Shoeybi, M., and Catanzaro, B. Instructretro: Instruction tuning post retrieval-augmented pretraining. *arXiv preprint arXiv:2310.07713*, 2023a.

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

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

Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., and Saurous, R. A. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International conference on machine learning*, pp. 5180–5189. PMLR, 2018.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

A Implementation Details

Table 5: Parameters used in the experiments

Parameter	Value
Batchsize	128
Number of Hardnegatives	7
Warm-up Steps	500
Training Steps	First stage - 20k Second stage - 18k
Learning Rate	First stage - 2e-5 Second stage - 1.5e-5
LoRA Params	Rank - 16 Alpha - 32 Dropout - 0.1
Weight Decay	0.03
Optimizer	Adam
Padding Side	right
Number of Latents (r)	512
Latent Width (d)	4096
Multi-Attention Heads	8

Table 6: Instructions and number of samples used for each training dataset. For an apples-to-apples comparison, we use the same instructions as in Wang et al. (2023b) for shared training datasets.

Task Name	Instruction Template	Number of Samples
ArguAna	Given a claim, find documents that refute the claim	16k
Natural Language Inference	Retrieve semantically similar text Given a premise, retrieve a hypothesis that is entailed by the premise	20k
PAQ, MSMARCO	Given a web search query, retrieve relevant passages that answer the query Given a question, retrieve passages that answer the question Given a question, retrieve documents that can help answer the question	100k, 200k
SQUAD	Given a question, retrieve Wikipedia passages that answer the question	100k
StackExchange	Given a question paragraph at StackExchange, retrieve a question duplicated paragraph	80k
Natural Question	Given a question, retrieve Wikipedia passages that answer the question	100k
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question	50k
FEVER	Given a claim, retrieve documents that support or refute the claim	50k
FiQA2018	Given a financial question, retrieve user replies that best answer the question	5k
BioASQ	Given a question, retrieve detailed question descriptions that are duplicates to the given question	2k
STS12, STS22, STSBenchmark	Retrieve semantically similar text.	40k
AmazonCounterfactualClassification	Classify a given Amazon customer review text as either counterfactual or not-counterfactual	10k
AmazonReviewsClassification	Classify the given Amazon review into its appropriate rating category	20k
Banking77Classification	Given a online banking query, find the corresponding intents	10k
EmotionClassification	Classify the emotion expressed in the given Twitter message into one of the six emotions:anger, fear, joy, love, sadness, and surprise	16k
ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset	15k
MTOPIntentClassification	Classify the intent of the given utterance in task-oriented conversation	10k
ToxicConversationsClassification	Classify the given comments as either toxic or not toxic	40k
TweetSentimentExtractionClassification	Classify the sentiment of a given tweet as either positive, negative, or neutral	40k
ArxivClusteringP2P	Identify the main and secondary category of Arxiv papers based on the titles and abstracts	25k
ArxivClusteringS2S	Identify the main and secondary category of Arxiv papers based on the titles	25k
BiorxivClusteringP2P	Identify the main category of Biorxiv papers based on the titles and abstracts	25k
BiorxivClusteringS2S	Identify the main category of Biorxiv papers based on the titles	15k
MedrxivClusteringP2P	Identify the main category of Medrxiv papers based on the titles and abstracts	15k
MedrxivClusteringS2S	Identify the main category of Medrxiv papers based on the titles	15k
TwentyNewsgroupsClustering	Identify the topic or theme of the given news articles	10k

Table 7: Instructions used for evaluation on the MTEB benchmark. "STS*" indicates we use the same instructions for all the STS tasks. For an apples-to-apples comparison, we use the same instructions as in Wang et al. (2023b).

Task Name	Instruction Template
ArguAna	Given a claim, find documents that refute the claim
ClimateFEVER	Given a claim about climate change, retrieve documents that support or refute the claim
DBPedia	Given a query, retrieve relevant entity descriptions from DBPedia
FEVER	Given a claim, retrieve documents that support or refute the claim
FiQA2018	Given a financial question, retrieve user replies that best answer the question
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
MSMARCO	Given a web search query, retrieve relevant passages that answer the query
NFCorpus	Given a question, retrieve relevant documents that best answer the question
Natural Question	Given a question, retrieve Wikipedia passages that answer the question
QuoraRetrieval	Given a question, retrieve questions that are semantically equivalent to the given question
SCIDOCS	Given a scientific paper title, retrieve paper abstracts that are cited by the given paper
SciFact	Given a scientific claim, retrieve documents that support or refute the claim
Touche2020	Given a question, retrieve detailed and persuasive arguments that answer the question
TREC-COVID	Given a query, retrieve documents that answer the query
STS	Retrieve semantically similar text.
SummEval	Given a news summary, retrieve other semantically similar summaries
AmazonCounterfactualClassification	Classify a given Amazon customer review text as either counterfactual or not-counterfactual
AmazonPolarityClassification	Classify Amazon reviews into positive or negative sentiment
AmazonReviewsClassification	Classify the given Amazon review into its appropriate rating category
Banking77Classification	Given a online banking query, find the corresponding intents
EmotionClassification	Classify the emotion expressed in the given Twitter message into one of the six emotions:anger, fear, joy, love, sadness, and surprise
ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset
MassiveIntentClassification	Given a user utterance as query, find the user intents
MassiveScenarioClassification	Given a user utterance as query, find the user scenarios
MTOPDomainClassification	Classify the intent domain of the given utterance in task-oriented conversation
MTOPIntentClassification	Classify the intent of the given utterance in task-oriented conversation
ToxicConversationsClassification	Classify the given comments as either toxic or not toxic
TweetSentimentExtractionClassification	Classify the sentiment of a given tweet as either positive, negative, or neutral
ArxivClusteringP2P	Identify the main and secondary category of Arxiv papers based on the titles and abstracts
ArxivClusteringS2S	Identify the main and secondary category of Arxiv papers based on the titles
BiorxivClusteringP2P	Identify the main category of Biorxiv papers based on the titles and abstracts
BiorxivClusteringS2S	Identify the main category of Biorxiv papers based on the titles
MedrxivClusteringP2P	Identify the main category of Medrxiv papers based on the titles and abstracts
MedrxivClusteringS2S	Identify the main category of Medrxiv papers based on the titles
RedditClustering	Identify the topic or theme of Reddit posts based on the titles
RedditClusteringP2P	Identify the topic or theme of Reddit posts based on the titles and posts
StackExchangeClustering	Identify the topic or theme of StackExchange posts based on the titles
StackExchangeClusteringP2P	Identify the topic or theme of StackExchange posts based on the given paragraphs
TwentyNewsgroupsClustering	Identify the topic or theme of the given news articles
AskUbuntuDupQuestions	Retrieve duplicate questions from AskUbuntu forum
MindSmallReranking	Retrieve relevant news articles based on user browsing history
SciDocsRR	Given a title of a scientific paper, retrieve the titles of other relevant papers
StackOverflowDupQuestions	Retrieve duplicate questions from StackOverflow forum
SprintDuplicateQuestions	Retrieve duplicate questions from Sprint forum
TwitterSemEval2015	Retrieve tweets that are semantically similar to the given tweet
TwitterURLCorpus	Retrieve tweets that are semantically similar to the given tweet