

# GigaEmbeddings — Efficient Russian Language Embedding Model

Kolodin Egor<sup>1,2</sup>, Khomich Daria<sup>2,3</sup>, Savushkin Nikita<sup>2,3</sup>,  
Ianina Anastasia<sup>1,4</sup>, Fyodor Minkin<sup>1,2</sup>

<sup>1</sup>MIPT, <sup>2</sup>SaluteDevices, <sup>3</sup>MSU, <sup>4</sup>Wildberries

Correspondence: [kolodin.ei@phystech.edu](mailto:kolodin.ei@phystech.edu)

## Abstract

We introduce GigaEmbeddings, a novel framework for training high-performance Russian-focused text embeddings through hierarchical instruction tuning of the decoder-only LLM designed specifically for Russian language (GigaChat-3B). Our three-stage pipeline, comprising large-scale contrastive pre-training in web-scale corpora, fine-tuning with hard negatives, and multitask generalization across retrieval, classification, and clustering tasks, addresses key limitations of existing methods by unifying diverse objectives and leveraging synthetic data generation. Architectural innovations include bidirectional attention for contextual modeling, latent attention pooling for robust sequence aggregation, and strategic pruning of 25% of transformer layers to enhance efficiency without compromising performance. Evaluated on the ruMTEB benchmark spanning 23 multilingual tasks, GigaEmbeddings achieves state-of-the-art results (69.1 avg. score), outperforming strong baselines with a larger number of parameters.

## 1 Introduction

Text embeddings, vector representations that encode semantic information from natural language, serve as foundational components across diverse natural language processing (NLP) applications. These include information retrieval (IR), question answering, semantic similarity evaluation, bi-text mining, and recommendation systems. In IR pipelines, embeddings enable efficient first-stage retrieval through approximate nearest neighbor search, narrowing vast corpora to manageable candidate sets. Their role extends to retrieval-augmented generation (RAG) (Lewis et al., 2020), where they dynamically ground large language models (LLMs) in external knowledge, and to source attribution frameworks (Gao et al., 2023), enhancing the transparency of LLM outputs.

The textual embeddings for languages other than English should be treated with even more attention. It requires careful consideration of language support and data which ideally should be large-scale and domain-specific. Thus, researching embedding models for less popular or even low-resource languages helps to improve the quality of various tasks, such as article recommendation, assessing semantic similarity, information retrieval, intent recognition and many more for different communities where English is not the main spoken language. Our study focuses on embedding models in the Russian language.

Early approaches to text embeddings, such as weighted averages of static word embeddings (Pennington et al., 2014), provided rudimentary semantic signals but lacked contextual awareness. The emergence of pre-trained language models (Devlin et al., 2019) catalyzed advances like SentenceBERT (Devlin et al., 2019) and SimCSE (Gao et al., 2021), which fine-tune BERT on natural language inference (NLI) tasks to produce context-sensitive embeddings. State-of-the-art methods, including E5 (Wang et al., 2024c) and BGE (Chen et al., 2024), further scale performance through multi-stage pipelines: pre-training on weakly supervised web-scale pairs followed by task-specific fine-tuning.

However, these paradigms face critical limitations. First, their multistage workflows demand labour-intensive curation of massive relevance pairs, often restricted to narrow task domains or high-resource languages - a particular challenge for Russian, which remains underserved because of its linguistic complexity and lack of dedicated embedding models. Second, reliance on BERT-style encoders ignores breakthroughs in modern LLM architectures, such as extended context windows (Wang et al., 2024a) and parameter-efficient adaptation techniques. Third, static training data fails to leverage synthetic data generation

capabilities of instruction-tuned LLMs, constraining cross-lingual generalization, a critical gap for Russian-centric applications, where low-resource constraints and domain-specific nuances demand flexible, language-aware training paradigms.

In this work, we address these gaps by introducing GigaEmbeddings, a three-stage instruction-tuning framework built on the GigaChat-3B decoder-only LLM backbone — a member of the state-of-the-art GigaChat family<sup>1</sup>, one of the most advanced Russian-language large language models developed by Salute Devices and renowned for its robust performance on Russian NLP benchmarks. Our approach (1) synthesizes diverse, multilingual training pairs via LLM-generated queries, eliminating dependency on manually curated datasets; (2) unifies retrieval, classification, and clustering objectives through dynamic task-aware instruction tuning; and (3) integrates architectural innovations such as latent attention pooling and layer pruning to enhance efficiency. By combining synthetic data scalability with the multilingual capabilities of the GigaChat foundation, which excels in both Russian and cross-lingual tasks, GigaEmbeddings achieves state-of-the-art performance across 23 multilingual tasks on the ruMTEB benchmark. This work demonstrates the viability of decoder-only LLMs as universal encoders and establishes a new paradigm for embedding model training in low-resource and multitask settings.

Our work introduces a three-stage instruction-tuning methodology for large language models (LLMs), designed to optimize performance and efficiency through pretraining, fine-tuning, and multitask learning. The pretraining phase employs contrastive learning with large batches (16,384 samples) and a combination of in-batch and cross-batch negatives, enabling robust semantic representation learning. During fine-tuning, we leverage high-quality labeled datasets and introduce hard negatives (7 per query) to sharpen discriminative capabilities. The final stage of multitask learning integrates classification and clustering tasks, enhancing the applicability of the model. To enhance computational efficiency without sacrificing performance, we prune 25% of the original LLM’s layers, guided by insights from Gromov et al. (2025), also achieving a reduction in inference latency. We open source our model at HuggingFace.<sup>2</sup>

<sup>1</sup><https://huggingface.co/ai-sage>

<sup>2</sup><https://huggingface.co/ai-sage/Giga-Embeddings-instruct>

## 2 Related work

Recent advances in embedding models have focused on improving generalization, efficiency, and multilingual capabilities.

### 2.1 Leveraging Pretrained Language Models

Pretrained LLMs are increasingly being adopted as the backbone for embedding models, reflecting a major shift in the field. For instance, Wang et al. (2024b) adapts the Mistral decoder-only LLM (Jiang et al., 2023) through contrastive fine-tuning, achieving top performance on the MTEB benchmark (Muennighoff et al., 2023). Similarly, LLM2Vec (BehnamGhader et al., 2024) transforms decoder-only models such as LLaMA (Touvron et al., 2023) and Mistral (Jiang et al., 2023) into universal text encoders via parameter-efficient methods (Xu et al., 2023), demonstrating that causal LLMs can rival traditional encoder-only architectures when properly adapted. These approaches align with Gecko (Lee et al., 2024), which distills ranking capabilities from GPT-4 into smaller models, proving that LLM-generated relevance signals can replace human-labeled data.

### 2.2 Efficient Architectures and Long-Context Modeling

Modern embedding models optimize efficiency and long-context handling with architectural redesign. Nussbaum et al. (2024) modifies BERT with rotary positional embeddings, Flash Attention, and Dynamic NTK interpolation to scale to 8k tokens. Sturua et al. (2024) employs task-specific LoRA adapters (Xu et al., 2023) and Matryoshka Representation Learning (Kusupati et al., 2022), enabling dimension reduction (1024 to 32) without performance loss. Merrick et al. (2024) focuses on dataset stratified batching and hard negative mining to improve training efficiency.

### 2.3 Synthetic Data for Training Embedding Models

The use of synthetic data to train embedding models has emerged as a critical strategy to address data scarcity and improve generalization. Recent work shows that LLMs can generate high-quality training pairs in diverse tasks and languages. For example, Wang et al. (2024b) introduces a taxonomy-driven approach to generate task-specific synthetic data. By categorizing tasks into asymmetric (e.g. short-long query-document matches) and symmetric (e.g.

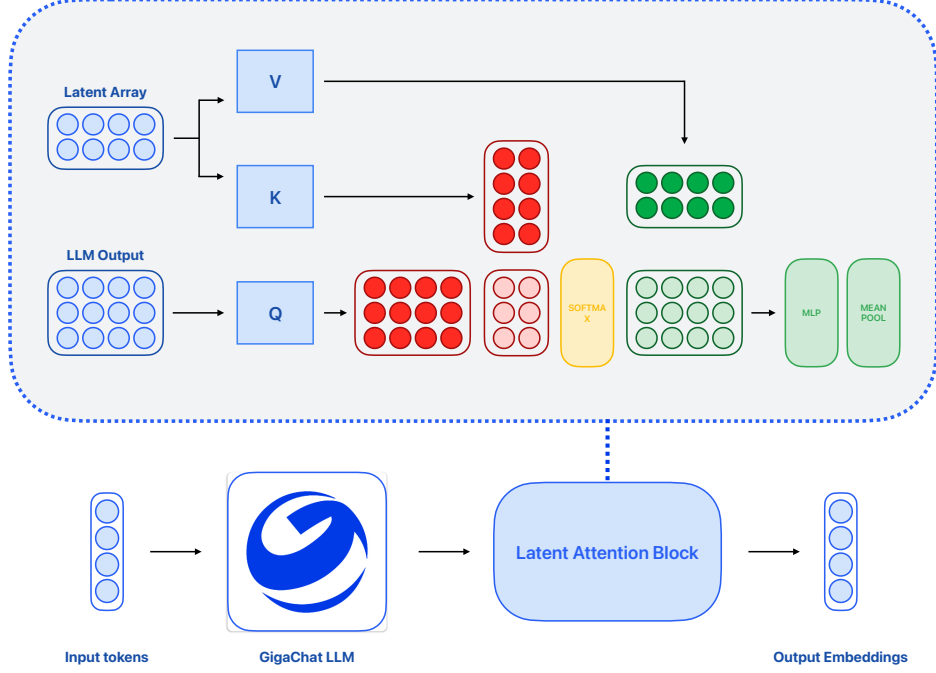


Figure 1: The design comprises a GigaChat LLM followed by a latent attention layer. This layer operates through a cross-attention mechanism where the decoder’s output acts as the query (Q), a trainable latent array provides the key (K) and value (V) inputs. The attention output is then processed by a multilayer perceptron (MLP).

semantic textual similarity) groups, the authors design structured prompt templates with randomized placeholders to maximize diversity. Their two-step prompting method: first brainstorming task definitions, then generating examples ensures coherence while scaling across 93 languages.

Further innovations refine synthetic data quality through iterative distillation and filtering. Lee et al. (2024) employs a two-step distillation process: Initial synthetic query-document pairs are generated by an LLM, followed by re-labeling hard negatives and positives using the same LLM to improve relevance signals. This approach enables Gecko to outperform larger models on the MTEB benchmark (Muennighoff et al., 2023). Similarly, Merrick et al. (2024) generates synthetic queries while grounding them with real negative documents, observing that LLMs struggle to produce high-quality negatives independently. Their hybrid strategy, combining LLM-generated queries with mined negatives, yields significant performance gains, as validated by HotpotQA (Yang et al., 2018) evaluations.

For specialized tasks such as long document and multilingual retrieval, Chen et al. (2024) augments training data by sampling lengthy articles from Wikipedia and mC4 (Raffel et al., 2023), then synthesizing questions through GPT-3.5. This task-

targeted generation mitigates data shortages in long-context scenarios while improving cross-lingual alignment.

### 3 Method

This section details our methodology, beginning with the core training objective, followed by the three-stage instruction-tuning pipeline, and concluding with architectural optimizations.

#### 3.1 Training Objective

We employ the classic InfoNCE loss (van den Oord et al., 2019) with a fixed temperature  $\tau = 0.02$ :

$$\min \mathcal{L} = -\log \frac{\phi(q, d^+)}{\phi(q, d^+) + \sum_{n_i \in \mathbb{N}} \phi(q, n_i)},$$

where  $\mathbb{N}$  denotes the set of all in-batch, cross-batch, and hard negatives, and  $\phi(q, d^+)$  is a function that computes the matching score between query  $q$  and positive document  $d^+$ . In this paper, we use the temperature-scaled cosine similarity function as follows:

$$\phi(q, d^+) = \exp\left(\frac{1}{\tau} \cos(\mathbf{h}_q, \mathbf{h}_{d^+})\right)$$

And  $\mathbf{h}_q, \mathbf{h}_{d^+}$  are vector embeddings obtained from the model.

### 3.2 Three-Stage Instruction-Tuning

The choice of the GigaChat-3B pretrain checkpoint as our backbone model reflects a deliberate balance between computational efficiency and model capability. With limited access to large-scale GPU clusters (e.g., 8xA100 80GB nodes), we prioritized a parameter scale that enables efficient fine-tuning and inference while retaining sufficient representational power for complex Russian-language tasks — a critical consideration given the scarcity of dedicated Russian embedding models. Building upon this foundation, we design a hierarchical three-stage instruction-tuning pipeline to progressively refine embeddings for diverse downstream tasks, ensuring adaptability to both low-resource constraints and multilingual applications.

#### 3.2.1 Pretraining with Large-Batch Contrastive Learning

The first stage leverages web-scale corpora in title-passage format sourced from Wikipedia, Reddit, StackExchange, and S2ORC (Lo et al., 2020), alongside mined raw text passages. To enhance query diversity, we employ an instruction-tuned LLM to synthetically generate contextually relevant queries for each passage. Training utilizes the InfoNCE loss with in-batch negatives (sampled from the same batch) and cross-batch negatives (cached from recent batches), enabling efficient contrastive learning across 16,384 samples per batch. This large batch size maximizes negative example diversity while adapting the decoder-only GigaChat LLM architecture to bidirectional attention for embedding tasks.

#### 3.2.2 Fine-Tuning with Hard Negatives

The second stage focuses on high-quality labeled datasets, including MS-MARCO (Bajaj et al., 2018), Natural Questions (NQ) (Kwiatkowski et al., 2019), SQuAD (Rajpurkar et al., 2016), MIRACL (Zhang et al., 2023), and Mr TyDi (Zhang et al., 2021), to refine retrieval-specific capabilities. We extend the InfoNCE loss by incorporating 7 curated hard negatives per query, mined via semantic similarity thresholds, alongside in-batch negatives. The batch size is reduced to 512 samples to prioritize precision over scale, ensuring robust discriminative training for challenging retrieval scenarios.

#### 3.2.3 Multitask Generalization

The final stage introduces classification and clustering tasks into the training mixture of retrieval,

classification and clustering tasks to broaden the applicability of the model. To prevent false negatives in non-retrieval tasks, we remove in-batch negatives with task-specific instruction tuning, employing a unified InfoNCE loss across all objectives. A batch size of 512 samples proves sufficient to balance computational efficiency with performance gains on classification and clustering benchmarks.

Our methodology employs an example-driven sampling strategy: positive instances are selected from the same class/cluster as the query example, while negative instances are drawn from different classes/clusters. This approach ensures intraclass cohesion and interclass distinction during training. (Lee et al., 2025)

This hierarchical approach, from large-scale pretraining to task-specialized fine-tuning and multitask generalization, ensures that the model first learns broad semantic patterns, then hones discriminative precision and finally achieves cross-task robustness.

### 3.3 Architectural Innovations

**Bidirectional Attention and Layer Pruning.** We remove causal attention masks during training to enable bidirectional context modeling. Furthermore, inspired by Gromov et al. (2025), we prune 25% of deeper transformer layers, reducing the computation with negligible quality loss.

**Latent Attention Pooling.** Inspired by Lee et al. (2025), we use a latent attention pooling head to process hidden activations to obtain the final embedding vector, since it works the best in our experiments. The model diagram is shown in Figure 1.

## 4 Experiments

### 4.1 Baselines

We compare our model with several strong baselines, according to ruMTEB benchmark.

- multilingual-E5-large-instruct (Wang et al., 2022);
- E5-Mistral-instruct (Wang et al., 2024b);
- SFR-Embedding-Mistral (Rui Meng, 2024);
- GritLM-7B (Muennighoff et al., 2025);
- BGE-M3 (Chen et al., 2024).



Model	Class.	Cluster.	MultiClass.	PairClass.	Rerank	Retrieval	STS
e5-large-instruct	66.28	63.13	41.15	<b>63.89</b>	64.35	68.23	<b>76.48</b>
e5-mistral-7b-instruct	69.07	64.24	42.93	60.81	69.96	74.19	73.71
SFR-Embedding-Mistral	69.81	64.92	42.95	60.65	70.46	-	74.31
GritLM-7B	69.92	64.3	41.96	58.93	69.99	<b>75.79</b>	74.63
BGE-M3	60.44	52.38	34.86	60.6	69.71	74.79	73.68
GigaEmbeddings	<b>72.7</b>	<b>65.36</b>	<b>51.75</b>	57.85	<b>73.42</b>	74.28	72.11

Table 1: Comparison with baselines on ruMTEB benchmark.

## 4.2 Main results

We assess the effectiveness of our model utilizing the ruMTEB benchmark (Snegirev et al., 2025) over 23 distinct tasks. Table 1 provides an overview of the average ruMTEB scores for seven sub-categories, comparing them with leading models from the ruMTEB leaderboard. Our model, referred to as GigaEmbeddings, achieves a score of 69.1, securing the top position on the ruMTEB as of December 2024 (comprehensive benchmark results can be found in Table 1). In the subsequent sections, we will detail pruning and ablation studies focusing on decisions regarding the model architecture, training methodology, and data selection strategy.

We evaluated our GigaEmbeddings against the latest front-edge embedding models using quantitative leaderboard evaluations. The e5-mistral-7b-instruct (Wang et al., 2024b) is trained with proprietary synthetic data in a single stage. Conversely, we acknowledge that retrieval tasks pose more challenges than other embedding tasks; therefore, we focus our training strategy on pre-training and fine-tuning our model for retrieval initially. Subsequently, we integrate the other subtasks into instruction-tuning within a multitask learning framework, resulting in significantly enhanced MTEB performance.

SFR-Embedding-Mistral (Rui Meng, 2024) demonstrates competitive scores on the ruMTEB benchmark by continuing to finetune the e5-mistral-7b-instruct model (Wang et al., 2024b). 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.

GritLM-7B (Muennighoff et al., 2025) — a generative representational instruction tuned language model. It unifies embeddings and text generation into a single model achieving state-of-the-art performance on both types of tasks.

BGE-M3 (Chen et al., 2024) — an embedding model, that provides a uniform support for the semantic retrieval of more than 100 languages. It can simultaneously accomplish dense retrieval, multi-vector retrieval, and sparse retrieval.

## 4.3 Pruning

We follow the findings Gromov et al. (2025) and prune 25% of the last blocks of our LLM model. We remove 9 out of 36 transformer blocks – self-attention and feed forward modules. And use this 2.5B LLM as the backbone of our embedding model. Table 2 presents the ablation.

Model	ruMTEB
GigaEmbeddings <i>full</i>	69.3
GigaEmbeddings <i>pruned</i>	69.1

Table 2: Comparison of original and pruned versions of the models.

## 4.4 Ablation studies

Since there are different types of tasks: symmetric (classification, clustering, STS) and asymmetric (retrieval, re-ranking), it was shown that the prompting model can benefit quality. We compare two ways of prompting model: prefix (Nussbaum et al., 2024, Wang et al., 2024c) and instruct (BehnamGhader et al., 2024, Lee et al., 2025). Table 3.

Prompting strategy	ruMTEB
GigaEmbeddings <i>prefix</i>	68.5
GigaEmbeddings <i>instruct</i>	69.3

Table 3: Comparison between prefix and instruct ways of model prompting.

The effectiveness of current text embedding models is substantially attributed to the use of weakly supervised contrastive pre-training (Wang et al., 2024c). Similarly to Wang et al. (2024b),

we decided to check the necessity of contrastive pre-training. We compared two ways of training models: three-stage training (with pre-training at the beginning) and two-stage training (fine-tune and multitask). Table 4.

Training strategy	ruMTEB
GigaEmbeddings w/ pre-training	69.3
GigaEmbeddings w/o pre-training	68.7

Table 4: Effectiveness of pre-training stage.

It was observed that including contrastive pre-training on weakly supervised data can further boost quality on retrieval tasks. We explain this by the fact that the model needs to reconfigure itself for the new task and also adapt to the new mechanism of encoder attention.

## 5 Conclusion

We presented GigaEmbeddings, a novel three-stage instruction-tuning framework to train high-performance text embeddings using the GigaChat-3B decoder-only LLM as its backbone. Our hierarchical pipeline, which spans large-scale contrastive pretraining, hard-negative fine-tuning, and multitask generalization, addresses critical limitations of existing methods by unifying retrieval, classification, and clustering objectives while leveraging synthetic data generation. Architectural innovations, including bidirectional attention and layer pruning (25% reduction in parameters).

Evaluation on the ruMTEB benchmark demonstrates state-of-the-art performance, with GigaEmbeddings achieving an average score of 69.1 on 23 tasks, outperforming strong baselines like e5-mistral-7b-instruct and SFR-Embedding-Mistral. Ablation studies validate the necessity of contrastive pretraining (3.4% gain in retrieval tasks with lower parameters) and the superiority of instruction-based prompting over prefix-based approaches. In particular, strategic pruning based on insights from Gromov et al. (2025) retains the full model performance while significantly enhancing efficiency.

By open-sourcing our model we provide a reproducible foundation for future research. Our work establishes decoder-only LLMs as versatile encoders and highlights the potential of synthetic data-driven training paradigms. Future directions include extending this framework to low-resource languages and integrating dynamic context window

scaling for long-document applications.

## Limitations

While GigaEmbeddings demonstrate strong performance on Russian and English tasks, several limitations merit consideration. First, the model’s multilingual capabilities are currently restricted to these two languages, as we did not evaluate its effectiveness on other languages, particularly low-resource ones. This narrow focus limits its applicability in truly global multilingual settings. Second, the computational demands of the 3B-parameter architecture and high-dimensional embeddings (e.g., 2048 dimensions) necessitate GPU acceleration, rendering deployment on CPU-only systems impractical for real-time applications. Third, while our model achieves competitive results on retrieval tasks, indicating room for improvement in dense retrieval efficiency.

These limitations highlight key directions for future work: expanding language coverage through targeted multilingual training, optimizing model size via quantization or distillation, and refining retrieval-specific architectures to bridge the performance gap.

## Acknowledgments

We would like to thank Alex Abramov, Artem Snegirev and other team members who participated in this research.

## 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*. *Preprint*, arXiv:1611.09268.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. *LLM2vec: Large language models are secretly powerful text encoders*. In *First Conference on Language Modeling*.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. *M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation*. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. [Enabling large language models to generate text with citations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore.
- Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2025. [The unreasonable ineffectiveness of the deeper layers](#). *Preprint*, arXiv:2403.17887.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. [NV-embed: Improved techniques for training LLMs as generalist embedding models](#). In *The Thirteenth International Conference on Learning Representations*.
- Jinhyuk Lee, Zhuyun Dai, and Xiaoqi Ren. 2024. [Gecko: Versatile text embeddings distilled from large language models](#). *Preprint*, arXiv:2403.20327.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K  ttler, Mike Lewis, Wen-tau Yih, Tim Rockt  schel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.
- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. [S2ORC: The semantic scholar open research corpus](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online.
- Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *arXiv preprint arXiv:2405.05374*.
- Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. [Generative representational instruction tuning](#). *Preprint*, arXiv:2402.09906.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. [MTEB: Massive text embedding benchmark](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia.
- Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. [Nomic embed: Training a reproducible long context text embedder](#). *ArXiv*, abs/2402.01613.
- 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.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Preprint*, arXiv:1910.10683.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas.
- Shafiq Rayhan Joty Caiming Xiong Yingbo Zhou Semih Yavuz Rui Meng, Ye Liu. 2024. [Sfr-embedding-mistral:enhance text retrieval with transfer learning](#). Salesforce AI Research Blog.
- Artem Snegirev, Maria Tikhonova, Anna Maksimova, Alena Fenogenova, and Alexander Abramov. 2025. [The russian-focused embedders’ exploration: rumteb benchmark and russian embedding model design](#). *Preprint*, arXiv:2408.12503.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. [jina-embeddings-v3: Multilingual embeddings with task lora](#). *Preprint*, arXiv:2409.10173.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. [Representation learning with contrastive predictive coding](#). *Preprint*, arXiv:1807.03748.

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. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. [Text embeddings by weakly-supervised contrastive pre-training](#). *Preprint*, arXiv:2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. [Improving text embeddings with large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand.

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

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. [Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment](#). *Preprint*, arXiv:2312.12148.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. [Mr. TyDi: A multi-lingual benchmark for dense retrieval](#). In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, Punta Cana, Dominican Republic.

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

## A Implementation Details

We list the hyperparameters in Table 5.

	pre-training	fine-tuning	multitask
learning rate	1e-5	1e-5	1e-5
warmup steps	1000	500	500
batch size	16K	512	512
max steps	6000	n.a.	n.a.
max length	512	512	512
epochs	n.a.	3	3
$\tau$	0.02	0.02	0.02
weight decay	0.01	0.01	0.01
hard negatives	0	7	7

Table 5: Hyperparameters for contrastive pre-training, fine-tuning and multitask stages.