The Russian-focused embedders' exploration: ruMTEB benchmark and Russian embedding model design

Artem Snegirev, Maria Tikhonova, Anna Maksimova,
Alena Fenogenova, Alexander Abramov
SaluteDevices
Correspondence: artem.s.snegirev@gmail.com

Abstract

Embedding models play a crucial role in Natural Language Processing (NLP) by creating text embeddings used in various tasks such as information retrieval and assessing semantic text similarity. This paper focuses on research related to embedding models in the Russian language. It introduces a new Russian-focused embedding model called ru-en-RoSBERTa and the ruMTEB benchmark, the Russian version extending the Massive Text Embedding Benchmark (MTEB). Our benchmark includes seven categories of tasks, such as semantic textual similarity, text classification, reranking, and retrieval. The research also assesses a representative set of Russian and multilingual models on the proposed benchmark. The findings indicate that the new model achieves results that are on par with state-of-the-art models in Russian. We release the model ru-en-RoSBERTa, and the ruMTEB framework comes with open-source code, integration into the original framework and a public leaderboard.

1 Introduction

Text embeddings play an important role in many Natural Language Processing (NLP) tasks, from clustering to semantic textual similarity (STS) and information retrieval (IR). The community has addressed this demand by releasing several powerful text embedding models (or embedders) (Wang et al., 2024, 2023a; Chen et al., 2024). However, there is still a lack of such embedders developed specifically for the Russian language. The most popular Russian-oriented models, such as rubert-tiny2 [1], SBERT$_{\text{large-nlu-ru}}$ [2], and SBERT$_{\text{large-mt-nlu-ru}}$ [3], have been released several years ago and thus do not include modern data in their training corpora. The latest models are based on an outdated version of the ruBERT (Zmitrovich et al., 2023) [4] model as a backbone. Moreover, being monolingual, they can not profit from knowledge transfer between languages.

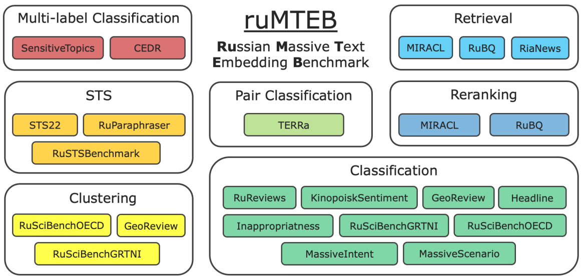

Figure 1: The scheme of the ruMTEB benchmark presenting all benchmark tasks divided into 7 task categories.

Given the usability of such text embeddings, evaluating their quality and the corresponding embedders is also important. One general approach is to evaluate text embeddings on a set of standard text embedding tasks (classification, clustering, etc.) For English, Massive Text Embedding Benchmark, or MTEB (Muennighoff et al., 2023), is considered to be a standard for such an evaluation. For Russian, however, there are significantly fewer evaluation resources. Only few tasks from MTEB contain Russian subsets, while until recently, the only embedding benchmark for Russian was enkodechka [5], which appeared several years ago and is still actively used. However, it has significantly fewer tasks than MTEB and has no tasks to evaluate the retrieval abilities of the model.

This paper addresses both problems by presenting a novel Russian-focused embedding model, also adapted for the English language, allowing knowledge transfer from this high resource lan-

[1] https://huggingface.co/cointegrated/rubert-tiny2

[2] https://huggingface.co/ai-forever/sbert_large_nlu_ru

[3] https://huggingface.co/ai-forever/sbert_large_mt_nlu_ru

[4] https://huggingface.co/ai-forever/ruBert-large

[5] https://github.com/avidale/encodechka

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 236–254
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

guage, and a new benchmark for text embedding evaluation in Russian called ruMTEB (see Figure 1 for its general structure), which contains 23 text embedding tasks in MTEB format. Among them, 17 tasks are new, and the other 6 are formed on the multilingual MTEB datasets. Thus, the contributions of our work are the following:

- we publicly release a Russian-focused text embedding model [6] adapted for the English language;
- we present the Russian version of MTEB and release 17 new Russian datasets for text embedding evaluation [7], which form the benchmark backbone, and a public leaderboard [8];
- we explore model cross-lingual transfer knowledge abilities and various model training hypotheses, which define our final training pipeline;
- we evaluate the presented model on ruMTEB and compare its performance with a set of open-source baseline solutions.

2 Related Work

2.1 Text Embedding Models

General text embedding models are widely used in various applications such as retrieval-augmented generation (RAG) (Lewis et al., 2020), STS, as well as multimodal scenarios (Radford et al., 2021). One of the first approaches for training such models was to fine-tune a pre-trained language model on the collection of labeled text pairs, such as SNLI (Bowman et al., 2015). Natural Language Inference (NLI) has been shown (Reimers and Gurevych, 2019) to help such models learn useful representations of texts for STS and other downstream applications. Recent approaches for model training utilize labeled datasets, which can be divided into symmetric (NLI, STS) and asymmetric (Retrieval) tasks. Hence, the training objective takes the form of multitask learning over one or multiple objectives, and the specialized instructions are applied for each task (Su et al., 2022).

Instead of training on limited labeled datasets, in (Wang et al., 2022a), it has been proposed to split fine-tuning into two stages: contrastive pre-training uses a large-scale pair dataset of noisy (or weakly-supervised) text examples, and contrastive fine-tuning utilize a smaller number of high-quality examples. The authors of the $\text{E5}_{\text{mistral-7b-instruct}}$ (Wang et al., 2023a) utilize an approach for model training which does not include expensive contrastive pre-training that has been shown to be useful for smaller encoder-only model XLM-R (Conneau et al., 2019). While their quality remains comparable, encoder-only models are more cost-effective during inference.

Examples of modern English-focused models include E5 (Wang et al., 2022a), BGE (Xiao et al., 2023a), GTE (Li et al., 2023), Nomic (Nussbaum et al., 2024) and Arctic Embed (Merrick et al., 2024). Scaling the number of languages supported (including Russian) has been demonstrated in mE5 (Wang et al., 2024) models and BGE-M3 (Chen et al., 2024), thereby extending their applicability in multilingual contexts. The Russian-oriented models are mainly represented by SBERT models and rubert-tiny2 and their modifications.

Models mentioned above are often used for additional fine-tuning on a specific task. To preserve the general ability of the embedding model, it has been proposed (Xiao et al., 2023b) to merge the fine-tuned model with its base model.

To address this lack of contemporary Russian-focused embedding models performing on par with their multilingual counterparts, we present **ru-en-RoSBERTa**.

2.2 Text Embedding Benchmarks

Model evaluation has always played an inevitable role in NLP progress. Starting from GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks have been a standard model evaluation method. As for text embedding representation evaluation, it has been in focus for many years and was commonly evaluated on a STS, for which yearly released SemEval (Agirre et al., 2016; Cer et al., 2017; Chen et al., 2022)[9] datasets were commonly used. Being a single dataset inevitably limits the SemEval expressivity. Following the same approach, SentEval (Conneau and Kiela, 2018), which focuses on classifier models on top of embedding, overcomes this limitation by aggregating multiple STS datasets. Still, it lacks the evaluation instruments for the suitability of embedding for retrieval

[6] https://huggingface.co/ai-forever/ru-en-RoSBERTa

[7] https://huggingface.co/collections/ai-forever/ru-mteb-6650a6a6708dc5107a9e0ba3

[8] https://huggingface.co/spaces/mteb/leaderboard

[9] https://semeval.github.io/

or clustering tasks. Due to the inefficiency of a single STS evaluation, USEB (Wang et al., 2021), focusing on reranking tasks, and BEIR (Thakur et al., 2021), aimed at zero-shot information retrieval evaluation, were created. SciRepEval (Singh et al., 2023), a multi-format benchmark for scientific document representations, includes 24 realistic tasks across four formats: classification, regression, ranking, and search. Finally, uniting and unifying all main classes of the embedding tasks MTEB (Muennighoff et al., 2023) has been proposed and is now considered a multilingual text embedding evaluation standard. Moreover, its approach was also adopted and recreated for the Scandinavian languages in the SEB (Enevoldsen et al., 2024) and Chinese in the C-MTEB (Xiao et al., 2023a)[10] benchmarks.

Most of the benchmarks mentioned above are English-focused. Even MTEB, despite being multilingual, lacks Russian-language tasks. Only few of the datasets contain Russian subsets, which is not enough for a proper embedding evaluation in this language. Apart from these few MTEB subsets, the only Russian embedding benchmark remains enkodechka, which has significantly fewer tasks than MTEB and no tasks to evaluate the model's retrieval abilities.

Still, there is a need to evaluate text embedding in Russian. To address this demand, we propose **ruMTEB** comprising a set of text embedding tasks in MTEB format.

3 ruMTEB Embedding Benchmark

3.1 Benchmark Structure and Evaluation Methodology

The ruMTEB benchmark unites 23 datasets, which can be divided into 7 task categories similar to the corresponding categories in the original MTEB benchmark: Classification (9 datasets), Clustering (3 datasets), MultiLabel Classification (2 tasks), Pair Classification (1 task), Reranking (2 tasks), Retrieval (3 tasks), and STS (3 tasks). Below each task category, the evaluation process is briefly described, and the dataset information can be found in Subsection 3.2.

Classification. The evaluation is performed in 10 consecutive experiments (bootstrap evaluation). For run, a bootstrap subset of n (by default, $n = 8$) training samples is sampled, and this down-sampled train and test parts are embedded using the embedding model. The training subset is used to train the logistic regression classifier (with 100 interactions maximum). Then, test predictions are scored using the standard Accuracy score.

Pair Classification. This group includes datasets where, given a pair of text labels, one has to predict a binary label. For evaluation, the two texts in each pair are embedded via the embedding model, and the cosine similarity between their embeddings is computed. Then, using the best binary threshold, average precision is computed.

Multi-label Classification. For evaluation, train and test sets are embedded. Then bootstrap evaluation with 10 runs is performed. In each run the training sets are down-sampled to 8 instances of each unique label. The train embeddings are used to train the kNN classifier (n_neighbours = 5). The result is evaluated on the test part using the standard Accuracy score.

Clustering. This task type includes datasets where, given a set of text fragments, one has to group them into meaningful clusters. For evaluation, text fragments are embedded. Then bootstrap evaluation with 10 runs is performed. For each run, a subset of embedding are samples, which are then clustered using K-means clustering. The result is evaluated via v-measure (Rosenberg and Hirschberg, 2007) and averaged over all experiments.

Semantic Textual Similarity (STS). Given a pair of sentences, the goal is to determine their textual similarity. Labels are continuous scores ranging from 0 to 1 (the closer to 1, the more similar). For evaluation, cosine similarity over the embedded sentences for each pair is computed. The result is evaluated with Spearman correlation (Reimers et al., 2016).

Reranking. Inputs are a query and a list of reference texts (relevant and irrelevant). The goal is to correctly rank these texts according to their relevance to the query. For evaluation, the texts for each query are ranked by the cosine similarity between the query embedding and the embedding of the given texts. The obtained ranking is scored with MAP@k ($k = 10$)[11] for each query and averaged over all queries.

Retrieval. For this task type, each dataset includes

[10] https://huggingface.co/C-MTEB

[11] The exception is MIRACLReranking which is evaluated using nDCG@10 following the original MTEB methodology.

a set of documents and queries and a mapping for each query to relevant documents. The task aims to find relevant documents for each task. For evaluation, each query document is ranked by the cosine similarity computed between the document embedding and the query embedding. The result is evaluated using nDCG@10.

3.2 Benchmark Tasks

ruMTEB comprises 23 datasets divided into 7 task types mentioned above: six datasets based on the Russian subsets from the original multilingual MTEB set (MassiveIntendClassification, MassiveScenarioClassification, MIRACLReranking, MIRACLRetrieval, RuParaphraserSTS, STS22) and 17 new datasets we release within the research. The latter are based on popular Russian time-tested and community-tested datasets.

We took the datasets based on the original MTEB without any changes. For the Russian community datasets, we selected only the tests with high-quality labeling, relying on the original publications. We performed data cleaning and automatic filtering where necessary, removed duplicates, manually verified small subsets of examples, and formatted them in the MTEB format. The main dataset information and their statics are given in Table 1, and the detailed task descriptions and preprocessing for the new sets are in Appendix A.3.

4 Text Embedding Model for Russian

This section is devoted to the text embedding model **ru-en-RoSBERTa** released within the research. We describe the training data, the base model, and the final training pipeline, motivated by the experiments described in Section 5.

4.1 Training Data

Following previous work (Wang et al., 2022a; Li et al., 2023; Nussbaum et al., 2024), we use publicly available data, high-quality and synthetic datasets to create training pairs (see Appendix A.1.1 for the full training list) [12], which, for experiment purpose (see Section 5), we divide into four groups described below.

Basic Russian Datasets. This group consists of 17 tasks. It includes pairs from SberQuAD (Efimov et al., 2020), XNLI (Conneau et al., 2018), parallel translations (Bañón et al., 2020; Tiedemann, 2012; Zhang et al., 2020), and publicly available data from various domains, such as news, blogs, QA platforms, and other Internet resources. We filter this data mostly with manual rules (see Appendix A.1.2 for the details).

Basic English Datasets. The group is formed from MEDI (Su et al., 2022) corpus without provided instructions. We also exclude instructional datasets from Super-NI (Wang et al., 2022b) and thus retain 30 datasets representing different domains and tasks. We do not apply any additional preprocessing steps.

Additional Synthetic Datasets. The group includes Query2doc MS-MARCO (Wang et al., 2023b), DINO-STS-x1x2 (Schick and Schütze, 2021), RuHNP (Malashenko et al., 2024a), entailment and contradiction pairs from RuWANLI (Malashenko et al., 2024b), and a sample of generated pairs by ruT5-base[13] model from WikiOmnia (Pisarevskaya and Shavrina, 2022). We do not change the data content and use the datasets as is.

Additional Retrieval Datasets. We use Russian and English parts of Mr. Tydi (Zhang et al., 2021) and MIRACL (Zhang et al., 2022) from BGE-M3 fine-tuning data. These datasets provide high-quality examples and are designed for the same retrieval tasks included in our benchmark.

We mine negatives similar to (Xiao et al., 2023a) using the mE5$_{\text{small}}$ (Wang et al., 2024) and sample documents by rank in the range of 20-100. For all datasets, the provided hard negatives are also used. For additional synthetic and retrieval datasets, the provided negatives are used (if available), and the rest are randomly sampled from the same dataset.

4.2 Base Model and English Language Adaptation

Since we focus on the Russian language, we use ruRoBERTa [14] (Zmitrovich et al., 2023), which has the highest scores on the classic Russian SuperGLUE (Shavrina et al., 2020) benchmark among the models of its size. In addition, we adapt it to the English language, allowing knowledge transfer from this high-resource language (see Section 5 for the corresponding experiments).

We extend the original ruRoBERTa tokenizer

[12] To avoid potential data leakage we use only the training parts of all the sets.

[13] https://huggingface.co/ai-forever/ruT5-base

[14] https://huggingface.co/ai-forever/ruRoberta-large

Task Category	**Task name**	**Data origin**	**Train**	**Val**	**Test**
Classification	GeoReviewClassification	Geo Reviews	50000	5000	5000
	HeadlineClassification	ParaPhraserPlus	36000	12000	12000
	InappropriatnessClassification	Inappropriate Sensitive Topics	4000	4000	10000
	KinopoiskSentimentClassification	Kinopoisk Movie Reviews	10500	1500	1500
	MassiveIntentClassification	MTEB	11514	2033	2974
	MassiveScenarioClassification	MTEB	11514	2033	2974
	RuReviewsClassification	RuReviews	45000	15000	15000
	RuSciBenchGRTNIClassification	RuSciBench	28476	–	2773
	RuSciBenchOECDClassification	RuSciBench	27783	–	3220
PairClassification	TERRa	TERRa	2616	307	–
MultiLabelClassification	CEDRClassification	CEDR	7529	–	1882
	SensitiveTopicsClassification	Inappropriate Sensitive topics	29178	–	2048
STS	RuSTSBenchmarkSTS	STS Benchmark	5224	1336	1264
	STS22	MTEB	–	–	265
	RuParaphraserSTS	MTEB	7227	–	1924
Clustering	GeoReviewClustering	Geo Reviews	–	–	2000
	RuSciBenchGRTNIClustering	RuSciBench	–	–	31080
	RuSciBenchOECDClustering	RuSciBench	–	–	30740
Reranking	*MIRACLReranking*	MTEB	–	44608	–
	RuBQReranking	RuBQ 2.0	–	–	1551
Retrieval	*MIRACLRetrieval*	MTEB	–	13100	–
	RiaNewsRetrieval	Ria News	–	–	10000
	RuBQRetrieval	RuBQ 2.0	–	–	2845

Table 1: The ruMTEB task outline. The **Train**, **Val**, and **Test** columns show the sizes of the dataset splits ("–" means the absence of the split). Datasets from the original MTEB benchmark are in Italic; for them, the sizes of the Russian subsets are reported.

Data Source	**Cls.**	**Clust.**	**MultiLabelCls.**	**PairCls.**	**Rerank.**	**Retr.**	**STS**	**Avg.**
Basic English Datasets	61.7	**56.6**	36.8	54.7	57.5	57.6	69.9	56.4
Basic Russian Datasets	60.0	54.2	38.0	56.3	60.8	61.7	72.3	57.6
Mixture	61.4	54.3	37.8	56.5	61.4	63.8	72.9	58.3
+ synthetic	**62.3**	54.6	**39.0**	59.7	62.1	64.1	73.6	59.3
+ synthetic & retrieval	62.1	53.9	**39.0**	**60.0**	**63.1**	**65.1**	**73.7**	**59.6**

Table 2: Different data sources impact. Model performance is measured on ruMTEB. **Avg.** stands for the average score and is computed as the mean of the category scores. The best score is put in bold, the second best is underlined.

with tokens from RoBERTa [15] (Liu et al., 2019). To learn new token embeddings, we train the model using Masked Language Modeling (MLM) objective (Devlin et al., 2018). We use the same hyperparameters as in the ruRoBERTa and the batch size of 1024. We use unique training texts from Section 4.1 and train for one epoch (~11k steps).

The whole process takes one day on two A100 80GB cards. To reduce the effect of catastrophic forgetting (Kirkpatrick et al., 2017), we merge encoder layers using spherical linear interpolation (SLERP) algorithm[16] with the factor of 0.25 to the original model. In our work, we refer to the obtained model version with the extended vocabulary as ru-en-RoBERTa.

4.3 Contrastive Fine-tuning

Following (Su et al., 2022), we perform contrastive fine-tuning for ru-en-RoBERTa on a mix of supervised and unsupervised data (from the Section 4.1). We use prefix strategy from (Reimers et al., 2023) applying prefixes for each pair to avoid a conflicting reward signal (see Appendix A.1.1 for the prefix rules and the full prefix list).

We employ the standard InfoNCE contrastive loss (Oord et al., 2018), keep a fixed temperature value of 0.02, and obtain normalized text embedding using CLS pooling. The batch is filled with pairs of the same dataset (stratified sampling), and proportional batch sampling is applied. Negative examples are formed from 7 hard negatives per query, and the remaining negatives are taken from

[15] https://huggingface.co/FacebookAI/roberta-large

[16] https://gist.github.com/dvschultz/3af50c40df002da3b751efab1daddf2c

a batch of the same device (in-batch negatives). After fine-tuning, the SLERP merging is applied to the base model with a factor of 0.1. See A.2 for the training details. We report the computational, energy, and carbon costs in Section 10.

5 Training Procedure Analysis

This section describes experiments we conducted to determine the final training pipeline. We used basic Russian, English, and additional synthetic datasets, the training approach described in Section 4 unless otherwise specified, and the full ruMTEB version for evaluation. Details on the model configurations in these experiments are given in A.2.2 and further findings are in Appendix A.6.

5.1 Cross-lingual Knowledge Transfer and Data Sources

We explored five training data configurations to study whether the model can profit from knowledge transfer between languages and various data sources. For this, we trained embedding models based on ru-en-RoBERTa: on basic English datasets only, basic Russian datasets only, and their mixture, simple or augmented with additional synthetic/synthetic+retrieval datasets. Each model is trained for 1500 steps.

Results presented in Table 2 indicate that the embedding model gets better results when trained on data in Russian and English simultaneously. Additional synthetic datasets and high-quality retrieval datasets further improve the embedding model quality despite the tasks these datasets solve already being well represented in the basic datasets. Given that in all scenarios, the number of steps is fixed, the change in the results could not account for longer training.

The model especially benefits from synthetic datasets on STS-related tasks, while quality degradation in clustering tasks remains unclear. Note that the model trained on almost all data (except the additional retrieval dataset is better by only 0.6 points. The results obtained on data in English may be due to the better quality of the tasks presented in MEDI.

5.2 English Language Adaptation

Having shown that the model can profit from cross-lingual knowledge transfer, we turned to selecting the optimal language adaptation strategy. Namely, we compared:

- *ruRoBERTa* and *XLM-R* used as baselines;
- *ru-en-RoBERTa* from subsection Section 4.2;
- *ru-en-RoBERTa w/ RetroMAE* same approach as previous where we substituted MLM with RetroMAE (Shitao et al., 2022), which proved beneficial for BGE-M3 in (Chen et al., 2024). For this configuration, set the masking ratio of decoder input tokens to 30%;
- *ru-en-RoBERTa w/o SLERP* same as *ru-en-RoBERTa* without SLERP after English language adaptation.

We perform contrastive fine-tuning for each model and then evaluate them on ruMTEB. Results (see Table 3) show that ru-en-RoBERTa outperforms both baselines by a significant margin. Additionally, the fact that XLM-R slightly outperforms ruRoBERTa may indicate that XLM-R copes better with knowledge transfer from basic English datasets, which provide diverse examples of high quality. Merging the encoder layers after language adaptation with the original model improves the model quality while using RetroMAE leads to decreased results.

5.3 Training Examples

In this series of experiments (see Table 3), we show the effects of prefixes, stratified sampling, and the number of hard negatives.

Remove prefixes. Fine-tuning the model on symmetric and asymmetric tasks simultaneously can hurt performance without instructions but improve it when instructions are used (Su et al., 2022). We found that removing prefixes consistently worsens the results, but STS-related tasks were not as affected.

Disable stratified sampling. To explore whether stratified sampling is beneficial (Merrick et al., 2024) in our case, we disabled it, used prefixes only for queries, and negatives were exchanged across devices. The latter increases the number of negatives per query to 8k. We found that the stratified version works better.

Hard negatives. To study whether adding more hard negatives (Ren et al., 2021) is beneficial, we increased their number to 15 and reduced per device batch size to 64, maintaining the total number of negative examples. To keep the same number of steps, we apply gradient accumulation. Similarly

	Cls.	**Clust.**	**MultiLabelCls.**	**PairCls.**	**Rerank.**	**Retr.**	**STS**	**Avg.**
English Language Adaptation								
XLM-R	**63.0**	**56.6**	38.7	59.6	60.7	62.6	73.9	59.3
ruRoBERTa	61.4	55.8	38.5	59.1	61.1	63.1	73.6	58.9
ru-en-RoBERTa†	62.5	55.8	39.1	60.0	62.8	65.3	73.6	**59.9**
w/ RetroMAE	62.2	55.7	37.9	59.1	60.1	61.9	72.7	58.5
w/o SLERP	62.2	55.3	38.8	59.9	62.9	64.9	73.3	59.6
Training Objective								
Additive margin	62.4	55.3	38.9	**60.9**	62.7	65.2	73.7	**59.9**
Document penalty	62.5	55.9	**40.0**	**60.9**	61.2	62.4	**74.1**	59.6
AnglE similarity	62.3	55.9	39.1	59.9	61.8	63.7	72.7	59.3
Mean pooling	62.6	55.5	38.4	59.2	61.1	63.1	72.5	58.9
Training Examples								
Increase hard negatives group	62.7	55.6	38.4	60.6	**63.1**	**65.7**	73.5	**59.9**
Disable stratified sampling	62.8	55.7	38.4	60.3	61.2	63.0	72.6	59.2
Remove prefixes	61.5	54.4	38.1	**60.9**	61.4	64.1	73.4	59.1

Table 3: Results of the model, method, and data variation. **Avg.** is the average of the category results.†The reference results for the training objective and training examples sections is model based on ru-en-RoBERTa. Each experiment changes a single component (e.g., use AnglE similarity instead of cosine). Model performance is evaluated on ruMTEB. The best score across all experiments is bold, the second best is underlined.

to (Nussbaum et al., 2024), we found that despite processing almost twice as many texts, the results did not improve.

5.4 Training Objective

In this experiment (see Table 3), we examine four modifications of the training objective described in Section 4.3.

Additive margin. Following (Yang et al., 2019), we applied an additive margin with the value of 0.01, and larger values caused convergence problems. We do not use additive margin in our final model and found that datasets are sensitive to margin values.

Document penalty. The authors of GTE (Li et al., 2023) add a penalty for query-query, document-document, and document-query matching in the denominator of InfoNCE loss that improves model performance on MTEB. We applied a similar approach to (Yang et al., 2019; Su et al., 2022), adding a penalty for document-query matching as an additional loss. The penalty significantly improved model performance on STS and worsened on Retrieval.

AnglE similarity. Normalized dot-product is usually used to score a pair of texts. AnglE similarity was proposed in (Li and Li, 2023) to optimize the angle difference of pairs in complex space. We replaced cosine similarity with AnglE. It is worth noting that in the original work AnglE was used in slightly different scenario, therefore, not finding any improvement, we left this for future work.

Mean pooling. Without an experiment, the choice of pooling strategy remains unclear. Mean pooling is used in E5, GTE and Nomic (Li et al., 2023; Nussbaum et al., 2024), BGE and Arctic Embed (Xiao et al., 2023a; Merrick et al., 2024) are apply CLS pooling. We observed consistent improvement of the latter, compared to mean pooling.

6 Evaluation

We evaluate *ru-en-RoSBERTa* and 9 publicly available embedding models for Russian, including the multilingual ones and the two instruct models, on the ruMTEB benchmark. See Table 4 for the baseline information and Appendix A.4 for other details.

We evaluate all models in the same environments and scenarios by the procedure described in 3.1. We use MTEB framework[17] for evaluation where we integrated evaluation on the new ruMTEB tasks [18][19].

[17] https://github.com/embeddings-benchmark/mteb/tree/1.14.12

[18] https://github.com/embeddings-benchmark/mteb/pull/815

[19] https://github.com/embeddings-benchmark/mteb/pull/881

Model Name	Parameters	HuggingFace Hub Link	Citation
rubert-tiny2	29.4M	cointegrated/rubert-tiny2	-
SBERT$_{\text{large-nlu-ru}}$	427M	ai-forever/sbert_large_nlu_ru	-
SBERT$_{\text{large-mt-nlu-ru}}$	427M	ai-forever/sbert_large_mt_nlu_ru	-
ru-en-RoSBERTa	404M	ai-forever/ru-en-RoSBERTa	-
mE5$_{\text{small}}$	118M	intfloat/multilingual-e5-small	Wang et al. (2024)
mE5$_{\text{base}}$	278M	intfloat/multilingual-e5-base	Wang et al. (2024)
mE5$_{\text{large}}$	560M	intfloat/multilingual-e5-large	Wang et al. (2024)
BGE-M3	567M	BAAI/bge-m3	Multi-Granularity
mE5$_{\text{large-instruct}}$	560M	intfloat/multilingual-e5-large-instruct	Wang et al. (2024)
E5$_{\text{mistral-7b-instruct}}$	7.11B	intfloat/e5-mistral-7b-instruct	Wang et al. (2023a)

Table 4: The evaluated mode description. Instruct models are marked with the corresponding suffix.

Model name	Cls.	Clust.	MultiLabelCls.	PairCls.	Rerank.	Retr.	STS	Avg.
rubert-tiny2	52.17	39.12	29.45	51.87	30.95	8.89	61.60	42.22
SBERT$_{\text{large-nlu-ru}}$	57.24	50.44	31.87	50.17	32.81	8.51	57.21	45.35
SBERT$_{\text{large-mt-nlu-ru}}$	57.52	51.29	32.67	51.97	40.56	19.13	64.40	48.72
mE5$_{\text{small}}$	56.44	51.35	31.99	55.14	65.28	65.85	69.48	57.29
mE5$_{\text{base}}$	58.26	50.27	33.65	54.98	66.24	67.14	70.16	58.34
mE5$_{\text{large}}$	61.01	52.23	36.00	58.42	69.65	74.04	71.62	61.41
BGE-M3	60.46	52.38	34.86	60.60	69.71	**74.79**	73.68	61.58
ru-en-RoSBERTa	62.74	56.06	38.88	60.79	63.89	66.52	73.97	61.77
mE5$_{\text{large-instruct}}$	66.31	63.21	41.15	**63.89**	69.17	74.41	**74.85**	66.03
E5$_{\text{mistral-7b-instruct}}$	**69.11**	**64.24**	**42.93**	60.81	**69.96**	74.19	73.71	**67.18**

Table 5: Average model results on ruMTEB task categories. The result for each category represents the mean model score on the tasks from the corresponding task types. **Avg.** stands for the average score and is computed as the mean of the task scores. The best score is put in bold, the second best is underlined.

7 Results

Table 5 shows model scores averaged within the task category, and detailed results of the task-wise model evaluation are in Appendix A.5[20].

Results analysis reveals that there is a gap between instruct and non-instruct models, mE5$_{\text{large-instruct}}$ and E5$_{\text{mistral-7b-instruct}}$ are better than their non-instruct competitors in all task categories except for Retrieval where BGE-M3 is heading the list. Moreover, while the instruct/non-instruct difference is not that significant for STS and retrieval, the advantage of the instruct models becomes obvious for other task categories.

As for the non-instruct model analysis, it can be seen that BGE-M3, ru-en-RoSBERTa, and mE5$_{\text{large}}$ perform practically on par. Moreover, ru-en-RoSBERTa performs better than its non-instruct competitors on all task categories except for 2 (Retrieval and Reranking), probably due to the absence of the contrastive pre-train. For Retrieval and Reranking, BGE-M3 and mE5$_{\text{large}}$ receive much better scores, resulting in ru-en-RoSBERTa being in the second place. The evaluation results show that ru-en-RoSBERTa is a robust embedding model suitable for various textual tasks. Additionally, unlike monolingual Russian models, the bilingual nature of ru-en-RoSBERTa allows it to be further trained or fine-tuned using the much more considerable amount of English data available.

The evaluation results positively characterize the benchmark as being complex enough for modern embedding models, allowing researchers to evaluate text embedding at a high level.

8 Conclusion

This paper introduces a new Russian-focused embedding model, which also supports English, and a new benchmark for text embedding evaluation, comprising 23 datasets divided into 7 task types. Among the benchmark datasets, 17 datasets are new and were created within this research.

We report the new embedding model architecture design, pre-training corpus, and training procedure details. We describe the datasets comprising the benchmark and propose the methodology for the text embedding evaluation on it inspired by the MTEB benchmark. We evaluate the presented

[20] This results are valid for 09.10.2024. Please, refer to the leaderboard at `https://huggingface.co/spaces/mteb/leaderboard` for the latest results.

encoding model and several baselines, thus verifying the ruMTEB complexity and performing the comparative analysis of our model results with the results of standard encoders.

9 Limitation

Model limitations. The training data for ru-en-RoSBERTa includes large segments from the Internet domain. Consequently, it contains various stereotypes and biases from English and Russian sources. Therefore, a proper model evaluation is still needed to explore their possible vulnerabilities in generalizing to out-of-domain data. The model's context is limited to a length of 512. One of the model's limitations is that due to limited computational resources, we skip the contrastive pre-training stage, leaving it for future work, although it was found (Wang et al., 2022a, 2023a) to improve the results on the retrieval-related tasks.

Lack of evaluation in English. In this work, we focus on the Russian language, and therefore, we do not conduct ru-en-RoSBERTa evaluation in English as this is beyond the scope of this work and quite resource-consuming. Nevertheless, we acknowledge that evaluating the model on the machine translation task or on the English data (e.g., the full MTEB benchmark) is valuable. We leave this to future work.

Speed and optimization. The ruMTEB benchmark comprises 23 tasks, including 6 tasks from the multilingual version. As the project is collaborative and we plan to expand the benchmark with new representative tasks, this may lead to resource-intensive and time-consuming runs. Additionally, continuously updating the benchmark makes previous model results obsolete. Due to the potential expansion of ruMTEB with new tasks and the general trend toward using larger models, there is a need to optimize the benchmark evaluation procedure.

Datasets. The collaborative nature and aggregation of the existing sets in the benchmark make it challenging to ensure uniformly high data quality across all tasks. For all benchmark datasets we checked the licenses and filtered the datasets. Unfortunately, despite the joint effort, some tasks still possess errors (e.g., incorrect labels for some examples, grammatical errors, surplus technical symbols, etc.). Moreover, there may still be biases in the data across different domains and sources, and there is still a need to extend tasks in some categories. We encourage researchers to collaborate further to fill the gaps and ensure a more comprehensive and balanced language and task representation in the benchmark.

Data leakage. All benchmark datasets are either publicly available or created using data found on the Web. This can lead to data leakage when some models trained on parts of the dataset may produce inflated scores on the benchmark. In the future, it's crucial to develop methods for automatically identifying data leakage in the task.

10 Ethical Considerations

Inference Costs. Evaluating embedding models on ruMTEB depends on its architecture and size and can be optimized with distributed inference libraries. For example, one run of ru-en-RoSBERTa of the complete evaluation experiment on a single A100 GPU 80GB takes approximately 19 hours.

Energy Efficiency and Usage. We compute the CO_2 emissions from pre-training and fine-tuning ru-en-RoSBERTa as Equation 1 (Strubell et al., 2019):

$$CO_2 = \frac{PUE * kWh * I^{CO2}}{1000} \qquad (1)$$

The power usage effectiveness (PUE) of our data centers is 1.3. The resulting CO2 emission is 3.66k kg. Model compression techniques can reduce the computational costs associated with model inference.

Potential Misuse. The ruMTEB can be used as training data for acceptability classifiers, potentially improving the quality of generated texts. We acknowledge that these improvements in text generation might lead to the misuse of LLMs for harmful purposes. The intended use of ruMTEB is for *research and development purposes*, and we are aware of the potential negative uses.

AI-assistants Help. We improve and proofread the text of this paper using Grammarly[21] to correct grammatical, spelling, and style errors and paraphrasing sentences. Thus, some segments of our publication can be potentially detected as AI-generated, AI-edited, or human-AI-generated.

[21] https://app.grammarly.com/

Acknowledgments

The authors express their sincere gratitude to all those who contributed to the success of the ruMTEB project. In particular, the authors would like to thank David Dale for his collaboration and valuable insights. The authors also acknowledge Ilya Gusev for his significant contributions to the dataset creation. Special thanks are extended to Roman Solomatin for his essential role in organizing the leaderboard.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511*. ACL (Association for Computational Linguistics).

Nikolay Babakov, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Detecting inappropriate messages on sensitive topics that could harm a company's reputation. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 26–36, Kiyv, Ukraine. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

PD Blinov, Maria Klekovkina, Eugeny Kotelnikov, and Oleg Pestov. 2013. Research of lexical approach and machine learning methods for sentiment analysis. *Computational Linguistics and Intellectual Technologies*, 2(12):48–58.

Samuel 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*, pages 632–642.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv e-prints*, pages arXiv–2402.

Xi Chen, Ali Zeynali, Chico Q Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw A Grabowicz, Scott A Hale, David Jurgens, and Mattia Samory. 2022. Semeval-2022 task 8: Multilingual news article similarity.

Yihao Chen, Xianbiao Qi, Jianan Wang, and Lei Zhang. 2023. Disco-clip: A distributed contrastive loss for memory efficient clip training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. Sberquad – russian reading comprehension dataset: Description and analysis. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 3–15. Springer International Publishing.

Kenneth Enevoldsen, Márton Kardos, Niklas Muennighoff, and Kristoffer Laigaard Nielbo. 2024. The scandinavian embedding benchmarks: Comprehensive assessment of multilingual and monolingual text embedding. *arXiv preprint arXiv:2406.02396*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Daniil Gavrilov, Pavel Kalaidin, and Valentin Malykh. 2019. Self-attentive model for headline generation. In *Proceedings of the 41st European Conference on Information Retrieval*.

Vadim Gudkov, Olga Mitrofanova, and Elizaveta Filippskikh. 2020. Automatically ranked russian paraphrase corpus for text generation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 54–59.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, et al. 2023. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

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

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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Boris Malashenko, Anton Zemerov, and Egor Spirin. 2024a. Ruhnp: Russian hard-non-paraphrases.

Boris Malashenko, Anton Zemerov, and Egor Spirin. 2024b. Ruwanli.

Nikita Martynov, Mark Baushenko, Anastasia Kozlova, Katerina Kolomeytseva, Aleksandr Abramov, and Alena Fenogenova. 2024. A methodology for generative spelling correction via natural spelling errors emulation across multiple domains and languages. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 138–155.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *arXiv preprint arXiv:2405.05374*.

Vladislav Mikhailov, Tatiana Shamardina, Max Ryabinin, Alena Pestova, Ivan Smurov, and Ekaterina Artemova. 2022. Rucola: Russian corpus of linguistic acceptability. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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.

Multi-Linguality Multi-Functionality Multi-Granularity. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *Preprint*, arXiv:2402.01613.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Dina Pisarevskaya and Tatiana Shavrina. 2022. WikiOmnia: filtration and evaluation of the generated QA corpus on the whole Russian Wikipedia. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 125–135, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Nils Reimers, Philip Beyer, and Iryna Gurevych. 2016. Task-oriented intrinsic evaluation of semantic textual similarity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 87–96.

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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. 2021. Rubq 2.0: an innovated russian question answering dataset. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*, pages 532–547. Springer.

Alexander Sboev, Aleksandr Naumov, and Roman Rybka. 2021. Data-driven model for emotion detection in russian texts. *Procedia Computer Science*, 190:637–642.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *Computing Research Repository*, arXiv:2104.07540.

Tatiana Shavrina, Alena Fenogenova, Anton Emelyanov, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. Russiansuperglue: A russian language understanding evaluation benchmark. *arXiv preprint arXiv:2010.15925*.

Xiao Shitao, Liu Zheng, Shao Yingxia, and Zhao Cao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. In *EMNLP*.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. Scirepeval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. Tsdae: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022a. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

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

Liang Wang, Nan Yang, and Furu Wei. 2023b. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023a. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. 2023b. Lm-cocktail: Resilient tuning of language models via model merging. *arXiv preprint arXiv:2311.13534*.

Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Improving multilingual sentence embedding using bidirectional dual encoder with additive margin softmax. *arXiv preprint arXiv:1902.08564*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952. IEEE Computer Society.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

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

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a miracl: Multilingual information retrieval across a continuum of languages. *arXiv preprint arXiv:2210.09984*.

Dmitry Zmitrovich, Alexander Abramov, Andrey Kalmykov, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergey Markov, et al. 2023. A family of pretrained transformer language models for russian. *arXiv preprint arXiv:2309.10931*.

A Appendix

A.1 Training Data Details

A.1.1 Training Data Information

The list of datasets included in ru-en-RoSBERTa training data and the corresponding prefix used for them are given in Table 6.

We use the following basic rules to choose a prefix:

- search_query and search_document prefixes are for answer or relevant paragraph retrieval
- clustering prefix is for asymmetric retrieval of title or summary and relevant document
- classification prefix is for symmetric paraphrasing related tasks (STS, NLI, bitext mining)

A.1.2 Data Filtration Details

We apply the following steps to the basic Russian datasets. First, texts longer than 500 tokens (ruRoBERTa-large[22] tokenizer is used) are filtered out. A small number of tokens is reserved for instructions or prefixes. Pairs from YandexQ[23], Pikabu[24], StackOverflow[25], Habr[26] and Habr QnA[27] are filtered by content popularity (e.g. views, ratings, votes). Cosine similarity obtained from LaBSE (Feng et al., 2022) is applied to filter NewsCommentary and MultiParaCrawl. We filter pairs from paraphrase-NMT-Leipzig[28] by p_good score (equivalent meaning). The XNLI is formed from entailment (relevant document) and contradiction (irrelevant negative) examples. For MIRACL, we use the title as the query and the first paragraphs (until we reach the token limit) as the document. We form pairs for Paraphrases [29] from paraphrases field, taking one as a query and the others as positive documents. The content of RuNews[30] is not changed. After exact match deduplication, the final training pairs for all datasets are randomly sampled from the remaining pairs.

A.2 Model Training Details

A.2.1 Default training details

We fine-tune the model in bf16 dtype with gradient checkpointing and use AdamW (Loshchilov and Hutter, 2017) with a learning rate of 1e-5 and weight decay of 0.01 for exactly one epoch, which is approximately 3700 steps, of which linear warmup is 200 steps. After fine-tuning, the SLERP merging is applied to the base model with a factor of 0.1.

We apply stratified sampling per device batch (mini-batch). Therefore, the global batch includes mini-batches consisting of pairs of different datasets. On the one hand, it becomes impossible to exchange negatives between devices and thus scale the number of in-batch negatives. On the other hand, this increases the diversity of sources in the global batch. Therefore, we do not apply the DisCo (Chen et al., 2023) trick to exchange

[22] https://huggingface.co/ai-forever/ruRoberta-large

[23] https://huggingface.co/datasets/IlyaGusev/yandex_q_full

[24] https://huggingface.co/datasets/IlyaGusev/pikabu

[25] https://huggingface.co/datasets/IlyaGusev/ru_stackoverflow

[26] https://huggingface.co/datasets/IlyaGusev/habr

[27] https://huggingface.co/datasets/its5Q/habr_qna

[28] https://huggingface.co/datasets/cointegrated/ru-paraphrase-NMT-Leipzig

[29] https://huggingface.co/datasets/inkoziev/paraphrases

[30] https://huggingface.co/datasets/IlyaGusev/ru_news

negative examples across devices. The batch size is 128 per device, giving 1024 documents per query. The context length is set to 512 for queries and documents (Merrick et al., 2024).

Training is conducted on a single H100 node. We utilize the BGE[31] codebase and adapt it to our experiments. PyTorch's `expandable_segments` helps us to mitigate fragmentation issues due to variable sequence length.

A.2.2 Ablation training details

Remove prefixes. We omit prefixes and keep the training process unchanged, preventing the model from identifying task types during training and inference.

Disable stratified sampling. Batch examples are randomly selected from all datasets instead of the single source. The prefixes are used only on the query side; otherwise the objective becomes easy to solve since different datasets have their own prefixes. DisCo is enabled to increase the number of negatives per query, unlike in stratified sampling.

Hard negatives. The number of hard negatives per query is increased from 7 to 15, while the batch size is reduced from 128 to 64 to maintain the same total number of negatives, and the gradient step accumulation (2 steps) is applied to keep training steps consistent.

A.3 ruMTEB Dataset Description

This section describes new tasks we present with the research and data preparation details.

A.3.1 Classification

KinopoiskSentimentClassification. In a sentiment classification dataset given a film review, one has to predict whether it is Positive, Neutral, or Negative (3 classes in total). The data was taken from the original dataset (Blinov et al., 2013)[32], which contains reviews from July 2004 to November 2012. In the preprocessing phase, we removed all mentions of the final rating from the review texts and balanced the set, leaving only 4,500 samples of each class. The resulting dataset was split into three parts (train, validation, and test), with the class balance preserved.

GeoReviewClassification. A classification dataset, where given a review text one has to predict its rating ranging from 1 to 5 (five classes in total). The set is based on the Yandex Maps[33] reviews[34]. The original dataset was balanced and split into three parts (train, validation, and test).

HeadlineClassification. In this dataset, the model needs to determine which news category the article title belongs to. The dataset was built based on ParaPhraserPlus (Gudkov et al., 2020) and contained 10,000 examples for each category, divided into train/validation/test splits of 6000, 2000, and 2000, respectively. A total of 6 classes are used: sports, incidents, politics, science, culture and economics. First, categories that contained at least 10,000 examples were selected. Other categories were discarded due to overlap between categories. For this purpose, we trained a classifier over SBERT$_{\text{large-nlu-ru}}$ embeddings.

RuReviewsClassification. A sentiment classification dataset where top-ranked goods from a major e-commerce site were provided, and user-ranked scores were used as class labels on a 5-point scale. The data was sourced from the original dataset RuReviews[35], which contains reviews in the "Women's Clothes and Accessories" category. During the preprocessing stage, duplicates were removed, and the dataset was balanced, resulting in only 25,000 samples for each class. The resulting dataset was divided into three parts (train, validation, and test) while maintaining class balance.

RuSciBenchGRTNI/OECDClassification. This is a dataset for the classification of scientific text headings. Each article has its OECD and GRNTI headings, with 29 OECD headings and 28 GRNTI headings in the dataset (e.g., Mathematics, Biological Sciences, Economics and Business, etc.). The data was sourced from the original dataset RuSciBench[36]. During preprocessing, duplicates were removed, the title and abstract were combined, and the set was balanced, leaving only the same number of samples for each class. The resulting dataset was then divided into test and training parts.

InappropriatnessClassification. The dataset aims to predict whether the message is inappropriate

[31] https://github.com/FlagOpen/FlagEmbedding

[32] https://huggingface.co/datasets/blinoff/kinopoisk

[33] https://yandex.ru/maps

[34] https://github.com/yandex/geo-reviews-dataset-2023

[35] https://github.com/sismetanin/rureviews?tab=readme-ov-file

[36] https://huggingface.co/datasets/mlsa-iai-msu-lab/ru_sci_bench

or not in the form of binary classification. The data is based on the Inappropriate Messages dataset (version 3)[37] (Babakov et al., 2021). We binarized the inappropriateness scores using the 0.5 threshold. The resulting dataset was balanced and split into three parts (train, validation, and test), with the class balance preserved.

A.3.2 Pair Classification

TERRa. The dataset was presented as one of the Russian SuperGlue tasks (Shavrina et al., 2020) and related to the Textual Entailment Recognition task. Given two texts, the task is to determine whether the meaning of one text entailed from the another text. Since the test split is hidden, we took the dev split without changes. A total of 307 examples are available.

A.3.3 Multi-Label Classification

CEDRClassification. The dataset is a task of classifying comments into five emotions (joy, sadness, surprise, fear, and anger). A total of 9,410 comments were presented from the following sources: social networks, news, and blogs. The dataset was used as is, without any modifications (Sboev et al., 2021). We took the original test split, which includes 1882 examples.

SensitiveTopicsClassification. The dataset contains sentences that can be classified into one or more sensitive topics[38] (Babakov et al., 2021). The original dataset includes 18 classes, all classes are used. Since part of the dataset is not only manually labeled, we first formed a test split from manually labeled examples, and the remaining examples were combined with semi-automatically labeled examples. We have selected the most reliable examples based on the confidence scores indicated in the examples. The final test split consists of 2048 examples and preserves the original class distribution.

A.3.4 Clustering

GeoReviewClustering. A clustering dataset based on the Yandex Maps[39] reviews[40], where given a review text one has to cluster the samples according to their rubrics or review categories (e.g., Bank, Supermarket, Pharmacy, etc.). The original dataset was balanced and split into three parts (train, validation, and test). For each review, we took its first rubric as the main label, leaving only samples corresponding to the top 100 most popular labels. This threshold limited the categories exceeding 10,000 examples. The final dataset was converted into the MTEB format.

RuSciBenchGRTNI/OECDClustering. This is a dataset for the clustering of scientific text headings. Each article has its OECD and GRNTI headings, and there are 29 OECD headings and 28 GRNTI headings in the dataset (e.g., Mathematics, Biological Sciences, Economics and Business, etc.). The data was sourced from the original dataset RuSciBench[41]. During preprocessing, duplicates were removed, the title and abstract were combined, and the set was balanced, leaving only the same number of samples for each class. The resulting dataset was then divided into test and training parts.

A.3.5 Semantic Textual Similarity (STS)

RuSTSBenchmarkSTS. The dataset used for the STS task is derived from the original multilingual STS Benchmark [42]. This multilingual set comprises various translations of the original English version of the STSbenchmark dataset, with the translations completed using deepl.com [43]. The Russian segment of the dataset was extracted and refined using the RuCoLa (Mikhailov et al., 2022) classifier [44]. In all parts of the sets (train/dev/test), instances categorized as not linguistically acceptable were excluded. Additionally, any duplicate entries were eliminated.

A.3.6 Reranking

RuBQReranking. The dataset is based on RuBQ version 2.0 (Rybin et al., 2021). The dataset contains examples of questions and paragraphs from Wikipedia. Paragraphs that answer the question are considered relevant. Paragraphs that contain the answer are used as positive documents. Negative documents are paragraphs relevant to the question's topic but not the answer. We only used questions

[37] https://github.com/s-nlp/inappropriate-sensitive-topics/blob/main/Version3/Inappapropriate_messages.csv

[38] https://github.com/s-nlp/inappropriate-sensitive-topics/blob/main/Version3/sensitive_topics.csv

[39] https://yandex.ru/maps

[40] https://github.com/yandex/geo-reviews-dataset-2023

[41] https://huggingface.co/datasets/mlsa-iai-msu-lab/ru_sci_bench

[42] https://github.com/PhilipMay/stsb-multi-mt

[43] https://www.deepl.com/ru/translator

[44] https://huggingface.co/RussianNLP/ruRoBERTa-large-rucola

from the test split with at least nine negative documents. The final test split contains 1551 examples.

A.3.7 Retrieval

RuBQRetrieval. Unique paragraphs from the dataset are used for the document bank, resulting in 56,826 documents. Documents were deduplicated while links to relevant documents were maintained. The original test split was taken without changes and has 2845 examples.

RiaNewsRetrieval. The original dataset RussiaSegodnya[45] (also known as RiaNews) consists of news articles and their headlines (Gavrilov et al., 2019). Texts are presented in lowercase format, and the capitalization of individual characters has not been changed. Since the article texts are available in HTML, we used the BeautifulSoup[46] library to clean them of markup. Additionally, texts were normalized, and extra spaces were removed. We also removed, if possible, the first sentence in each article text since it does not relate to the article's content and is a kind of meta information ("Moscow, 1 Dec — ria news."). We filtered out the texts of articles with more than 2000 characters so that models limited to a context of 512 tokens could handle the entire text. All examples were deduplicated based on the headline and text of the article. Our final dataset consists of 10,000 randomly sampled headlines as queries, and article texts (724344) are used as documents.

A.4 Experimental Setup Details

This section describes the prompt and embedding configuration we used in our experiments. Namely, we use normalized embeddings for evaluation on all ruMTEB tasks. We use pooling and instruction strategies required by the corresponding model we evaluate. Table 7 presents all the prefixes and instructions used. Specifically:

- we do not utilize any special prompts for rubert-tiny2, BGE-M3, SBERT$_{\text{large-nlu-ru}}$, and SBERT$_{\text{large-mt-nlu-ru}}$;
- we use special prefixes for ru-en-RoSBERTa;
- mE5$_{\text{small/medium/large}}$ models share the same set of prefixes;
- mE5$_{\text{large-instruct}}$ and E5$_{\text{mistral-7b-instruct}}$ models share the same set of instructions.

A.5 Detailed Results

Table 8 shows results on individual ruMTEB datasets. We run evaluation on NVIDIA A100 80GB with torch 2.2.1+cu118 and transformers 4.40.2. Please refer to PR[47] to access the results.

A.6 Additional Experimental Findings

In this part, we describe early-stage experiments that were conducted on different data subsets and different base models.

Prefixes. We found that the E5 prefixes (Wang et al., 2022a) performed slightly worse and assume that the variant we use helps to better separate tasks during training. The `clustering` prefix is more suitable for tasks where thematic identification is required, so in many classification problems, we use it instead of `classification`, despite the name. We tried adding prefixes with some probability; this improved the results without using prefixes and also worsened the results with them. In addition to stratified sampling, we implemented a sampling strategy that takes pairs with the same prefix but saw no improvement.

Losses. It was shown that Sigmoid Loss (Zhai et al., 2023) performed better at smaller batch sizes. We found that SigLIP is more sensitive to selecting the initial values of the bias and temperature parameters to achieve convergence. CoSENT loss (Li and Li, 2023) shows better results for STS-like tasks; we adapted the loss for the case with many negatives. In both cases, we were unable to achieve comparable results and left this for further work.

Augmentations. Although the model trained on 1500 steps shows comparable results to full training, we tried to apply text level and embedding level augmentations but found no meaningful performance improvement. For the text level, we used character-level augmentation from the Augmentex[48] library (Martynov et al., 2024) for both languages. In another experiment, we applied the NEFTune (Jain et al., 2023) with 3, 5, and 10 alpha parameters.

[45] https://github.com/RossiyaSegodnya/ria_news_dataset

[46] https://www.crummy.com/software/BeautifulSoup

[47] https://github.com/embeddings-benchmark/results/pull/19

[48] https://github.com/ai-forever/augmentex

Dataset	Target task	# of pairs (K)	Prefix type
Basic English Datasets (Su et al., 2022)			
AGNews	Clustering	45.0	clustering
AmazonQA	Retrieval	100.0	search_query/search_document
AmazonReviews	Clustering	100.0	clustering
CCNews	Clustering	25.0	clustering
CodeSearchNet	Clustering	15.0	clustering
ELI5	Retrieval	25.0	search_query/search_document
Fever	Retrieval	75.0	search_query/search_document
Flickr30k	STS	25.0	classification
Gooaq	Retrieval	25.0	search_query/search_document
HotpotQA	Retrieval	40.0	search_query/search_document
MedMCQA	Retrieval	30.0	search_query/search_document
MSMARCO	Retrieval	175.0	search_query/search_document
AllNLI	NLI	50.0	classification
NPR	Clustering	25.0	clustering
NQ	Retrieval	50.0	search_query/search_document
PAQ	Retrieval	25.0	search_query/search_document
PubMed	Clustering	30.0	clustering
S2ORC Title-Abstract	Clustering	100.0	clustering
SimpleWiki	STS	5.0	classification
SPECTER	STS	50.0	classification
SQuAD	Retrieval	25.0	search_query/search_document
StackExchange Duplicates	STS	25.0	classification
Trex	Retrieval	30.0	search_query/search_document
TriviaQA	Retrieval	50.0	search_query/search_document
WikiAnswers	STS	25.0	classification
WikiHow	Clustering	25.0	clustering
WoW	Retrieval	5.0	search_query/search_document
XSUM	Clustering	30.0	clustering
Yahoo Title-Answer	Retrieval	10.0	search_query/search_document
ZeroshotRE	Retrieval	15.0	search_query/search_document
Basic Russian Datasets [a]			
HabrQnA QA	Retrieval	100.0	search_query/search_document
HabrQnA Title-Body	Clustering	100.0	clustering
Habr Title-Abstract	Clustering	50.0	clustering
Paraphrases	STS	15.0	classification
MIRACL Title-Paragraph (Zhang et al., 2022)	Clustering	100.0	clustering
MultiParaCrawl (Bañón et al., 2020)	Bitext Mining	300.0	classification
NewsCommentary (Tiedemann, 2012)	Bitext Mining	25.0	classification
paraphrase-NMT-Leipzig	STS	210.0	classification
OPUS-100 (Zhang et al., 2020; Tiedemann, 2012)	Bitext Mining	175.0	classification
Pikabu Title-Body	Clustering	100.0	clustering
RuNews Title-Body	Clustering	100.0	clustering
SberQuAD (Efimov et al., 2020)	Retrieval	45.0	search_query/search_document
StackOverflow QA	Retrieval	100.0	search_query/search_document
StackOverflow Title-Body	Clustering	75.0	clustering
XNLI (Conneau et al., 2018)	NLI	125.0	classification
YandexQ QA	Retrieval	100.0	search_query/search_document
YandexQ Title-Body	Clustering	55.0	clustering
Additional Synthetic Datasets			
DINO-STS-x1x2 (Schick and Schütze, 2021)	STS	13.0	classification
Query2doc (Wang et al., 2023b)	Retrieval	500.0	search_query/search_document
RuHNP (Malashenko et al., 2024a)	STS	100.0	classification
RuWANLI (Malashenko et al., 2024b)	NLI	34.0	classification
WikiOmnia (Pisarevskaya and Shavrina, 2022)	Retrieval	100.0	search_query/search_document
Additional Retrieval Datasets			
MIRACL (Zhang et al., 2022; Multi-Granularity)	Retrieval	11.0	search_query/search_document
Mr. Tydi (Zhang et al., 2021; Multi-Granularity)	Retrieval	9.0	search_query/search_document

Table 6: The full training corpus with corresponding prefixes. We report the number of pairs in thousands. For the tasks with two different prompts, query and document, they are written with a slash.

[a] For non-cited datasets please refer to A.1.2

Task name	**ru-en-RoSBERTa**	**E5 prefix**	**E5 instruction**
Classification			
GeoreviewClassification	classification	query	Classify the organization rating based on the reviews
HeadlineClassification	clustering	query	Classify the topic or theme of the given news headline
InappropriatenessClassification	clustering	query	Classify the given message as either sensitive topic or not
KinopoiskClassification	classification	query	Classify the sentiment expressed in the given movie review text
MassiveIntentClassification	classification	query	Given a user utterance as query, find the user intents
MassiveScenarioClassification	clustering	query	Given a user utterance as query, find the user scenarios
RuReviewsClassification	classification	query	Classify product reviews into positive, negative or neutral sentiment
RuSciBenchGRNTIClassification	clustering	query	Classify the category of scientific papers based on the titles and abstracts
RuSciBenchOECDClassification	clustering	query	Classify the category of scientific papers based on the titles and abstracts
Clustering			
GeoreviewClusteringP2P	clustering	query	Identify the organization category based on the reviews
RuSciBenchGRNTIClusteringP2P	clustering	query	Identify the category of scientific papers based on the titles and abstracts
RuSciBenchOECDClusteringP2P	clustering	query	Identify the category of scientific papers based on the titles and abstracts
MultiLabelClassification			
CEDRClassification	classification	query	Given a comment as query, find expressed emotions (joy, sadness, surprise, fear, and anger)
SensitiveTopicsClassification	clustering	query	Given a sentence as query, find sensitive topics
PairClassification			
TERRa	classification	query	Given a premise, retrieve a hypothesis that is entailed by the premise
Reranking			
MIRACLReranking	search_query/search_document	query/passage	Given a question, retrieve Wikipedia passages that answer the question
RuBQReranking	search_query/search_document	query/passage	Given a question, retrieve Wikipedia passages that answer the question
Retrieval			
MIRACLRetrieval	search_query/search_document	query/passage	Given a question, retrieve Wikipedia passages that answer the question
RiaNewsRetrieval	search_query/search_document	query/passage	Given a news title, retrieve relevant news article
RuBQRetrieval	search_query/search_document	query/passage	Given a question, retrieve Wikipedia passages that answer the question
STS			
RUParaPhraserSTS	classification	query	Retrieve semantically similar text
RuSTSBenchmarkSTS	classification	query	Retrieve semantically similar text
STS22	clustering	query	Retrieve semantically similar text

Table 7: Prompts used for ruMTEB evaluation. For the tasks with two different prompts, query and document, they are written with a slash. *E5 prefix* shows prefixes used for mE5$_{\text{small/medium/large}}$ models. *E5 instruction* shows instructions used for E5$_{\text{mistral-7b-instruct}}$ and mE5$_{\text{large-instruct}}$ models.

	rubert tiny2	SBERT large-nlu-ru	SBERT large-mt nlu-ru	mE5 small	mE5 base	mE5 large	BGE- M3	ru-en- RoSBERTa	mE5 large- instruct	E5 mistral-7b- instruct
Classification										
GeoreviewClassification	39.64	39.97	39.67	44.66	46.05	49.69	48.27	49.70	55.90	**56.72**
HeadlineClassification	74.19	79.26	77.19	73.94	75.64	77.19	70.32	78.00	86.18	**87.02**
InappropriatenessClassification	58.57	62.52	64.64	59.16	58.78	61.59	59.87	61.32	65.53	**70.36**
KinopoiskClassification	49.06	49.51	50.33	49.96	50.89	56.59	58.23	63.27	66.12	**68.35**
MassiveIntentClassification	50.83	61.09	61.42	58.43	62.78	65.76	68.76	66.97	67.60	**73.74**
MassiveScenarioClassification	59.15	67.60	68.13	63.89	68.21	70.85	73.42	71.80	71.59	**77.10**
RuReviewsClassification	56.99	58.27	58.29	61.18	62.99	65.28	66.91	67.96	68.56	**70.57**
RuSciBenchGRNTIClassification	45.63	53.90	54.19	54.99	56.28	58.20	55.81	59.33	65.07	**66.05**
RuSciBenchOECDClassification	35.48	43.04	43.80	41.72	42.69	43.91	42.57	46.33	50.21	**52.11**
Clustering										
GeoreviewClusteringP2P	41.58	57.12	57.07	58.57	54.46	59.59	63.09	65.42	74.34	**76.32**
RuSciBenchGRNTIClusteringP2P	39.78	49.70	51.44	51.14	51.56	51.98	50.83	55.47	62.21	**62.27**
RuSciBenchOECDClusteringP2P	35.98	44.48	45.36	44.33	44.79	45.12	43.21	47.29	53.09	**54.13**
MultiLabelClassification										
CEDRClassification	36.87	35.84	36.81	40.07	42.32	44.84	43.47	44.69	50.01	**51.94**
SensitiveTopicsClassification	22.03	27.90	28.54	23.91	24.98	27.17	26.25	33.07	32.29	**33.92**
PairClassification										
TERRa	51.87	50.17	51.97	55.14	54.98	58.42	60.60	60.79	**63.89**	60.81
Reranking										
MIRACLReranking	15.81	18.80	24.99	59.11	60.47	63.71	**65.38**	56.91	62.49	63.61
RuBQReranking	46.09	46.81	56.14	71.45	72.01	75.60	74.03	70.87	75.84	**76.32**
Retrieval										
MIRACLRetrieval	1.89	1.98	6.20	59.01	61.60	67.33	**70.16**	53.91	66.08	67.66
RiaNewsRetrieval	13.92	11.11	21.40	70.00	70.24	80.67	82.99	78.86	**83.26**	78.94
RuBQRetrieval	10.87	12.45	29.80	68.53	69.58	74.13	71.22	66.77	73.90	**75.98**
STS										
RUParaPhraserSTS	65.14	62.06	65.17	70.46	70.17	71.82	74.90	76.16	75.40	**76.17**
RuSTSBenchmarkSTS	69.43	58.82	71.22	78.08	79.64	83.15	79.87	78.69	83.97	**84.13**
STS22	50.23	50.75	56.82	59.90	60.67	59.89	66.26	**67.06**	65.17	60.83
Average	42.22	45.35	48.72	57.29	58.34	61.41	61.58	61.77	66.03	**67.18**

Table 8: The full results of model evaluation on the ruMTEB benchmark. The aggregated score for each task category is reported in Section 7. The best score is put in bold, the second best is underlined.