A Family of Pretrained Transformer Language Models for Russian

Dmitry Zmitrovich[1], **Alexander Abramov**[1], **Andrey Kalmykov**[1],
Maria Tikhonova[1], **Ekaterina Taktasheva**[2∗], **Danil Astafurov**[1],
Mark Baushenko[1], **Artem Snegirev**[1], **Vitalii Kadulin**[1], **Sergey Markov**[1],
Tatiana Shavrina[3∗], **Vladislav Mikhailov**[4∗], **and Alena Fenogenova**[1]
[1]SaluteDevices, [2]University of Edinburgh, [3]Institute of Linguistics, RAS, [4]University of Oslo
Correspondence: alenush93@gmail.com

Abstract

Transformer language models (LMs) are fundamental to NLP research methodologies and applications in various languages. However, developing such models specifically for the Russian language has received little attention. This paper introduces a collection of 13 Russian Transformer LMs, which spans encoder (ruBERT, ruRoBERTa, ruELECTRA), decoder (ruGPT-3), and encoder-decoder (ruT5, FRED-T5) architectures. We provide a report on the model architecture design and pretraining, and the results of evaluating their generalization abilities on Russian language understanding and generation datasets and benchmarks. By pretraining and releasing these specialized Transformer LMs, we aim to broaden the scope of the NLP research directions and enable the development of industrial solutions for the Russian language.

Keywords: Russian language models, Russian language understanding, Russian language generation

1. Introduction

Transformer language models (LMs; Vaswani et al., 2017) have emerged as an essential component of state-of-the-art approaches for various natural language understanding and generation tasks. These LMs undergo pretraining in a self-supervised manner at scale on large text corpora before being adapted to a downstream task via finetuning, few-shot learning, and instruction tuning (Ruder et al., 2019; Bommasani et al., 2022; Chowdhery et al., 2022; Ouyang et al., 2022; Touvron et al., 2023). Open access to the pretrained models' weights allows the community to accelerate research and develop efficient industrial solutions (Wolf et al., 2020). However, most of these LMs are developed for English, which imposes substantial constraints on the potential of the language technologies.

The community has addressed this problem by releasing massively multilingual LMs (e.g., Conneau and Lample, 2019; Conneau et al., 2020; Liu et al., 2020b; Xue et al., 2021; Scao et al., 2023) and monolingual LMs for typologically diverse languages (e.g., Polignano et al., 2019; Le et al., 2020; Delobelle et al., 2020; Cui et al., 2020; Kutuzov et al., 2021). Nowadays, there is still a lack of Transformer LMs developed specifically for the Russian Language.

This paper introduces a family of pretrained Transformers LMs for Russian, which spans a diverse set of model architectures. We offer Russian versions of the BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2019), GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2020), and UL2 (Tay et al., 2022) models in multiple sizes. We report the development of our LMs and focus on evaluating them on a suite of standard Russian language understanding and generation datasets and benchmarks. The results show that our LMs outperform their multilingual counterparts and related Russian Transformer LMs on most tasks, achieving state-of-the-art performance. The main *contributions* are the following:

1. We pretrain and release 13 Transformer-based LMs for the Russian language: ruBERT-base[1], ruBERT-large[2], ruRoBERTa-large[3], ruELECTRA-small[4], ruELECTRA-medium[5], ruELECTRA-large[6], ruGPT-3-small[7], ruGPT-3-medium[8], ruGPT-3-large[9], ruT5-base[10], ruT5-large[11], FRED-T5-large[12], and FRED-T5-XL[13]. The LMs have been released over the last few years under the MIT license.
2. We conduct a series of experiments to evaluate the generalization abilities of our LMs on a wide range of tasks, including machine reading comprehension, natural language inference, word

∗ Work done while at SaluteDevices.

[1]hf.co/ai-forever/ruBERT-base
[2]hf.co/ai-forever/ruBERT-large
[3]hf.co/ai-forever/ruRoBERTa-large
[4]hf.co/ai-forever/ruELECTRA-small
[5]hf.co/ai-forever/ruELECTRA-medium
[6]hf.co/ai-forever/ruELECTRA-large
[7]hf.co/ai-forever/ruGPT-3-small
[8]hf.co/ai-forever/ruGPT-3-medium
[9]hf.co/ai-forever/ruGPT-3-large
[10]hf.co/ai-forever/ruT5-base
[11]hf.co/ai-forever/ruT5-large
[12]hf.co/ai-forever/FRED-T5-large
[13]hf.co/ai-forever/FRED-T5-XL

LREC-COLING 2024, pages 507–524
20-25 May, 2024.

sense disambiguation, coreference resolution, acceptability classification, inappropriateness identification, text simplification, text summarization, and text detoxification. The evaluation codebase is publicly available[14].

2. Related Work

2.1. Multilingual Language Models

Russian is well-represented in the pretraining corpus of various massively multilingual LMs, such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), RemBERT (Chung et al., 2021), mBART (Liu et al., 2020b), mT5 (Xue et al., 2021), XGLM (Lin et al., 2022), mGPT (Shliazhko et al., 2022), BLOOM (Scao et al., 2023), and mDeBERTa (He et al., 2023), *inter alia*. The multilingual LMs have significantly contributed to achieving notable results in standard NLP tasks for Russian and its related languages (Arkhipov et al., 2019). However, with the development of their monolingual counterparts (see §2.2), these LMs have primarily served as strong baselines for more complex Russian language understanding and generation tasks (e.g., Shavrina et al., 2020; Sakhovskiy et al., 2021; Mikhailov et al., 2022).

2.2. Russian Language Models

DeepPavlov (Burtsev et al., 2018) pretrained one of the first monolingual BERT-based LMs for Russian. The model configurations include (i) the RuBERT-base model pretrained on the Russian Wikipedia and news corpora (Kuratov and Arkhipov, 2019), (ii) the RuBERT-base-conversational model[15] pretrained on OpenSubtitles (Lison and Tiedemann, 2016) and social media texts, and (iii) a distilled version of RuBERT-base-conversational (Kolesnikova et al., 2022). Yandex released RuLeanALBERT[16], a Russian version of the ALBERT model (Lan et al., 2020), and YaLM-100B[17], the largest publicly available Russian LM. The LMs are pretrained on a corpus of web texts, Wikipedia articles, texts from the Taiga corpus (Shavrina and Shapovalova, 2017), and other multiple sources.

In line with these works, we have contributed to developing open-source Russian LMs of various model architectures, which are widely used within the Russian NLP community for research and development purposes (e.g., Dementieva et al., 2022; Artemova et al., 2022; Shamardina et al., 2022).

[14] github.com/ai-forever/russian-lm-evaluation

[15] hf.co/DeepPavlov/rubert-base-conversational

[16] hf.co/yandex/RuLeanALBERT

[17] hf.co/yandex/YaLM-100B

Model	Wikipedia (ru/en)	News	Books	C4	OpenSubtitles	Size
ruBERT	✓/✗	✓	✗	✗	✗	30GB
ruRoBERTa	✓/✗	✓	✓	✗	✗	250GB
ruELECTRA	✓/✗	✓	✓	✗	✓	70GB
ruGPT-3	✓/✓	✓	✓	✓	✗	450GB
ruT5	✓/✗	✓	✓	✓	✗	300GB
FRED-T5	✓/✗	✓	✓	✓	✗	300GB

Table 1: The pretraining corpus statistics.

3. Models

This section describes the model pretraining corpus, architecture design, and pretraining details.

3.1. Pretraining Corpus

Data Collection Table 1 summarizes the general statistics of our pretraining corpus. The corpus includes texts from various publicly available resources, which represent diverse domains:

- Wikipedia — a collection of general-domain texts from the Russian and English Wikipedia corpora. The Wikipedia articles are extracted from the corresponding dumps with the help of the WikiExtractor tool (Attardi, 2015).
- News — a collection of news articles from the Taiga corpus and the Lenta, Gazeta, and Interfax news sources from the corus[18] library.
- Books — a collection of literary texts from the librusec corpus (Panchenko et al., 2017) and poetic texts from the Taiga corpus. The texts are downloaded via the corus library.
- Colossal Clean Crawled Corpus (C4; Raffel et al., 2020) — a collection of web texts in Russian. The C4 data is downloaded using the Tensorflow datasets (Paper, 2021).
- OpenSubtitles — a collection of movie and TV subtitles extracted from parallel corpora.

In general, different domains and sizes of the subcorpora are included in the resulting pretraining corpora of our LMs, which range from 30GB (ruBERT) to 450GB (ruGPT-3). This variability is primarily due to multiple factors. First, our models have undergone pretraining over a few years based on methodological advancements in developing LMs and creating pretraining corpora. For instance, the ruGPT-3's C4 sub-corpus differs from the ruT5 and FRED-T5 ones in that it is filtered according to the procedure described in Ortiz Suárez et al. (2019). Second, the amount of textual data in the publicly available resources has increased over time, promoting an improved coverage of the world changes and domain representation.

[18] github.com/natasha/corus

Model	Encoder	Decoder	Objective	Parameters	# Layers	d_{model}	d_{ff}	Tokenizer	# Heads
ruBERT-base	✓	✗	MLM & NSP	178M	12	768	3072	BPE, $12 \cdot 10^4$	12
ruBERT-large	✓	✗	MLM & NSP	427M	24	1024	4096	BPE, $12 \cdot 10^4$	16
ruRoBERTa-large	✓	✗	MLM	355M	24	1024	4096	BBPE, $5 \cdot 10^4$	16
ruELECTRA-small	✓	✗	RTD	42M	12	256	1024	BPE, $256 \cdot 10^3$	4
ruELECTRA-medium	✓	✗	RTD	85M	12	576	2304	BPE, $64 \cdot 10^3$	12
ruELECTRA-large	✓	✗	RTD	427M	24	1024	4096	BPE, $120 \cdot 10^3$	16
ruGPT-3-small	✗	✓	LM	125M	12	768	3072	BBPE, $5 \cdot 10^4$	12
ruGPT-3-medium	✗	✓	LM	355M	24	1024	4096	BBPE, $5 \cdot 10^4$	16
ruGPT-3-large	✗	✓	LM	760M	24	1536	6144	BBPE, $5 \cdot 10^4$	16
ruT5-base	✓	✓	SP	222M	12	768	3072	SentencePiece, $32 \cdot 10^3$	12
ruT5-large	✓	✓	SP	737M	24	1024	4096	SentencePiece, $32 \cdot 10^3$	16
FRED-T5-large	✓	✓	MoD	820M	24	1024	2816	BBPE, $5 \cdot 10^4$	16
FRED-T5-XL	✓	✓	MoD	1.74B	24	1536	4096	BBPE, $5 \cdot 10^4$	24

Table 2: Summary of the model architecture configurations. Pretraining objectives: language modeling (LM), masked language modeling (MLM), next sentence prediction (NSP), replaced token detection (RTD), span corruption (SP), and a mixture of denoisers (MoD). d_{model} is the hidden layer dimension, and d_{ff} is the feed-forward layer dimension. **Tokenizer** is the tokenization method and the vocabulary size.

3.2. Architecture & Pretraining Details

The pretraining objectives, model architecture, scaling strategies, and other design choices for our LMs are summarized in Table 2. The model configuration choices are based on extensive empirical studies described in detail in Devlin et al. (2019); Liu et al. (2019); Clark et al. (2020); Brown et al. (2020); Tay et al. (2022), and other factors, such as availability of the data and computational resources, LM standards, and field state at a particular period of time, starting from the BERT model architecture.

3.2.1. ruBERT

Architecture ruBERT is based on BERT (Devlin et al., 2019) and pretrained on (i) a masked language modeling (MLM) objective to predict masked-out tokens in the input and (ii) a next sentence prediction (NSP) objective to predict whether two sentences follow each other. We use two BERT versions (BERT-base and BERT-large) and the Byte-pair Encoding (BPE; Wang et al., 2020) tokenization, with the vocabulary size of $12 \cdot 10^4$ tokens. The main differences between DeepPavlov's ruBERT and our ruBERT LMs are the following. First, we pretrain and release the first ruBERT-large model. Second, DeepPavlov's ruBERT models are pretrained with a small batch size on a limited number of GPUs. In contrast, we pretrain our ruBERT LMs on a similar pretraining corpus using a larger batch size and more computational resources, which results in improved model performance (see §4).

Pretraining Details We pretrain ruBERT-base and ruBERT-large with a maximum sequence length of 512 tokens using a linear scheduler with an initial learning rate of $1e{-}4$ and the Adam optimizer (Kingma and Ba, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e^{-8}$. The masking probability is 0.15. The total number of pretraining steps is 10^6. ruBERT-base is pretrained for 8 days on 16 V100 GPUs, and ruBERT-large is pretrained for 20 days on 16 V100 GPUs.

3.2.2. ruRoBERTa

Architecture We use the RoBERTa-large configuration (Liu et al., 2019) for ruRoBERTa-large. The pretraining objective is MLM, the tokenization method is Byte-level BPE (BBPE; Wang et al., 2020), and the vocabulary counts $5 \cdot 10^4$ tokens.

Pretraining Details We pretrain the model with a total batch size of 4096, the maximum sequence length of 512 tokens, a linear scheduler with an initial learning rate of $1e^{-4}$, and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e^{-8}$. The masking probability is 0.15. The model has seen 2T tokens during pretraining, which has taken 21 days on 64 V100 GPUs.

3.2.3. ruELECTRA

Architecture We use the ELECTRA architecture configurations and follow the pretraining procedure described in Clark et al. (2020). The models are pretrained with the replaced token detection (RTD) objective to predict which input tokens are masked by the MLM-based "generator". We use BPE with the vocabulary size of $256 \cdot 10^3$, $64 \cdot 10^3$, and $120 \cdot 10^3$ tokens for ruELECTRA-small, ruELECTRA-medium, and ruELECTRA-large, respectively.

Pretraining Details We pretrain the ruELECTRAmodels using the learning rate of $2e^{-4}$, the masking probability of 0.25, the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e^{-6}$, and the maximum sequence length of 512 tokens. ruELECTRA-small, ruELECTRA-medium, and ruELECTRA-large are pretrained with a batch size of 128, 64, and 48 for 7, 8, and 10 days on 4 V100 GPUs for the total number of steps of $1 \cdot 10^6$, $1 \cdot 10^6$, and $4 \cdot 10^5$, respectively.

3.2.4. ruGPT-3

Architecture ruGPT-3 is a Russian counterpart of GPT-3 (Brown et al., 2020). We use the model architecture description by Brown et al. (2020) and the GPT-2 code base (Radford et al., 2019) from the Transformers library (Wolf et al., 2020). ruGPT-3 is pretrained on the language modeling objective. We use the BBPE tokenization with the vocabulary size of $5 \cdot 10^4$ tokens.

Pretraining Details The ruGPT-3 models are pretrained with a maximum sequence length of 1024 tokens for three epochs and 2048 tokens for one epoch. We use the initial learning rate of $1e^{-4}$ and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e^{-8}$. The total number of tokens seen during pretraining is 80B. The pretraining of ruGPT3-small, ruGPT3-medium, and ruGPT3-large has taken 7, 16, and 16 days on 32, 64, and 128 V100-SXM3 GPUs, respectively.

3.2.5. ruT5

Architecture ruT5 is one of the first encoder-decoder LMs pretrained only on Russian-language textual data. ruT5 is designed analogically to T5 (Raffel et al., 2020) and is available in two model configurations: ruT5-base and ruT5-large. The models are pretrained on an MLM span corruption objective, where consecutive spans of the input tokens are masked, and the model is trained to reconstruct the masked tokens. We use the SentencePiece tokenization (Kudo and Richardson, 2018) with the vocabulary size of $32 \cdot 10^3$ tokens.

Pretraining Details The ruT5 models are pretrained using a linear scheduler with the learning rate of $1e^{-4}$ and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e^{-8}$. The sequence length is set to 512/512 for inputs and targets. The ruT5-base and ruT5-large models are pretrained with a total batch size of 2048 for 14 days on 32 V100 GPUs and 21 days on 64 V100 GPUs, respectively.

3.2.6. FRED-T5

Architecture FRED-T5 (Full-scale Russian Enhanced Denoisers) is an encoder-decoder model based on T5 and UL2 (Tay et al., 2022), available in two configurations: FRED-T5-large and FRED-T5-XL. In contrast to ruT5, FRED-T5 uses the gated GELU function instead of ReLU. Drawing inspiration from (Tay et al., 2022), we pretrain FRED-T5 on a mixture of denoisers, a set of diverse pretraining objectives. The R-Denoiser is an MLM span corruption objective used in T5. The S-Denoiser follows the language modeling objective, where the input sequence is split into the context and target tokens so that the targets do not rely on future information. The X-Denoiser aims to recover much of the input based on the span corruption and language modeling objectives.

The main differences in the pretraining approaches between UL2 and FRED-T5 are the following: (i) we use seven denoisers with a uniform distribution of the hyperparameters μ (the average span length), r (the corruption rate), and n (the number of corrupted spans) instead of the normal distribution, and (ii) we use BBPE instead of SentencePiece, with a vocabulary size of $5 \cdot 10^4$ tokens.

We use the following special tokens and hyperparameters for the FRED-T5 denoisers: <LM> ($\mu = L/4$, $r = 0.25$, $n = 1$), <SC1> ($\mu = 3$, $r = 0.15$, $n = 1$), <SC2> ($\mu = 8$, $r = 0.15$, $n = 1$), <SC3> ($\mu = 64$, $r = 0.15$, $n = 1$), <SC4> ($\mu = 3$, $r = 0.5$, $n = 1$), <SC5> ($\mu = 8$, $r = 0.5$, $n = 1$), <SC6> ($\mu = 64$, $r = 0.5$, $n = 1$), where L is the input length. The <LM> token corresponds to the S-Denoiser.

Pretraining Details FRED-T5 is pretrained using a linear scheduler with the initial learning rate of $1e^{-4}$ and the Adafactor optimizer (Shazeer and Stern, 2018) with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 1e^{-8}$. The sequence length is set to 512/512 for inputs and targets. The FRED-T5-large and FRED-T5-XL models are pretrained with a total batch size of 2048 for 35 days on 160 V100 GPUs, followed by 5 days on 80 A100 GPUs, and for 45 days on 112 A100 GPUs, respectively.

4. Empirical Evaluation

This section describes the experimental setup and presents the key results of evaluating our LMs on a suite of standard benchmarks and datasets for Russian. The optimal resulting hyperparameters are summarized in Table 10 (see §10.1).

4.1. Natural Language Understanding

4.1.1. General Language Understanding

Tasks Russian SuperGLUE (Shavrina et al., 2020) includes nine tasks on common sense understanding (RUSSE, PARus), natural language inference (TERRa, RCB), reasoning (RWSD), machine reading comprehension (MuSeRC, RuCoS; Fenogenova et al., 2020) and world knowledge (DaNetQA; Glushkova et al., 2021), and a broad-coverage diagnostic test set (LiDiRus). The **performance metrics** are the accuracy score (Acc.; PARus, TERRa, RUSSE, RWSD, RCB, and DaNetQA), exact match (EM; MuSeRC, RuCoS) the F1-score (F1; RCB, RuCoS), the macro-average F1-score (F1$_a$; MuSeRC), and the Matthews Correlation Coefficient (MCC; LiDiRus).

Model	Overall	LiDiRus MCC	RCB F1/Acc.	PARus Acc.	MuSeRC $F1_a$/EM	TERRa Acc.	RUSSE Acc.	RWSD Acc.	DaNetQA Acc.	RuCoS F1/EM
					Encoder LMs					
ruBERT-base	60.3	17.2	35.7 / 47.7	70.4	75.9 / 41.4	69.4	73.9	66.9	59.9	85.0 / 84.9
ruBERT-large	61.7	20.1	38.1 / 49.3	70.2	79.4 / 47.9	70.5	70.5	66.9	67.8	82.0 / 82.0
ruRoBERTa-large	68.1	34.1	40.9 / 46.3	76.4	84.5 / 58.1	79.3	74.9	66.9	81.1	85.0 / 85.0
ruELECTRA-small	50.5	10.6	34.6 / 46.1	56.4	62.8 / 21.0	54.0	59.2	66.9	65.8	60.0 / 59.6
ruELECTRA-medium	52.4	18.2	41.3 / 52.5	57.6	61.5 / 18.9	54.4	64.9	66.9	60.0	63.0 / 62.4
ruELECTRA-large	52.2	19.7	38.6 / 45.9	64.4	54.9 / 7.8	58.3	63.2	66.9	62.7	61.0 / 60.7
ruBERT-base (DP)*	57.6	19.9	26.5 / 45.7	54.2	77.7 / 43.3	64.8	71.4	66.9	60.1	84.0 / 84.0
ruBERT-base-conv (DP)*	50.0	17.8	45.2 / 48.4	50.8	68.7 / 27.8	64.0	72.9	66.9	60.6	22.0 / 21.8
mBERT*	54.7	8.4	34.4 / 42.2	53.2	76.8 / 41.5	57.8	65.3	66.9	62.2	80.0 / 80.4
XLM-R-large*	63.9	35.1	32.3 / 46.8	51.0	81.5 / 50.7	79.1	77.0	66.9	73.7	86.0 / 86.3
RuLeanALBERT*	69.8	40.3	36.1 / 41.3	79.6	87.4 / 65.4	81.2	78.9	66.9	76.0	90.0 / 90.2
FRED-T5-XL encoder-only*	69.4	42.1	31.1 / 44.1	80.6	88.2 / 66.6	83.1	72.3	66.9	73.5	91.0 / <u>91.1</u>
					Decoder LMs					
ruGPT-3-small	43.8	-1.3	35.6 / 47.3	56.2	65.3 / 22.1	48.8	57.0	66.9	61.0	21.0 / 20.4
ruGPT-3-medium	46.8	1.0	37.2 / 46.1	59.8	70.6 / 30.8	50.5	64.2	66.9	63.4	23.0 / 22.4
ruGPT-3-large	50.5	23.1	41.7 / 48.4	58.4	72.9 / 33.3	65.4	64.7	63.6	60.4	21.0 / 20.2
YaLM P-tune*	71.1	36.4	35.7 / 47.9	83.4	<u>89.2</u> / <u>70.7</u>	84.1	71.0	66.9	85.0	<u>92.0</u> / **91.6**
					Encoder-decoder LMs					
ruT5-base	62.3	21.3	42.5 / 47.9	57.8	80.2 / 47.1	73.0	71.3	66.9	76.9	85.0 / 84.8
ruT5-large	68.3	35.1	46.1 / 51.6	73.2	84.9 / 58.9	77.9	76.6	66.9	78.0	86.0 / 86.0
FRED-T5-large	69.0	33.8	45.0 / 48.4	72.6	88.0 / 66.4	79.6	78.0	66.9	81.7	85.0 / 84.5
FRED-T5-XL	<u>75.2</u>	<u>46.5</u>	<u>51.1</u> / <u>54.6</u>	<u>81.8</u>	**91.7** / **76.2**	<u>86.9</u>	**81.7**	66.9	<u>88.2</u>	88.0 / 88.0
mT5-base*	51.6	0.06	37.5 / 48.6	49.4	65.6 / 22.7	57.9	57.6	66.9	68.7	71.0 / 69.7
mT5-large*	56.0	17.0	34.4 / 42.7	50.4	77.6 / 42.9	67.3	56.4	66.9	74.3	74.0 / 72.8
Human	**81.1**	**62.6**	**68.0** / **70.2**	**98.2**	80.6 / 42.0	**92.0**	<u>80.5</u>	**84.0**	**91.5**	**93.0** / 89.0

Table 3: Results on Russian SuperGLUE. All values are scaled by 100. DP=DeepPavlov (Burtsev et al., 2018). **Overall** is the overall average score. The best score is in bold, and the second best is underlined. The baseline models are marked with an asterisk.

Method We estimate the model performance via finetuning and zero-shot evaluation. The encoder and encoder-decoder LMs are finetuned for a maximum of 40 epochs with an early stopping based on the task-specific performance metric or their average on the validation set. The task example templates are presented in Table 11 (see § 10.2).

- Encoder LMs: we finetune the encoders via the Transformers library using the AdamW optimizer (Loshchilov and Hutter, 2019), learning rate of $1 \cdot 10^{-5}$, weight decay of 0.01, and batch size of 32.
- Decoder LMs: the decoder-only models are evaluated in a zero-shot setting, where the target label is selected based on the lowest perplexity of the resulting prompt templates. The ruGPT-3 results are taken from the official leaderboard as of September 2023: `russiansuperglue.com/leaderboard`.
- Encoder-decoder LMs: we formulate the tasks in the text-to-text format and follow the two-stage finetuning procedure (Raffel et al., 2020). The first stage is multi-task pretraining, where the model is continuously pretrained on a combination of tasks. Each input starts with a task-specific prefix. Next, the model is finetuned on each task individually using the bf16 precision. We experiment with using the combinations of Adam & linear scheduler with a learning rate of $1 \cdot 10^{-5}$, and Adafactor & constant scheduler with the learning rate of $1 \cdot 10^{-3}$.

Baselines We finetune ruBERT-base by DeepPavlov, mBERT, mT5-base, mT5-large and XLM-R-large as described above. We also compare our LMs with the following official leaderboard results: human annotators, ruBERT-base-conversational by DeepPavlov (ruBERT-base-conv), YaLM 3.3B & P-tuning (YaLM P-tune), RuLeanALBERT, and the FRED-T5-XL encoder-only finetuned on each RSG task independently.

Results The results are shown in Table 3. FRED-T5-XL performs best on most tasks, with an overall score of 75.2. Finetuning only the FRED-T5-XL encoder leads to strong results on PARus, MuSeRC, TERRa, RUSSE, and RuCoS. ruRoBERTa-large receives the overall best performance among the proposed encoder LMs (68.1), performing on par with ruT5-large. Comparing results with the best-performing encoder, we find that ruRoBERTa-large outperforms RuLeanALBERT on RCB and DaNetQA. We also find that our ruBERT-based LMs outperform DeepPavlov's ruBERT models. ruELECTRA performs worse on the machine reading comprehension tasks, which results in a lower overall score. The overall zero-shot performance of the decoder-only LMs is similar to the ruBERT-base-conv and ruELECTRA-based LMs. The larger versions of the ruGPT-based LMs outperform the encoders on RCB, PARus, and MuSeRC (e.g., mBERT, XLM-R-large, and ruELECTRA).

Our LMs have promoted new state-of-the-art results on most of the Russian SuperGLUE tasks,

Model	Overall		In-domain		Out-of-domain	
	Acc.	MCC	Acc.	MCC	Acc.	MCC
Encoder LMs						
ruBERT-base	74.50 $\pm$ 0.60	0.41 $\pm$ 0.01	76.95 $\pm$ 0.72	0.36 $\pm$ 0.01	73.17 $\pm$ 0.74	0.43 $\pm$ 0.01
ruBERT-large	75.90 $\pm$ 0.42	0.42 $\pm$ 0.01	78.82 $\pm$ 0.57	0.40 $\pm$ 0.01	74.30 $\pm$ 0.71	0.42 $\pm$ 0.01
ruRoBERTa-large	<u>80.80</u> $\pm$ 0.47	<u>0.54</u> $\pm$ 0.01	<u>83.48</u> $\pm$ 0.45	<u>0.53</u> $\pm$ 0.01	<u>79.34</u> $\pm$ 0.57	<u>0.53</u> $\pm$ 0.01
ruELECTRA-small	61.74 $\pm$ 1.09	0.20 $\pm$ 0.02	70.09 $\pm$ 1.29	0.21 $\pm$ 0.01	56.70 $\pm$ 1.58	0.17 $\pm$ 0.03
ruELECTRA-medium	74.11 $\pm$ 0.85	0.38 $\pm$ 0.02	76.14 $\pm$ 0.88	0.34 $\pm$ 0.02	73.00 $\pm$ 1.05	0.38 $\pm$ 0.02
ruELECTRA-large	65.65 $\pm$ 0.65	0.20 $\pm$ 0.02	72.79 $\pm$ 0.31	0.22 $\pm$ 0.01	61.75 $\pm$ 1.02	0.17 $\pm$ 0.02
mBERT*	67.47 $\pm$ 1.33	0.19 $\pm$ 0.01	72.69 $\pm$ 1.40	0.19 $\pm$ 0.02	64.63 $\pm$ 1.62	0.18 $\pm$ 0.02
ruBERT-base (DP)*	72.57 $\pm$ 1.92	0.35 $\pm$ 0.12	75.02 $\pm$ 1.21	0.30 $\pm$ 0.11	71.23 $\pm$ 2.52	0.38 $\pm$ 0.12
ruBERT-base-conv (DP)*	75.33 $\pm$ 1.55	0.38 $\pm$ 0.02	78.98 $\pm$ 0.79	0.38 $\pm$ 0.01	73.33 $\pm$ 2.08	0.38 $\pm$ 0.04
RuLeanALBERT*	80.00 $\pm$ 0.0	0.52 $\pm$ 0.0	82.00 $\pm$ 0.0	0.49 $\pm$ 0.0	78.00 $\pm$ 0.0	0.52 $\pm$ 0.0
XLM-R*	65.73 $\pm$ 2.33	0.17 $\pm$ 0.04	74.17 $\pm$ 1.75	0.22 $\pm$ 0.03	61.13 $\pm$ 2.9	0.13 $\pm$ 0.05
RemBERT*	76.21 $\pm$ 0.33	0.44 $\pm$ 0.01	78.32 $\pm$ 0.75	0.40 $\pm$ 0.02	75.06 $\pm$ 0.55	0.44 $\pm$ 0.01
Decoder LMs (PenLP)						
ruGPT-3-small	53.89 $\pm$ 0.0	0.25 $\pm$ 0.0	57.46 $\pm$ 0.0	0.19 $\pm$ 0.0	51.94 $\pm$ 0.0	0.27 $\pm$ 0.0
ruGPT-3-medium	55.79 $\pm$ 0.0	0.27 $\pm$ 0.0	59.39 $\pm$ 0.0	0.19 $\pm$ 0.0	53.82 $\pm$ 0.0	0.30 $\pm$ 0.0
ruGPT-3-large	56.83 $\pm$ 0.0	0.29 $\pm$ 0.0	61.22 $\pm$ 0.0	0.22 $\pm$ 0.0	54.43 $\pm$ 0.0	0.31 $\pm$ 0.0
mGPT-XL*	60.60 $\pm$ 0.0	0.27 $\pm$ 0.0	62.84 $\pm$ 0.0	0.16 $\pm$ 0.0	59.37 $\pm$ 0.0	0.29 $\pm$ 0.0
Encoder-decoder LMs						
ruT5-base	71.26 $\pm$ 1.31	0.27 $\pm$ 0.03	76.49 $\pm$ 1.54	0.33 $\pm$ 0.03	68.41 $\pm$ 1.55	0.25 $\pm$ 0.04
ruT5-large	74.29 $\pm$ 3.80	0.37 $\pm$ 0.07	74.82 $\pm$ 1.67	0.33 $\pm$ 0.29	74.00 $\pm$ 5.33	0.40 $\pm$ 0.10
FRED-T5-large	75.83 $\pm$ 0.0	0.40 $\pm$ 0.0	77.36 $\pm$ 0.0	0.34 $\pm$ 0.0	75.0 $\pm$ 0.0	0.42 $\pm$ 0.0
FRED-T5-XL	77.37 $\pm$ 0.0	0.46 $\pm$ 0.0	80.5 $\pm$ 0.0	0.46 $\pm$ 0.0	75.66 $\pm$ 0.0	0.45 $\pm$ 0.0
Human	**84.08**	**0.63**	**83.55**	**0.57**	**84.59**	**0.67**

Table 4: Results for acceptability classification on the RuCoLA test set. The best score is in bold, and the second-best one is underlined. The baseline models are marked with an asterisk.

and the overall performance gap between humans and the LMs has been narrowed by up to 4.9. However, there is still room for model improvement on the RWSD, RCB, TERRa, and PARus tasks.

4.1.2. Acceptability Classification

Task RuCoLA (Mikhailov et al., 2022) consists of in-domain sentences from linguistic publications and out-of-domain sentences produced by generative LMs. The task is to predict if a given sentence is acceptable or not. The **performance metrics** are the accuracy score (Acc.) and MCC.

Method We follow the finetuning and evaluation procedure described in Mikhailov et al. (2022). We use the ruRoBERTa-large, ruGPT-3-medium, and ruT5-base results from Mikhailov et al. (2022). The best model configuration is selected based on the MCC on the validation set.

- Encoder LMs: the encoders (ruBERT, ruELECTRA) are finetuned for 5 epochs using the AdamW optimizer via a grid search over a set of hyperparameters: the learning rates $\{10^{-5}, 3 \cdot 10^{-5}, 5 \cdot 10^{-5}\}$ and the weight decay values $\{10^{-4}, 10^{-2}, 0.1\}$. The results are averaged over 10 experiment runs with different random seeds.
- Decoder LMs: the ruGPT-3-small and ruGPT-3-large models are evaluated using a classification approach based on a threshold for the PenLP acceptability measure (Lau et al., 2020). The threshold is selected on the training set via 10-fold cross-validation to maximize MCC on the validation set: -19.65 (ruGPT-3-small), -20.91 (ruGPT-3-medium), and -19.39 (ruGPT-3-large).
- Encoder-decoder LMs: ruT5-large is finetuned for 20 epochs, with the search space of $\{10^{-4}, 10^{-3}\}$ for the learning rate and $\{0, 10^{-4}\}$ for the weight decay. We finetune the FRED-T5 models for 20 epochs using the Adafactor optimizer, the learning rate of $5 \cdot 10^{-4}$, weight decay of 0.0, and batch size of 16. We report the results for only one experiment run.

Baselines We finetune ruBERT-base by DeepPavlov, ruBERT-base-conv, and mBERT as described above. The PenLP threshold for mGPT-XL[19] is -54.37. We use the results for human annotators, XLM-R, and RemBERT from Mikhailov et al. (2022). Results for RuLeanALBERT are from the RuCoLA leaderboard as of September 2023: `rucola-benchmark.com/leaderboard`.

Results The results for acceptability classification are presented in Table 4. In general, our LMs outperform their monolingual and multilingual counterparts. ruRoBERTa-large receives the best performance among the LMs, falling short behind expert human annotators. The second-best is RuLeanALBERT, followed by FRED-T5-XL and RemBERT. At the same time, ruELECTRA outperforms mBERT and XLMR. We observe that ruGPT-3-large performs the best among the threshold-based classifiers, and the ruGPT-3-medium performance is

[19] `hf.co/ai-forever/mGPT`

Model	F1-score
Encoder LMs	
ruBERT-base	80.75 ± 0.32
ruBERT-large	81.27 ± 0.34
ruRoBERTa-large	<u>82.44</u> ± 1.02
ruELECTRA-small	78.46 ± 0.77
ruELECTRA-medium	79.05 ± 0.43
ruELECTRA-large	80.27 ± 1.30
mBERT*	78.24 ± 0.56
ruBERT-base (DP)*	79.59 ± 0.07
ruBERT-base-conv (DP)*	81.14 ± 0.64
Decoder LMs	
ruGPT-3-small	64.68 ± 0.0
ruGPT-3-medium	64.32 ± 0.0
ruGPT-3-large	64.39 ± 0.0
mGPT-XL*	64.78 ± 0.0
Encoder-decoder LMs	
ruT5-base	75.45 ± 0.0
ruT5-large	75.20 ± 0.0
FRED-T5-large	82.13 ± 0.0
FRED-T5-XL	**82.86** ± 0.0
mT5-base*	75.63 ± 0.0
mT5-large*	77.33 ± 0.0

Table 5: Results for inappropriateness identification. DP=DeepPavlov (Burtsev et al., 2018). The best score is in bold, and the second best is underlined. The baseline models are marked with an asterisk.

similar to mGPT 1.3B. Our LMs generalize well to machine-generated sentences, showing minor performance differences between the in- and out-of-domain sets.

4.1.3. Inappropriateness Identification

Task We use the dataset by Babakov et al. (2021) to evaluate the model's ability to identify inappropriate messages, which can cover a sensitive topic (e.g., crime, body shaming, and sexism) and harm the reputation of the user. The target **performance metric** is the macro-average F1-score.

Method We finetune and evaluate the encoder, decoder, and encoder-decoder LMs as described in §4.1.2. The PenLP thresholds are -37.66 (ruGPT-3-small), -35.82 (ruGPT-3-medium), and -35.39 (ruGPT-3-large).

Baselines We finetune and evaluate mBERT, ruBERT-base by DeepPavlov, ruBERT-base-conv, mT5-base, and mT5-large as described in §4.1.2. The PenLP threshold for mGPT-XL is -32.54.

Results The results for inappropriateness identification are presented in Table 5. Overall, all models receive strong performance, and the encoder and decoder-only LMs perform on par. The performance improves with the model scaling, except for the decoder-only and ruT5 models. FRED-T5-XL shows the best results among the LMs, followed by ruRoBERTa-large and FRED-T5-large.

Model	Public test		Private test	
	SARI	BERTScore	SARI	BERTScore
Decoder LMs				
ruGPT-3-small	37.96	0.81	37.54	0.79
ruGPT-3-medium	39.00	0.91	39.21	0.91
ruGPT-3-large	39.09	0.90	39.37	0.90
mGPT-XL*	42.45	0.98	42.22	0.97
Encoder-decoder LMs				
ruT5-base	43.34	<u>1.0</u>	43.29	<u>1.0</u>
ruT5-large	43.33	<u>1.0</u>	43.22	<u>1.0</u>
FRED-T5-large	<u>43.95</u>	0.99	43.40	0.99
FRED-T5-XL	43.41	<u>1.0</u>	43.35	0.99
mBART-large-50*	39.75	0.95	40.47	0.96
mT5-base*	43.63	0.99	43.55	0.99
mT5-large*	43.62	<u>1.0</u>	43.68	<u>1.0</u>
Input sentence*	43.90	<u>1.0</u>	<u>43.92</u>	<u>1.0</u>
Human	**66.72**	0.82	**66.11**	0.82

Table 6: Results for text simplification on the RuSimpleSentEval-2021 test sets. The best score is in bold, and the second best one is underlined. The baseline models are marked with an asterisk.

4.2. Natural Language Generation

4.2.1. Text Simplification

Task RuSimpleSentEval-2021 (Sakhovskiy et al., 2021) is a corpus of pairs of sentences comprising complex sentences and their simplified versions. The task is to rewrite the input sentence in a less complicated way. The **performance metrics** are SARI (Xu et al., 2015) and BERTScore (Zhang et al., 2020) computed between the input and the output using mBERT.

Method We finetune the decoder and encoder-decoder LMs using the AdamW optimizer, the learning rate of $5 \cdot 10^{-5}$, and batch size of 2 for 3 and 10 epochs, respectively. The decoding strategy and hyperparameters for inference are selected based on the validation performance and manual analysis of the model outputs. The resulting strategy is beam search with 5 beams for all models.

Baselines We report human reference scores and a non-neural baseline of the input sentence without any change (Input sentence). Then, following the procedure described above, we finetune mBART-large-50 (Tang et al., 2021), mGPT-XL, mT5-base, and mT5-large.

Results The results for the text simplification task are presented in Table 6. For all tested models except for ruGPT3-small, BERTScore exceeds 0.9, which means that simplified predictions are very close to the input sentence with slight simplifications, mainly at the word level. Overall, our manual analysis of the model outputs suggests that the target metric (SARI) does not indicate the intended

Model	ROUGE-L	BERTScore	BLEU	METEOR	ChrF1
		Decoder LMs			
ruGPT-3-small	17.28	71.78	6.18	20.13	30.66
ruGPT-3-medium	19.27	72.37	6.89	21.81	32.72
ruGPT-3-large	19.66	72.62	7.24	22.39	33.37
		Encoder-decoder LMs			
ruT5-base	18.72	73.15	7.42	22.78	33.17
ruT5-large	20.12	73.53	8.11	23.9	34.59
FRED-T5-large	22.48	73.69	8.35	24.29	34.97
FRED-T5-XL	**22.95**	**73.9**	**8.61**	**24.72**	**35.36**
mBART-large-50*	18.53	72.58	7.46	22.63	34.95
mT5-base*	17.76	71.96	6.16	20.45	30.95
mT5-large*	17.80	72.73	7.16	21.84	33.38

Table 7: Results for text summarization on Gazeta. The best score is in bold, second best is underlined. The baseline models are marked with an asterisk.

performance. For instance, the multilingual LMs (mT5 and mBART-large-50) tend to copy most parts of the input, which results in high BERTScore (over 0.96) and strong SARI scores. At the same time, SARI does not always improve with the model scaling. We also find that encoder-decoder LMs outperform decoder-only LMs, and ruT5-base leaves the input sentence unchanged, similar to mT5 and mBART-large-50. The results indicate that it is necessary to conduct a human-based evaluation to get a more complete picture of the model performance.

4.2.2. Text Summarization

Task Gazeta (Gusev, 2020) is a corpus of news articles and their summaries for abstractive summarization. The **performance metrics** are standard summarization evaluation metrics: ROUGE-L (Lin, 2004), BERTScore, BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ChrF1 (Popović, 2015).

Method We finetune the decoder-only models for 3 epochs using AdamW optimizer, a linear scheduler with a warmup, and a learning rate of $5 \cdot 10^{-5}$. The encoder-decoder models are finetuned with Adafactor with a constant learning rate of $1 \cdot 10^{-3}$. We examine different generation strategies and hyperparameters on the validation set. The resulting strategy is beam search with 5 beams for all LMs.

Baselines We finetune mBART-large-50, mT5-base, and mT5-large as described above.

Results The results for text summarization are shown in Table 7. The scores demonstrate that the performance improves as the model size increases. ruGPT-3-large achieves the highest scores among the decoder LMs, and FRED-T5-XL receives the best performance among the encoder-decoder LMs. The manual analysis of the model outputs indicates that the ruGPT-3 models tend to copy parts of the inputs, while the ruT5 and FRED-T5 models produce more plausible summaries. Overall, our LMs show higher scores as opposed to their multilingual counterparts.

Model	STA	SIM	FL	Joint	ChrF1
		Decoder LMs			
ruGPT-3-small	74.0	80.2	83.5	50.4	51.8
ruGPT-3-medium	78.0	79.8	83.6	53.1	54.0
ruGPT-3-large	75.4	81.4	82.6	50.8	55.5
		Encoder-decoder LMs			
ruT5-base	80.0	81.9	83.0	55.3	57.2
ruT5-large	78.8	81.6	83.2	54.4	56.8
FRED-T5-large	81.9	81.8	84.8	57.8	57.6
FRED-T5-XL	82.3	82.1	85.3	**58.5**	58.1
mBART-large-50*	81.4	77.5	79.7	51.5	53.6
mT5-base*	61.5	86.4	83.1	42.8	54.9
mT5-large*	77.4	84.5	86.1	56.7	56.9
Duplicate*	24.0	**100.0**	**100.0**	24.0	56.0
Delete*	55.8	88.7	85.2	40.6	52.6
Human	**85.0**	72.0	78.0	49.0	**77.0**

Table 8: Results for detoxification. Performance metrics: STA=Style Transfer Accuracy, SIM=Content Similarity, FL=Fluency. The best score is in bold, second best is underlined. The baseline models are marked with an asterisk.

4.2.3. Text Detoxification

Task The RUSSE Detoxification corpus (Dementieva et al., 2022) tests the model's capability of generating a detoxified version of the toxic text. The **performance metrics** are based on Dementieva et al. (2022): ChrF1 score, style transfer accuracy, content similarity, fluency, and the "Joint" score (multiplication of last three metrics).

Method We conduct finetuning of the LMs over five epochs using AdamW for the rGPT-based models and Adafactor for the ruT5-based models. We experiment with multiple decoding strategies on the validation set, analyzing the performance metrics and conducting manual analysis of the outputs. We use beam search with 5 beams and the repetition penalty of 1.05 at the inference stage.

Baselines. We report human reference scores and baseline results provided by Dementieva et al. (2022): (i) a trivial "Duplicate" baseline, which leaves the original text intact and acts as a lower performance threshold; (ii) a "Delete" baseline, which removes toxic words based on a predefined vocabulary. Additionally, we finetune and evaluate mBART-large-50, mT5-base, and mT5-large with the same parameters as the LMs above.

Results The text detoxification results are presented in Table 8. The scores show that the LMs demonstrate a significant performance improvement over the baselines when considering the "Joint" score and surpass human performance with regard to text similarity and fluency. The performance difference between the decoder-only and encoder-decoder LMs is not substantial. However, the encoder-decoder LMs perform better, with FRED-T5-XL achieving the highest Joint score (58.5) and the best model ChrF1 score (58.1).

5. Conclusion

This paper introduces 13 Russian Transformer LMs of various model architectures, pretraining objectives, and model sizes. We have released our LMs over the last few years, facilitating research advancements and the development of specialized downstream solutions for the Russian language. We provide a report on the model architecture design, pretraining corpus, and pretraining. We empirically evaluate our LMs, their multilingual counterparts, and other open-source Russian LMs on standard Russian NLP benchmarks and datasets. The results indicate that our LMs promote state-of-the-art performance on Russian SuperGLUE and RuCoLA and match the human performance on the machine reading comprehension and text detoxification tasks. We outline the following future work research directions that are out of the scope of this paper: (i) analyzing the model performance when finetuning data is limited, (ii) exploring the effect of pretraining corpus composition, (iii) other techniques for adapting language models to Russian, such as initializing from a multilingual LM, (iv) conducting a more optimal hyperparameter search, and (v) performing a human-based generation evaluation. We aim to continue to develop novel Russian LMs in the future.

6. Limitations

Limited Context Size Although our generative LMs achieve strong results and promote state-of-the-art performance on various tasks, their context window size (maximum 2048 tokens) limits the model application on long-context tasks. We leave experiments with efficient finetuning approaches to extending the context size for future work (e.g., Chen et al., 2023).

Social Bias Evaluation The evaluation experiments conducted in this paper do not – and de facto cannot – address all possible scenarios. We aim to assess our model generalization abilities on standard academic datasets and benchmarks, covering various natural language understanding and generation tasks. Still, our experimental setup is limited due to the lack of peer-reviewed resources for specific evaluation cases, such as detecting social biases, stereotypes, and hate speech. Therefore, before deploying our LMs, developers should perform safety evaluations for their specific model application scenarios.

Language Generation Evaluation The performance metrics for natural language generation tasks do not always capture the task-specific properties (e.g., Fomicheva and Specia, 2019; Colombo et al., 2022; Chhun et al., 2022). Our manual analysis of the model outputs confirms these findings for the text simplification task (see §4.2.1). While we follow the evaluation approach based on a combination of standard performance metrics of different types, these metrics may not comprehensively evaluate the model generation abilities. We suggest a human-based side-by-side model evaluation may help get a complete picture of the performance.

Domain Shifts Our LMs' pretraining corpus features various domains, including general domain, news, books, web texts, and subtitles. However, pretraining the LMs[20] on different sub-corpora can hinder their performance in domain-specific applications and on out-of-domain data. Nevertheless, we empirically show that our LMs receive strong performance on domains not well represented in the pretraining corpus, ranging from linguistic publications (§4.1.2) to user messages (§4.1.3).

7. Ethical Considerations

The development of the new LMs detailed in this paper adheres to standard ethical guidelines. We advocate for these models' responsible and impartial utilization, carefully considering their potential societal impacts. Special attention is given to filtering harmful content and ensuring a diverse range of perspectives and sources are included in the model pretraining corpora. Furthermore, we recognize the importance of ongoing vigilance in monitoring and addressing the unintended consequences of deploying these models in real-world applications.

Possible Misuse We believe that our research should not be involved in creating content that somehow affects the individual or communal well-being, including (i) legislative application or censorship, (ii) disinformation, infringement of the rights of access to information, (iii) dehumanizing, misrepresenting, or otherwise harmful representations of people or their religions, culture, belief, (iv) promoting harmful or discriminatory content.

[20] Recall that our LMs have been pretrained over the last several years, and the domain choice and sub-corpora sizes are based on multiple factors (see §3.1).

Model	CO_2 (kg)
Encoder LMs	
ruBERT-base	1.17k
ruBERT-large	2.94k
ruRoBERTa-large	12.37k
ruELECTRA-small	0.25k
ruELECTRA-medium	0.29k
ruELECTRA-large	0.36k
Encoder-decoder LMs	
ruT5-base	4.12k
ruT5-large	12.37k
FRED-T5-large	55.7k
FRED-T5-XL	52.7k
Decoder LMs	
ruGPT-3-small	2.06k
ruGPT-3-medium	9.43k
ruGPT-3-large	16.94k

Table 9: CO_2 emissions of pretraining models.

Biases and data quality The pretraining data for some of the presented models includes large segments from the internet domain and, consequently, contains various stereotypes and biases. Therefore, proper model evaluation is still needed to explore their possible vulnerabilities in generalizing to the out-of-domain data.

Energy Efficiency and Usage We compute the CO_2 emissions from pretraining our LMs as Equation 1 (Strubell et al., 2019):

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

The power usage effectiveness (PUE) of our data centers is 1.3. The CO_2 emissions in kg are presented in Table 9. Model compression techniques and parameter-efficient finetuning methods can reduce the computational costs associated with model inference. Note that while the ruELECTRA models underperform the baselines on some natural language understanding tasks (e.g., machine reading comprehension), these LMs are highly efficient due to their size (e.g., the small and medium versions have 42M and 85M, respectively). We recommend the user conduct their own evaluation for a downstream task of interest accounting for both performance and efficiency.

8. Bibliographical References

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Giuseppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the Opportunities and Risks of Foundation Models.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. DeepPavlov: Open-Source Library for Dialogue Systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models.

Cyril Chhun, Pierre Colombo, Fabian M. Suchanek, and Chloé Clavel. 2022. Of Human Criteria and Automatic Metrics: A Benchmark of the Evaluation of Story Generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5794–5836, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Pierre Colombo, Maxime Peyrard, Nathan Noiry, Robert West, and Pablo Piantanida. 2022. The Glass Ceiling of Automatic Evaluation in Natural Language Generation.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. *Advances in neural information processing systems*, 32.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Marina Fomicheva and Lucia Specia. 2019. Taking MT Evaluation Metrics to Extremes: Beyond Correlation with Human Judgments. *Computational Linguistics*, 45(3):515–558.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing.

D Kinga, Jimmy Ba Adam, et al. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5, page 6. San Diego, California;.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization.

Alina Kolesnikova, Yuri Kuratov, Vasily Konovalov, and Mikhail Burtsev. 2022. Knowledge Distillation of Russian Language Models with Reduction of Vocabulary.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-Scale Contextualised Language Modelling for Norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How Furiously Can Colorless Green Ideas Sleep? Sentence Acceptability in Context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pre-training for French. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot Learning with Multilingual Language Models.

Qi Liu, Matt J. Kusner, and Phil Blunsom. 2020a. A Survey on Contextual Embeddings.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020b. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020c. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Brian W. Matthews. 1975. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochimica et biophysica acta*, 405 2:442–51.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, Online. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 Technical Report.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipelines for Processing Huge Corpora on Medium to Low Resource Infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

David Paper. 2021. TensorFlow Datasets. *State-of-the-Art Deep Learning Models in TensorFlow: Modern Machine Learning in the Google Colab Ecosystem*, pages 65–91.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. 2019. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6. CEUR.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving Language Understanding by Generative Pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:1–67.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer Learning in Natural Language Processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-Shot Learners Go Multilingual.

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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and fine-tuning. *arXiv preprint arXiv:2008.00401*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual Translation from Denoising Pre-Training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *Advances in Neural Information Processing Systems*, 30.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural Machine Translation with Byte-Level Subwords. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9154–9160.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations (ICLR)*.

9. Language Resource References

Ekaterina Artemova, Maxim Zmeev, Natalia Loukachevitch, Igor Rozhkov, Tatiana Batura, Vladimir Ivanov, and Elena Tutubalina. 2022. RuNNE-2022 Shared Task: Recognizing Nested Named Entities.

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.

Daryna Dementieva, Varvara Logacheva, Irina Nikishina, Alena Fenogenova, David Dale, Irina Krotova, Nikita Semenov, Tatiana Shavrina, and Alexander Panchenko. 2022. RUSSE-2022: Findings of the First Russian Detoxification Shared Task Based on Parallel Corpora.

Alena Fenogenova, Vladislav Mikhailov, and Denis Shevelev. 2020. Read and Reason with MuSeRC and RuCoS: Datasets for Machine Reading Comprehension for Russian. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6481–6497, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Taisia Glushkova, Alexey Machnev, Alena Fenogenova, Tatiana Shavrina, Ekaterina Artemova, and Dmitry I Ignatov. 2021. DaNetQA: a yes/no question answering dataset for the russian language. In *Analysis of Images, Social Networks and Texts: 9th International Conference, AIST 2020, Skolkovo, Moscow, Russia, October 15–16, 2020, Revised Selected Papers*, pages 57–68. Springer.

Ilya Gusev. 2020. Dataset for Automatic Summarization of Russian News. In *Artificial Intelligence and Natural Language: 9th Conference, AINL 2020, Helsinki, Finland, October 7–9, 2020, Proceedings 9*, pages 122–134. Springer.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

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*, pages 5207–5227, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexander Panchenko, Dmitry Ustalov, Nikolay Arefyev, Denis Paperno, Natalia Konstantinova, Natalia Loukachevitch, and Chris Biemann. 2017. *Human and Machine Judgements for Russian Semantic Relatedness*, pages 221–235. Springer International Publishing, Cham.

Andrey Sakhovskiy, Alexandra Izhevskaya, Alena Pestova, Elena Tutubalina, Valentin Malykh, Ivan Smurov, and Ekaterina Artemova. 2021. RuSimpleSentEval-2021 Shared Task: Evaluating Sentence Simplification for Russian. In *Proceedings of the International Conference "Dialogue 2021"*, pages 607–617.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the The RuATD Shared Task 2022 on Artificial Text Detection in Russian.

Tatiana Shavrina, Alena Fenogenova, Emelyanov Anton, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4717–4726, Online. Association for Computational Linguistics.

Tatiana Shavrina and Olga Shapovalova. 2017. To the Methodology of Corpus Construction for Machine Learning: "Taiga" Syntax Tree Corpus and Parser. In *Proceedings of "CORPORA-2017" International Conference*, pages 78–84.

10. Appendix

10.1. Hyperparameter Values

Model	Optimizer	Learning Rate	Weight Decay	Batch Size
Russian SuperGLUE				
Encoder LMs	AdamW	$1 \cdot 10^{-5}$	0.01	32
Decoder LMs	✗	✗	✗	✗
Encoder-decoder LMs (I)	Adafactor	$1 \cdot 10^{-3}$	15	16
Encoder-decoder LMs (II)	Adam	$1 \cdot 10^{-5}$	20	16
RuCoLA				
ruBERT-base	AdamW	$3 \cdot 10^{-5}$	$1e^{-4}$	64
ruBERT-large	AdamW	$3 \cdot 10^{-5}$	0.1	64
ruBERT-base (DP)	AdamW	$3 \cdot 10^{-5}$	0.01	64
ruBERT-base-conv (DP)	AdamW	$1 \cdot 10^{-5}$	0.01	32
mBERT	AdamW	$1 \cdot 10^{-5}$	0.1	32
ruRoBERTa-large	AdamW	10^{-5}	10^{-4}	32
ruELECTRA-small	AdamW	$5 \cdot 10^{-5}$	0.1	32
ruELECTRA-medium	AdamW	$5 \cdot 10^{-5}$	0.1	32
ruELECTRA-large	AdamW	$3 \cdot 10^{-5}$	10^{-4}	32
ruT5-base	Adafactor	10^{-4}	0	128
ruT5-large	Adafactor	10^{-4}	0	128
FRED-T5-large	Adafactor	$5 \cdot 10^{-4}$	0	16
FRED-T5-XL	Adafactor	$5 \cdot 10^{-4}$	0	16
Inappropriateness Identification				
ruBERT-base	AdamW	$1 \cdot 10^{-5}$	0.1	64
ruBERT-large	AdamW	$1 \cdot 10^{-5}$	0.1	16
ruBERT-base (DP)	AdamW	$1 \cdot 10^{-5}$	0.1	64
ruBERT-base-conv (DP)	AdamW	$1 \cdot 10^{-5}$	0.01	64
mBERT	AdamW	$3 \cdot 10^{-5}$	0.01	32
ruRoBERTa-large	AdamW	10^{-5}	10^{-4}	32
ruELECTRA-small	AdamW	$5 \cdot 10^{-5}$	10^{-3}	64
ruELECTRA-medium	AdamW	$5 \cdot 10^{-5}$	0.01	64
ruELECTRA-large	AdamW			
ruT5-base	Adafactor	10^{-4}	0	128
ruT5-large	Adafactor	10^{-4}	0	128
FRED-T5-large	Adafactor	$5 \cdot 10^{-4}$	0	16
FRED-T5-XL	Adafactor	$5 \cdot 10^{-4}$	0	16
Text Simplification				
Decoder LMs	AdamW	$1 \cdot 10^{-5}$	0	2
Encoder-decoder LMs	AdamW	$1 \cdot 10^{-5}$	0	2
Text Detoxification				
Decoder LMs	AdamW	$5 \cdot 10^{-5}$	0.01	2
Encoder-decoder LMs	Adafactor	$1 \cdot 10^{-4}$	0.01	8
Text Summarization				
Decoder LMs	AdamW	$5 \cdot 10^{-5}$	0.01	4
Encoder-decoder LMs	Adafactor	$1 \cdot 10^{-3}$	0.01	2

Table 10: Optimal hyperparameter values found in the experiments. I/II=finetuning stage. DP=DeepPavlov (Burtsev et al., 2018).

10.2. Russian SuperGLUE Templates

<table>
<tr><th>Model</th><th>Format</th><th>Labels</th></tr>
<tr><td colspan="3">LiDiRus</td></tr>
<tr><td>ruRoBERTa</td><td><s> {premise} </s></s> {hypothesis} </s></td><td>entailment | not_entailment</td></tr>
<tr><td>ruBERT</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>entailment | not_entailment</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>entailment | not_entailment</td></tr>
<tr><td>ruT5</td><td>lidirus premise: {premise} hypothesis: {hypothesis}</td><td>entails | doesn’t entail</td></tr>
<tr><td>FRED-T5</td><td>lidirus premise: {premise} hypothesis: {hypothesis}</td><td>entails | doesn’t entail</td></tr>
<tr><td colspan="3">RCB</td></tr>
<tr><td>ruRoBERTa</td><td><s> {premise} </s></s> {hypothesis} </s></td><td>entailment | contradiction | neutral</td></tr>
<tr><td>ruBERT</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>entailment | contradiction | neutral</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>entailment | contradiction | neutral</td></tr>
<tr><td>ruT5</td><td>rcb premise: {premise} hypothesis: {hypothesis}</td><td>entailment | contradiction | neutral</td></tr>
<tr><td>FRED-T5</td><td>rcb premise: {premise} hypothesis: {hypothesis}</td><td>entailment | contradiction | neutral</td></tr>
<tr><td colspan="3">PARus</td></tr>
<tr><td>ruRoBERTa</td><td><s> {premise} </s></s> {hypothesis} </s></td><td>0 | 1</td></tr>
<tr><td>ruBERT</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruT5</td><td>parus premise: {premise} hypothesis1: {choice1} hypothesis2: {choice2}</td><td>hypothesis1 | hypothesis2</td></tr>
<tr><td>FRED-T5</td><td>parus premise: {premise} hypothesis1: {choice1} hypothesis2: {choice2}</td><td>hypothesis1 | hypothesis2</td></tr>
<tr><td colspan="3">MuSeRC</td></tr>
<tr><td>ruRoBERTa</td><td><s> {passage} </s></s> {question} {answer} </s></td><td>0 | 1</td></tr>
<tr><td>ruBERT</td><td>[CLS] {passage} [SEP] {question} {answer} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {passage} [SEP]{question} {answer} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruT5</td><td>muserc question: {question} answer: {answer} text: {passage}</td><td>no | yes</td></tr>
<tr><td>FRED-T5</td><td>muserc question: {question} answer: {answer} text: {passage}</td><td>no | yes</td></tr>
<tr><td colspan="3">TERRa</td></tr>
<tr><td>ruRoBERTa</td><td><s> {premise} </s></s> {hypothesis} </s></td><td>entailment | not_entailment</td></tr>
<tr><td>ruBERT</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>entailment | not_entailment</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {premise} [SEP] {hypothesis} [SEP]</td><td>entailment | not_entailment</td></tr>
<tr><td>ruT5</td><td>terra premise: {premise} hypothesis: {hypothesis}</td><td>entails | doesn’t entail</td></tr>
<tr><td>FRED-T5</td><td>terra premise: {premise} hypothesis: {hypothesis}</td><td>entails | doesn’t entail</td></tr>
<tr><td colspan="3">RUSSE</td></tr>
<tr><td>ruRoBERTa</td><td><s> {sentence1} </s></s> {sentence2} </s></s> {word} </s></td><td>True | False</td></tr>
<tr><td>ruBERT</td><td>[CLS] {sentence1} [SEP] {sentence2} [SEP]</td><td>True | False</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {sentence1} [SEP] {sentence2} [SEP]</td><td>True | False</td></tr>
<tr><td>ruT5</td><td>russe sentence1: {sentence1} sentence2: {sentence2} slovo: {word}</td><td>no | yes</td></tr>
<tr><td>FRED-T5</td><td>russe sentence1: {sentence1} sentence2: {sentence2} slovo: {word}</td><td>no | yes</td></tr>
<tr><td colspan="3">RWSD*</td></tr>
<tr><td>ruRoBERTa</td><td>False</td><td></td></tr>
<tr><td>ruBERT</td><td>False</td><td></td></tr>
<tr><td>ruELECTRA</td><td>False</td><td></td></tr>
<tr><td>ruT5</td><td>False</td><td></td></tr>
<tr><td>FRED-T5</td><td>False</td><td></td></tr>
<tr><td colspan="3">DaNetQA</td></tr>
<tr><td>ruRoBERTa</td><td><s> {passage} </s></s> {question} </s></td><td>0 | 1</td></tr>
<tr><td>ruBERT</td><td>[CLS] {passage} [SEP] {question} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {passage} [SEP] {question} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruT5</td><td>danetqa question: {question} text: {passage}</td><td>no | yes</td></tr>
<tr><td>FRED-T5</td><td>danetqa question: {question} text: {passage}</td><td>no | yes</td></tr>
<tr><td colspan="3">RuCoS</td></tr>
<tr><td>ruRoBERTa</td><td><s> {passage} </s></s> {query.replace(’@placeholder’, entities[i])} </s></td><td>0 | 1</td></tr>
<tr><td>ruBERT</td><td>[CLS] {passage} [SEP] {query.replace(’@placeholder’, entities[i])} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruELECTRA</td><td>[CLS] {passage} [SEP] {query.replace(’@placeholder’, entities[i])} [SEP]</td><td>0 | 1</td></tr>
<tr><td>ruT5</td><td>rucos question: {query} entities: {’, ’.join(entities)}</td><td>{entities[i]}</td></tr>
<tr><td>FRED-T5</td><td>danetqa question: {question} text: {passage}</td><td>{entities[i]}</td></tr>
</table>

Table 11: Example templates for the RussianSuperGLUE tasks. * – due to the task complexity, we submit the majority baseline for the RWSD task as our best performing model.