# Read and Reason with MuSeRC and RuCoS: Datasets for Machine Reading Comprehension for Russian

**Alena Fenogenova[1], Vladislav Mikhailov[1,2], Denis Shevelev[1]**
[1] Sberbank / Moscow, Russia
[2] National Research University Higher School of Economics / Moscow, Russia
{Fenogenova.A.S, Mikhaylov.V.Nikola, Shevelev.D.M}@sberbank.ru

## Abstract

The paper introduces two Russian machine reading comprehension (MRC) datasets, called MuSeRC and RuCoS, which require reasoning over multiple sentences and commonsense knowledge to infer the answer. The former follows the design of MultiRC, while the latter is a counterpart of the ReCoRD dataset. The datasets are included in RussianSuperGLUE, the Russian general language understanding benchmark. We provide a comparative analysis and demonstrate that the proposed tasks are relatively more complex as compared to the original ones for English. Besides, performance results of human solvers and BERT-based models show that MuSeRC and RuCoS represent a challenge for recent advanced neural models. We thus hope to facilitate research in the field of MRC for Russian and prompt the study of multi-hop reasoning in a cross-lingual scenario.

## 1 Introduction

Machine reading comprehension (MRC) is a central task in natural language processing that simulates a human ability to read a text and provide the correct answer for a given question. Therefore, it requires general language understanding, knowledge about the world, and interpretive reasoning. The task has been widely explored for the English language targeting different aspects of reading comprehension (Hermann et al., 2015; Hill et al., 2015; Rajpurkar et al., 2016; Trischler et al., 2016; Joshi et al., 2017; Rajpurkar et al., 2018). Recently, the paradigm has shifted towards a more complex setting, where the model is tested to infer the answer based on interpretive analysis of text (Lai et al., 2017), reasoning over multiple documents (Yang et al., 2018) or joint natural language inference and commonsense reasoning (Zellers et al., 2018). However, MRC in languages other than English, specifically Russian, has not been well-addressed primarily due to the lack of high-quality and large-scale datasets. Recognizing this need, several cross-lingual machine reading datasets have been constructed (Asai et al., 2018; Liu et al., 2019; Lewis et al., 2019b; Artetxe et al., 2019; Clark et al., 2020) with a few of them being a part of cross-lingual benchmarks such as XGLUE (Liang et al., 2020) and XTREME (Hu et al., 2020). Though allowing to evaluate the current state of language transferring methods, these cover only a small number of languages, use a back-translation technique for the assembly or combine data from different annotation schemes.

Surprisingly, little prior research has been devoted to the task of MRC for Russian, and no attempts have been made to explore the subject in multi-hop and commonsense reasoning scenarios. SberQuAD (Efimov et al., 2019) is the only one Russian MRC dataset designed as a competition challenge analogous to SQuAD (Rajpurkar et al., 2016). The task is formalized as an extractive reading comprehension, where the answer to a natural language question is a text span from the corresponding Wikipedia passage. Still, a significant portion of the questions can be answered by matching the language patterns between the question and the passage segment that contains the answer.

To this end, we introduce two novel Russian MRC datasets, called RuCoS and MuSeRC, which require commonsense knowledge and reasoning over multiple sentences. The datasets are publicly available and





*Proceedings of the 28th International Conference on Computational Linguistics*, pages 6481–6497
Barcelona, Spain (Online), December 8-13, 2020

included in the general language understanding benchmark RussianSuperGLUE[1] analogous to SuperGLUE for English (Wang et al., 2019). The datasets follow the design of SuperGLUE tasks, namely ReCoRD (Zhang et al., 2018) and MultiRC (Khashabi et al., 2018). We provide a thorough comparative analysis of the Russian and English datasets and demonstrate that in some aspects the proposed datasets are more complex as opposed to the original tasks, particularly due to the language specifics.

The remainder is organized as follows. Section 2 and Section 3 outline the description of RuCoS and MuSeRC datasets, particularly task abstraction, dataset collection procedure, annotation procedure and comparative analysis. Section 4 describes TF-IDF & BERT-based baselines, evaluation of human performance, and comparison to the corresponding SuperGLUE tasks. We highlight the main contributions of this work and conclude in Section 6.

## 2 RuCoS

### 2.1 Task Description

**Russian Reading Comprehension with Commonsense Reasoning (RuCoS)** is a large-scale dataset for Russian MRC which consists of 87,027 samples. The majority of the samples require general language understanding and commonsense reasoning to arrive at the answer. Each sample includes an excerpt from a news article, a cloze-style query with a missing named entity (a placeholder), a set of named entities that are possible answers to the query, and a set of referents to the answer entity. The task is framed as the selection of one of the candidate referents that best fit the placeholder. To achieve this, the model is supposed to behave like a human: to read a text describing an event and fill the placeholder in the query based on text understanding, commonsense knowledge, and deduction of the plausible consequences or details of the event. Notably, the answer entity may be expressed by an abbreviation, an acronym, or a set of surface forms. Thus, the task requires understanding of rich inflectional morphology and lexical variability of Russian. Consider an example in Figure 1 and find the original version in Appendix A.1. The *passage* is a textual segment from a news article, and underlined italics correspond to named entities in the passage which are possible answers. The *query* is a consequent statement supported by the passage. The model is required to find the missing entity (or one of its referents) that best fits the placeholder.

**Passage:** The mother of two boys who were abandoned by their father at *Moscow's Sheremetyevo airport* has taken them. This was reported to *TASS* by the press service of the *Ministry of education and science of the Khabarovsk territory*. Now the younger child attends kindergarten, and the older one goes to school. In educational institutions, full-time psychologists work with them as necessary. Also, the *Ministry of social protection of the population* is considering the issue of free health improvement for children in the summer. A few days after *Viktor Gavrilov* abandoned his children at the airport, he turned himself in to investigators in the city of *Bataysk*, *Rostov region*.

**Query:** On January 26, <placeholder> abandoned his sons, aged five and seven, in Sheremetyevo.

**Correct Entities:** Viktor Gavrilov

Figure 1: An automatically translated sample from RuCoS.

### 2.2 Data Collection

We adapted the ReCoRD methodology to assemble RuCoS as follows. (1) First, we curated news articles from publicly available resources, namely Lenta news dataset[2] and Deutsche Welle[3] website. We then

[1] https://russiansuperglue.com/

[2] https://github.com/yutkin/lenta.ru-news-dataset

[3] https://www.dw.com/ru/

applied BERT-based NER model provided by DeepPavlov (Burtsev et al., 2018) to extract mentions of entities in the articles. (2) Next, we automatically generated passages, queries, and answers from the obtained articles. Each passage is the first few paragraphs of a news article which usually summarize and describe the news event. We enriched the passages with top-3 titles of related articles using cosine similarity between the passage and the titles over TF-IDF vectors. The titles provide an additional context or complementary summary points. The remainder of the article was split into sentences using rusenttokenize[4], a rule-based sentence segmenter for Russian. A sentence is considered to be a query if it contains at least one named entity that is mentioned in the passage. The named entity is further to be replaced with a placeholder to generate a cloze-style query. Besides, the sentence should satisfy a number of criteria defined in (Zhang et al., 2018). We use a morphological analyzer pymorphy2 (Korobov, 2015) to identify references of the answer entity in the passage based on lemma intersection. The result of this stage is a set of passage-query-answer triples. (3) Furthermore, we computed the IPM frequency of each passage using the New Frequency Vocabulary of Russian Words[5]. We averaged IPM values of each token lemma in the sentence (if present in the vocabulary) over a total number of token lemmas. We discarded triples that contain passages of the IPM frequency lower than 0.7. (4) Finally, we filtered out the generated triples with two MRC models for Russian, particularly R-Net (Wang et al., 2017) and RuBERT (Kuratov and Arkhipov, 2019) fine-tuned on SberQuAD. The models are released as a part of DeepPavlov framework[6]. We excluded all triples correctly answered by the models. Thus, the triples obtained at this step contain only those queries that represent a challenge for the existing Russian reading comprehension models.

The resulted triples were randomly split into train and dev sets. We obtained additional 8K samples from Lenta[7] website for the test set to eliminate potential cheating and data leakage. Each set was balanced by the source of news. We now describe the annotation procedure.

### 2.3 Annotation Procedure

Since the previous steps are fully automated, the resulted triples may contain errors obtained with any preprocessing tool or include an incomplete set of the answer referents due to the high lexical variability of Russian. Hence, we conducted the human filtering procedure using the Russian crowd-sourcing service Yandex.Toloka[8]. Due to limited resources, only dev and test samples were validated by the crowd-workers. The crowd-workers were required to: (1) successfully complete a training pool of 10 assignments, (2) have the user rating of more than 60%, and (3) spend at least 30 seconds on each assignment. To ensure the high quality of the annotation procedure, we manually annotated a set of 200 control tasks. The control tasks are used to discard those annotators whose quality performance on the tasks is lower than 50%. Besides, we set the dynamic overlap of 3 performers, i.e. each assignment was completed by at least 3 crowd-workers in full accordance with the above-mentioned requirements.

Appendix A.1 outlines the crowd-sourcing assignment interface. First, the crowd-workers had to complete the training pool to better understand the task. The task instruction could be accessed at any time. Each assignment consists of a passage in which the named entities are colored and represented as a checkbox list. After reading the passage, the crowd-workers were given a cloze-style query with a missing entity. The annotation task is framed as to (1) validate coherence between the passage and the query, (2) report if the answer is not obvious or ambiguous, (3) select all the answer referents from a set of candidates, and (4) report any inconsistency and errors in the assignment, e.g. an incomplete entity markup, misspellings, etc. The annotation results were aggregated over the majority vote. Moreover, we manually validated each resulted sample and corrected all the reported drawbacks. The size of the dev and test sets is 7,577 and 7,257 samples, respectively.

[4] https://pypi.org/project/rusenttokenize/
[5] http://dict.ruslang.ru/freq.php
[6] http://docs.deeppavlov.ai/en/master/features/models/squad.html
[7] https://lenta.ru/
[8] https://toloka.yandex.com/tasks

### 2.4 Comparison to ReCoRD

We applied rusenttokenize to split passages into sentences and spaCy Russian Tokenizer[9] to build vocabularies over passages and queries. We used spaCy library[10] to compute statistics for ReCoRD (version available as a part of SuperGLUE benchmark tasks[11]). RuCoS counts 627,872 sentences and $1.2 \cdot 10^7$ tokens. Table 1 summarizes statistics of the datasets which is mainly based on the ReCoRD paper. ReCoRD is balanced by the news source in the ratio of 44% (CNN News) to 56% (Daily Mail News), while RuCoS samples are proportioned as of 67% (Lenta) to 33% (Deutsche Welle). In contrast to ReCoRD, RuCoS is designed so that the samples contain unique passages and queries, i.e. there are no multiple queries to a single passage. We additionally undersampled RuCoS over top-10 entities and answers to alleviate a potential shift in the frequency distribution.

| | **ReCoRD** | | | | **RuCoS** | | | |
|---|---|---|---|---|---|---|---|---|
| | Train | Dev | Test | Overall | Train | Dev | Test | Overall |
| number of samples | 65,709 | 7,481 | 7,484 | 80,674 | 72,193 | 7,577 | 7,257 | 87,027 |
| queries | 100,730 | 10,000 | 10,000 | 120,730 | 72,193 | 7,577 | 7,257 | 87,027 |
| unique queries | 99,713 | 9,977 | 9,968 | 80,179 | 72,193 | 7,577 | 7,257 | 87,027 |
| unique passages | 65,258 | 7,133 | 7,279 | 79,670 | 72,193 | 7,577 | 7,257 | 87,027 |
| query vocab | 119,069 | 30,844 | 31,028 | 134,397 | 109,899 | 30,203 | 27,813 | 120,410 |
| passage vocab | 352,491 | 93,171 | 94,386 | 395,356 | 279,333 | 90,699 | 83,237 | 303,647 |
| tokens / query | 21.3 | 22.1 | 22.2 | 21.4 | 22.2 | 22.1 | 21.6 | 22.2 |
| tokens / passage | 169.5 | 168.6 | 168.1 | 169.3 | 146.6 | 146.2 | 142.5 | 146.2 |
| entities / passage | 17.2 | 17.3 | 17.2 | 17.2 | 12.7 | 14.3 | 13.3 | 12.9 |
| answers / passage | 2.6 | 2.9 | - | 2.6 | 2.7 | 3.2 | - | 3.0 |
| entity frequency | 7.1 | 4.4 | 4.3 | 7.5 | 8.9 | 5.0 | 5.3 | 9.6 |
| answer frequency | 6.8 | 4.7 | - | 6.5 | 10.2 | 4.1 | - | 10.2 |
| IPM query | - | - | - | - | 0.86 | 0.85 | 0.86 | 0.86 |
| IPM passage | - | - | - | - | 0.82 | 0.81 | 0.82 | 0.82 |

Table 1: Comparative statistics of ReCoRD and RuCoS datasets.

ReCoRD tends to be more diverse regarding the entity and answer vocabularies. We assume that this may be caused by language peculiarities, specifics of the data sources, and the topic distribution. Besides, RuCoS comprises only single passage-query-answer triples, i.e. there are no multiple queries to one passage. We believe that this setting allows for better coherence and cohesion between the passage and the query. Specifically, ReCoRD query may be potentially generated from the last paragraphs of a news article that may describe other consequences or details of the news event. Notably, each RuCoS sample was validated on the cohesion as opposed to ReCoRD.

Note a few language peculiarities of the datasets. First, possessive adjectives (e.g. "English", "American", "British") are very common in news articles; these are extracted as named entities in English as opposed to Russian. Second, the answer entity in RuCoS may be expressed by a set of referents and surface forms. For example, "The President of Russian Federation", "Vladimir Vladimirovich", "V. Putin" and "Vladimira Vladimirovicha Putina" refer to the same entity. Besides, the answer entities in RuCoS may not be concorded in the query context. Therefore, the model is required to employ understanding of rich inflectional morphology and high lexical variability of Russian.

[9] https://github.com/aatimofeev/spacy_russian_tokenizer
[10] https://github.com/explosion/spaCy
[11] https://super.gluebenchmark.com/tasks

## 3 MuSeRC

### 3.1 Task Description

**Russian Multi-Sentence Reading Comprehension (MuSeRC)** is a reading comprehension dataset that requires to reason over multiple sentences to obtain the answer. MuSeRC is the first to study multi-hop MRC for Russian over an open-ended set of question types that require not only enhanced natural language understanding, but also interpretive reasoning. MuSeRC is designed following three main principles outlined in (Khashabi et al., 2018):

- Any question should be multi-hop, i.e. answered by inferring information spread across multiple sentences in a passage;
- The answer is not necessarily a text span and can not be easily extracted. It thus requires reasoning skills and deep text understanding;
- There can be a varying number of possible answers to the question which are independent of one another. The task is therefore not only to find the best answer candidate but to evaluate the relevance of each answer candidate.

We refer to multi-hop question as follows. *Multi-hop question* is a question that requires reasoning over information spread across several sentences in a passage. Besides, the model is supposed to perform interpretive reasoning and infer from general language understanding. Consider the example of the following passage *"(1) Mother bought apples. (2) They were on the table. (3) John has never eaten apples, that's why he couldn't stand it and tried one."* and the question *"Where were fruits that were eaten by a boy?"*. The question is multi-hop since the answer can be obtained with only information aggregated from more than one sentence. Moreover, the model is to employ coreference resolution and general language understanding.

MuSeRC task is framed as a binary classification over a set of the answer candidates. Specifically, the model is supposed to read a text and identify if a candidate is an answer to a given question. Each sample consists of a passage with enumerated sentences, a natural language question, and a set of possible answers. There can be multiple correct answers to the question. Such setting tests the model's ability to decide on the relevance of each candidate answer independently of others. MuSeRC is designed so that the answer may only be received by gathering information from multiple sentences.

(1) The missing daughter of a top Manager of "LUKOIL" Victoria Teslyuk was found dead in the Moscow region. (2) This was reported on may 3 by RIA New with reference to a source in law enforcement agencies. (3) "We have found the girl by accident, on the side of the road near Taldom, when the snow melted", - said the Agency interlocutor. (4) The investigative Committee of the Russian Federation confirmed to the Interfax news agency that the body of a young woman was found. (5) However, the UK stated that her identity has not yet been officially established. (6) A genetic examination will be required to identify the body. (7) At the same time, an unofficial source of Interfax stated that Teslyuk's valuables and personal belongings were found with the girl. (8) The cause of death, according to preliminary data, was a skull fracture. (9) The 16-year-old daughter of the top Manager of LUKOIL, Robert Teslyuk, Victoria Teslyuk, disappeared on March 26. (10) She left her house in the village of Gribki and went to a math tutor in Moscow. (11) However, the girl did not reach the tutor. (12) At first her phone was not answered, and then it was out of the access zone. (13) In April, the media reported that Teslyuk's dismembered body was allegedly found in Arkhangelsk. (14) However, this information was later refused.

**Question**
What does RIA New report on may 3?

**Answers**
- Victoria Teslyuk was found dead (correct)
- The snow melted
- Daughter of a top Manager was found dead. (correct)
- The tutor was out of the access zone

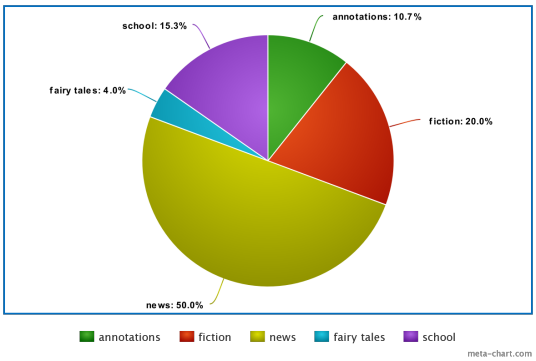


Figure 2: An example of MuSeRC sample and the data sources distribution.

### 3.2 Data Collection

MuSeRC samples have diverse provenance collected over 5 different domains, and hence are expected to be more diverse in their contents as compared to single-domain MRC datasets. Our dataset contains more than 900 paragraphs across 5 different domains, namely: (1) elementary school texts, (2) news, (3) fiction stories, (4) fairy tales, and (5) summaries of TV series and books. The distribution of the

data sources is presented in Figure 2. Notably, school stories and fairy tales are relatively simple for understanding, while news and summaries tend to be more complicated.

We used a variety of publicly available resources to construct MuSeRC such as the news segment of Taiga corpus (Shavrina and Shapovalova, 2017), elementary school texts from Russian state exam tests, and materials[12], etc. We filtered samples which correspond to the following criteria: (1) the passage length is of less than 1.5K characters, (2) the passage must contain named entities, and (3) if the passage contains only one named entity, then the entity must have one or more coreference relations. Besides, we manually validated each sample on coherence and cohesion. Each passage was segmented into sentences via rusenttokenize. The resulted sentences were manually validated on the correctness of segmentation.

### 3.3 Annotation Procedure

We now describe a two-step annotation procedure for obtaining natural language questions and answer candidates, and their further validation. We used Yandex.Toloka to conduct the annotation procedure. The first step is to collect natural language questions and their corresponding answers for each passage obtained in 3.2. The crowd-workers were required to: (1) pose a natural language question to a given passage, (2) specify sentence numbers needed to obtain the answer, and (3) provide a set of both correct and incorrect answers. We filtered out samples that require only one sentence to get the answer. To ensure the high quality at this step, we manually validated each submitted annotation assignment. Besides, we prepared a training pool for the crowd-workers to practice. A detailed instruction was available at any time. First, we analyzed the assignments to check if the workers understand the task correctly. Unfortunately, 70% of the questions were not relevant. Most of the workers cheated or posed only single-hop questions, i.e. the questions that can be answered based on a single sentence from a passage. This step helped us to re-design the annotation procedure so that to obtain the required data.

Hence, we incorporated the following changes for the second step as to (1) cut the training assignments since this proved to confuse the crowd-workers, (2) provide more information on multi-hop questions accompanied with examples in the instruction, (3) ask the workers to provide a fixed number of both correct and incorrect answers, and (4) write a filtering script used to discard irrelevant samples based on the assignment analysis.

In the second step, we automatically filtered out all the assignments that: (1) involve potential cheating, (2) contain single-hop questions, and (3) include inappropriate answers. The crowd-workers were asked to validate the results obtained in the first iteration, specifically to check if the question can be answered using the given passage and if the answer requires the information over multiple sentences. Besides, all the assignments were then manually validated. We present the examples of the web interface for the annotation procedure in Appendix A.2.

### 3.4 Comparison to MultiRC

MuSeRC consists of 12,805 sentences and $2.53 \cdot 10^5$ tokens computed with rusenttokenize and spaCy Russian Tokenizer. Table 2 represents the comparative statistics of the MultiRC and MuSeRC datasets. MultiRC dataset contains more data than MuSeRC: questions and answers. It should be noted that MultiRC includes nearly 2K single-hop questions. In contrast, MuSeRC is designed so that it contains only multi-hop questions. However, the number of MuSeRC multi-hop questions is lower than that of the MultiRC, as such questions are very time and source consuming.

Figure 3 outlines the distribution of the most frequent questions in MultiRC and MuSeRC tasks (MultiRC is on the left; MuSeRC is on the right). It's not correct to compare question types directly but some consistent patterns could be identified. Though the wh-questions are very common for both datasets, MuSeRC exhibits a wide variety of the interrogative expressions due to specifics of Russian. This also indicates a broader diversity of the question types. Notably, the variety of MuSeRC questions is higher as compared to MultiRC, where, for example, almost 35% of questions start with the interrogative pronoun "what", while in Russian it constitutes about 15%. About 28% of MultiRC questions require binary decisions (true/false or yes/no), while in MuSeRC it's only 1%.

[12]https://fipi.ru/

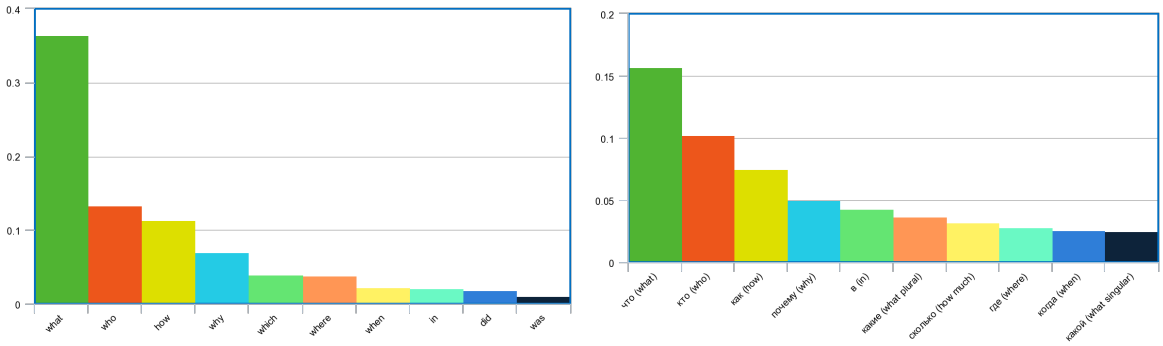


Figure 3: Most frequent first chunks of the questions

| | **MultiRC** | **MuSeRC** |
|---|---|---|
| number of paragraphs (train) | 456 | 500 |
| number of paragraphs (dev) | 83 | 100 |
| number of paragraphs (test) | 166 | 322 |
| number of questions (train) | 5,130 | 2,896 |
| number of questions (dev) | 952 | 528 |
| number of questions (test) | 1,819 | 1,812 |
| number of answers (train) | 27,242 | 11,950 |
| number of answers (dev) | 4,845 | 2,235 |
| number of answers (test) | 9,691 | 7,613 |
| number of multi-hop questions | 5,825 | 5,228 |
| candidates / question | 5.28 | 4.16 |
| answers / question | 2.31 | 1.86 |
| sentence / passage | 14.3 | 13.875 |
| tokens / passage | 258.9 | 203.9 |
| tokens / question | 10.9 | 7.61 |
| tokens / answer | 4.7 | 5.3 |
| yes/no/true/false questions (%) | 27.57% | 1% |

Table 2: Comparative statistics of MultiRC and MuSeRC datasets.

## 4 Experiments

In this section, we describe a two-step baseline approach and human performance on each task. We first used TF-IDF method as a naive baseline (see Section 4.1.1) and three BERT-based models as advanced baselines (see Section 4.1.2). We outline the design of the human benchmark in Section 4.2 and provide more details in the Appendices.

**Metrics** We roughly follow the evaluation procedure by (Zhang et al., 2018; Khashabi et al., 2018).

- **MuSeRC** Since each answer-option can be assessed independently, we apply **F1-averaged (F1a)** to evaluate binary decisions over all the answer options in the dataset. It is a harmonic mean of precision and recall per question. **Exact Match (EM)** is the exact match per each instance, i.e. each set of predictions should be the same as of the answers.

- **RuCoS EM** here measures the percentage of predictions that match any one of the answer options exactly. **Macro-averaged F1** measures the average overlap between the prediction and the answer referents by averaging the maximum F1 scores for each instance over a total number of instances.

### 4.1 Baselines

#### 4.1.1 Naive Baseline

TF-IDF vectorization method via Scikit-learn library (Pedregosa et al., 2011) is used as a naive baseline (`TF-IDF`). For RuCoS task, we replaced the cloze-style query with each candidate answer. We then computed the cosine similarity between TF-IDF vector representations of the passage and the generated query. The answer is the candidate of the maximum similarity value. TF-IDF solution for MuSeRC task is similar: we concatenated the passage with each answer option, and then computed the cosine similarity between TF-IDF vector representations of the resulted concatenations and the question. The answer of the maximum similarity value is considered as the prediction.

#### 4.1.2 Advanced Baselines

We fine-tuned multilingual BERT model (Devlin et al., 2019) and two monolingual ones by DeepPavlov[13] which are a part of HuggingFace library (Wolf et al., 2019).

- **Multilingual BERT** (`MultiBERT`) is a multilingual language model pre-trained over concatenated monolingual Wikipedia corpora in 104 languages including Russian. We fine-tuned this model on each English and Russian MRC tasks to compare the performance.

- **RuBERT** (`RuBERT`) is a monolingual BERT-based model that was trained on the Russian segment of Wikipedia and Russian news data. Notably, `RuBERT` outperforms `MultiBERT` over a number of NLP tasks for Russian.

- **Conversational RuBERT** (`RuBERT-Conv`) was trained by DeepPavlov on a number of publicly available sources that reflect Russian relatively non-formal discourse, including OpenSubtitles (Lison and Tiedemann, 2016), Social Media segment of Taiga corpus, and many others.

### 4.2 Human Benchmark

We designed two human benchmark tasks using Yandex.Toloka to evaluate human performance. We provide examples of web interface for the tasks in the Appendices, particularly RuCoS A.1 and MuSeRC A.2. Besides, the results of the human benchmark tasks and more detailed information are publicly available[14]. The crowd-workers were required to successfully complete a training pool of the corresponding human benchmark task assignments. The expandable instruction was available at any time.

**RuCoS Human Benchmark Task** was framed as to (1) read the passage and the cloze-style query with a placeholder, (2) select all the referents to the answer entity that best fits the placeholder, and (3) report any errors, including inconsistency, incoherence, and ambiguous answers.

**MuSeRC Human Benchmark Task** required crowd-workers to (1) read the passage and the question, (2) check if the answer could be obtained using the passage (3) select the number of sentences needed to infer the answer, and (4) select one or more possible answers from a set of candidates.

Requirements to the crowd-workers are similar to those described in Section 3.3. We did not consider the results of the crowd-workers whose quality performance on the control tasks was lower than 50%. The dynamic overlap of 5 annotators allowed for the high quality of the inter-annotator agreement (Cohen's kappa between each pair of annotators ranges between 0.31 and 0.78. Mean average across each pair of annotators is 0.55 for MuSeRC and 0.48 for RuCoS). The platform allows to analyze the submissions, their consistency, the level of performers' skills, and may automatically increase the overlap within the range to ensure the best quality. Additionally, we manually validated all the samples that contained any feedback from the crowd-workers. The results were aggregated using the majority vote over each sample. We used metrics described in Section 4 to assess the performance of human solvers.

[13] `http://docs.deeppavlov.ai/en/master/features/models/bert.html`

[14] `https://github.com/RussianNLP/RussianSuperGLUE/tree/master/HumanBenchmark`

## 5 Results

The results of the baseline models and human solvers are presented in Table 3. We did not evaluate the performance of TF-IDF and BERT-based baselines on the English datasets, as it's not appropriate to compare them directly with Russian-oriented baselines. As for the English human benchmark tasks, we used the scores from (Zhang et al., 2018; Khashabi et al., 2018). We now give a brief description of the performance results. Compared to MuSeRC task, MultiRC human solvers perform slightly better showing the difference of 1.2 F1a score and 9.9 EM score. Meanwhile, RuCoS human solvers achieve better results as opposed to ReCoRD task. Despite that, human solvers obtained prominent performance for each MRC task. TF-IDF baseline shows the worst results. The best MuSeRC performance among the BERT-based models is achieved by `RuBERT`. Notably, `RuBERT` demonstrates the best results for RuCoS task. Still, there is a substantial difference between the human and the baseline results.

It is worth mentioning that recent state-of-the-art language models for English, specifically T5 model (Raffel et al., 2019), have outperformed the human results on both ReCoRD and MultiRC datasets. The model achieved 94.1% of F1 score and 93.4% of EM score for ReCoRD. The following results are obtained for MultiRC: 88.1% of F1a score and 63.3% of EM score. T5 demonstrates impressive results, which we hope can be achieved for Russian as well. In future work, we are going to explore the language patterns in questions of different reasoning types for both English and Russian MRC datasets. This may be useful when studying the linguistic properties of the language models, specifically multilingual ones. Another line of research is to analyze top-k best leader-board models in a cross-lingual scenario. Particularly, we suppose that the training objectives of language models such as text infilling or sentence shuffling as in T5 and BART (Lewis et al., 2019a) may play a big role for the outstanding performance.

| Model | MultiRC | MuSeRC | ReCoRD | RuCoS |
|---|---|---|---|---|
| Human Benchmark | **81.8/51.9** | **80.6/42.0** | **91.7/91.3** | **93.0/92.4** |
| `MultiBERT` | 54.8/12.0 | 66.8/33.6 | 39.7/38.9 | 30.6/29.6 |
| `RuBERT-Conv` | - | 71.7/32.9 | - | 26.4/25.9 |
| `RuBERT` | - | 71.7/33.6 | - | 34.4/33.9 |
| `TF-IDF` | - | 58.9/24.4 | - | 25.6/25.1 |

Table 3: Comparative results of the naive & advanced baselines, and human solvers for RuCoS and MuSeRC.

## 6 Conclusion

This work is devoted to the assembly of RuCoS and MuSeRC, two novel machine reading comprehension datasets for Russian. The datasets are publicly available[15] and included in the evaluation suite of RussianSuperGLUE, the Russian general language understanding benchmark. The tasks require reasoning over multiple sentences, commonsense knowledge, and advanced natural language understanding. We hope to provide a detailed description of the construction procedure, as well as a comparative analysis of the proposed datasets, and their analogous tasks for English. Due to the language specifics, RuCoS and MuSeRC tend to be relatively more complicated in some aspects as opposed to ReCoRD and MultiRC. Our baselines, including recent state-of-the-art BERT-based models for Russian, can not compete with human solvers falling beyond their performance. We hope that RuCoS and MuSeRC will spur more research in the field of Russian reading comprehension, and prompt the study of multi-hop reasoning in a cross-lingual scenario.

## Acknowledgements

We thank our reviewers for their insightful comments. We wish to acknowledge the help with the annotation procedures provided by Andrey Evlampiev (Sberbank, Moscow, Russia). We also thank our team for their support and useful discussions.

[15] https://github.com/RussianNLP/RussianSuperGLUE

## References


Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive reading comprehension by runtime machine translation. *arXiv preprint arXiv:1809.03275*.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yurii Kuratov, Denis Kuznetsov, et al. 2018. Deeppavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *arXiv preprint arXiv:2003.05002*.

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.

Pavel Efimov, Leonid Boytsov, and Pavel Braslavski. 2019. Sberquad–russian reading comprehension dataset: Description and analysis. *arXiv preprint arXiv:1912.09723*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *NAACL*.

Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 320–332. Springer.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019b. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Pengyuan Liu, Yuning Deng, Chenghao Zhu, and Han Hu. 2019. Xcmrc: Evaluating cross-lingual machine reading comprehension. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 552–564. Springer.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning:"taiga" syntax tree corpus and parser. *–2017*, page 78.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

W Wang, N Yang, F Wei, B Chang, and M Zhou. 2017. R-net: Machine reading comprehension with self-matching networks. *Natural Lang. Comput. Group, Microsoft Res. Asia, Beijing, China, Tech. Rep*, 5.

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. In *Advances in Neural Information Processing Systems*, pages 3266–3280.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

# A Appendix

We provide examples of Yandex.Toloka tasks for MuSeRC and RuCoS annotation procedures, as well as the web interfaces of the human benchmark tasks.

## A.1 RuCoS

Figure 4 shows an original sample from RuCoS dataset which was automatically translated and illustrated in Section 2.1.

**Passage:** Мать двух мальчиков, брошенных отцом в московском аэропорту *Шереметьево*, забрала их. Об этом сообщили *ТАСС* в пресс-службе министерства образования и науки *Хабаровского края*. Сейчас младший ребенок посещает детский сад, а старший ходит в школу. В учебных заведе- ниях с ними по необходимости работают штатные психологи. Также министерство социальной защиты населения рассматривает вопрос о бесплатном оздоровлении детей в летнее время. Через несколько дней после того, как *Виктор Гаврилов* бросил своих детей в аэропорту, он явился с повинной к следователям в городе *Батайске* *Ростовской области*.

**Query:** 26 января <placeholder> бросил сыновей в возрасте пяти и семи лет в Шереметьево.

**Correct Entities:** Виктор Гаврилов

Figure 4: An original sample from RuCoS dataset provided in Section 2.1

In RuCoS annotation tasks, the crowd-workers are shown a passage, a cloze-style query, and a set of the answer candidates organized as a checkbox list. The named entities are colored in dark green. The workers were encouraged to report any inconsistency and errors, or give any other feedback. We manually checked each assignment and corrected the reported errors.

Figure 5 illustrates an example of the web interface for RuCoS annotation procedure. The crowd-workers were asked to validate the coherence between the passage and the cloze-style query, select all the answer referents, and report any inconsistency and errors.

Figure 6 shows a sample of the web interface for RuCoS human benchmark task. The task is similar to that of the annotation procedure. The crowd-workers were required to select all the answer referents that best fit the query placeholder, and encouraged to give any feedback.

## A.2 MuSeRC

Figure 7 shows an original sample from MuSeRC dataset which was automatically translated and provided in Section 3.1.

Figure 8 demonstrates a case of the interface for the first step of MuSeRC annotation procedure. The crowd-workers were required to pose natural language questions to a given passage, select sentences needed to infer the answer and provide a set of both correct and incorrect answers to the posed question.

Figure 9 outlines an example of the web interface for the second step of MuSeRC annotation procedure. The crowd-workers were asked to check if a given question can be answered based on the passage (obtained from the first step). Besides, this step allows for additional validation of the question type, specifically one-hop or multi-hop. The workers were also asked to pose a new natural language question to the given passage and also provide a set of both correct and incorrect answers.

Finally, Figure 10 illustrates an example of a web interface for the MuSeRC human benchmark task. The crowd-workers were given a passage, a natural language question, and a set of the answer candidates.

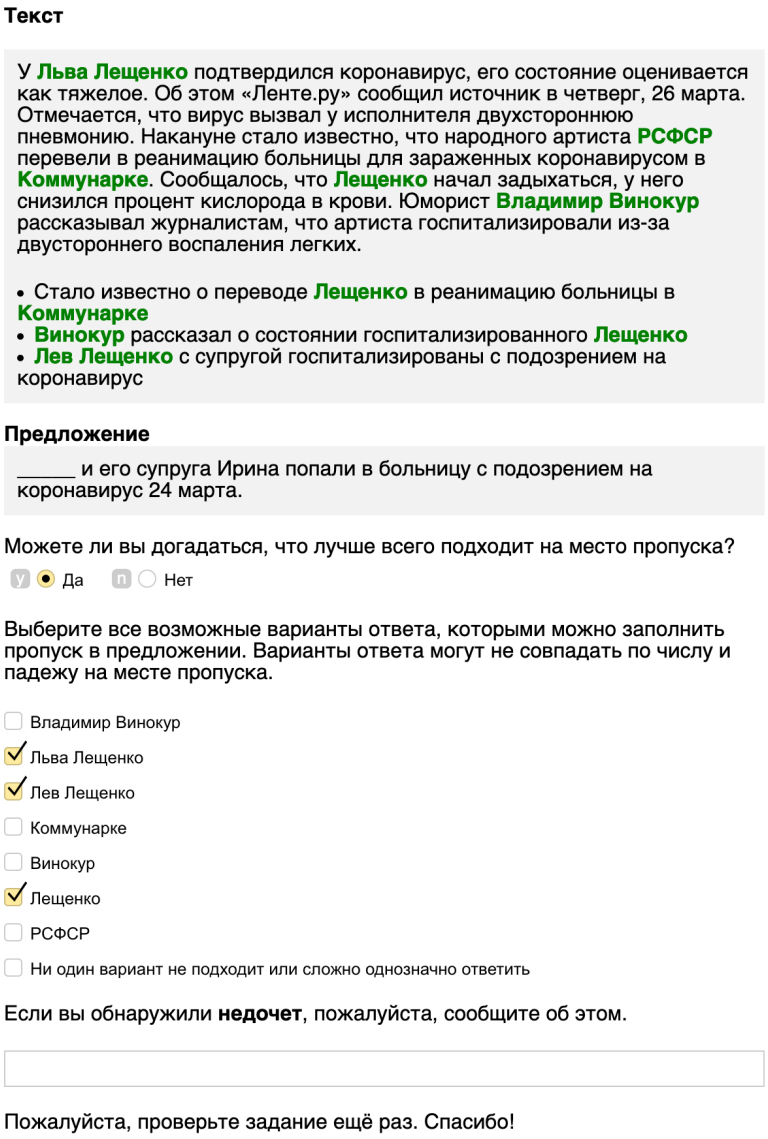


Figure 5: The web interface for RuCoS annotation procedure.

The task was to (1) check if the answer can be obtained using the passage, (2) select whether one or more sentences are required to infer the answer, and (3) select one or more possible answers to the question.

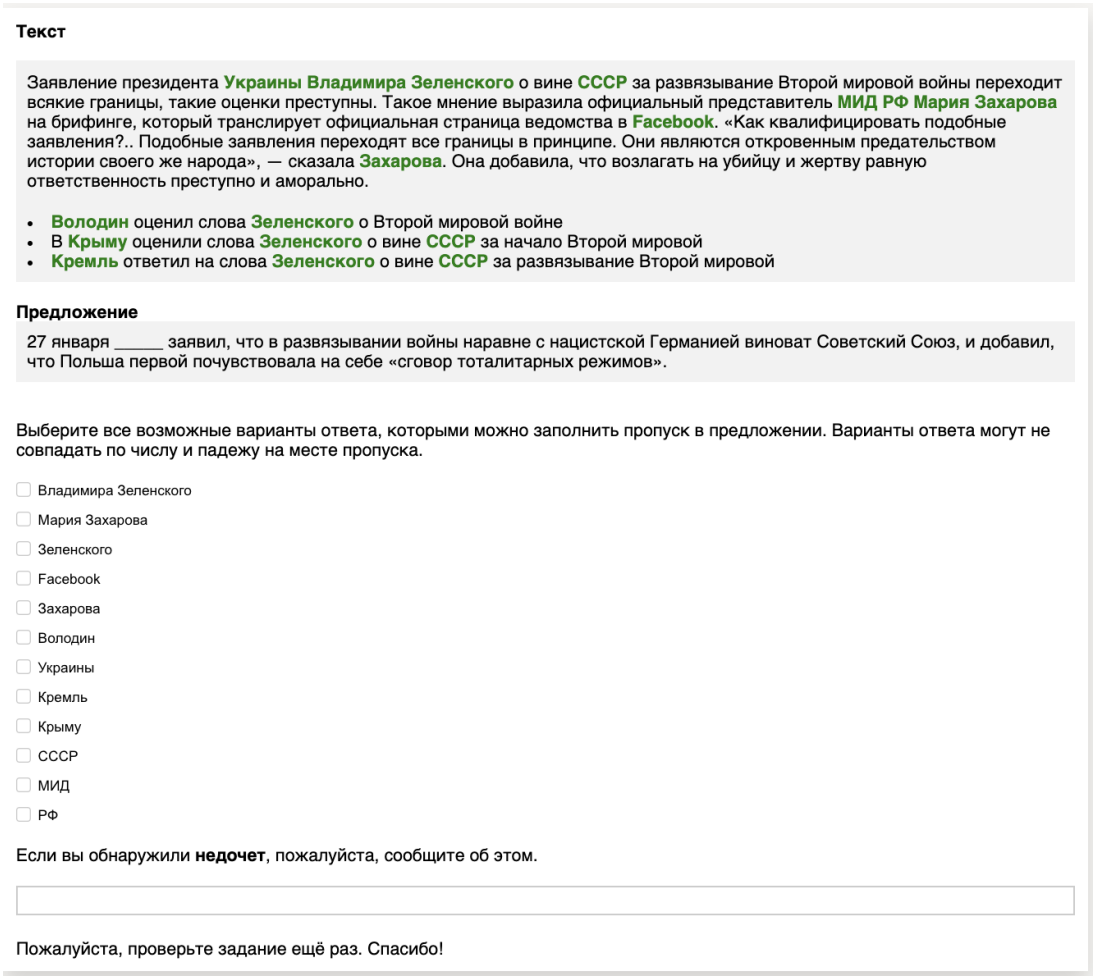


Figure 6: The web interface for RuCoS human benchmark task.

**Passage**:

(1) Пропавшая дочь топ-менеджера "Лукойла" Виктория Теслюк найдена мёртвой в Подмосковье. (2) Об этом 3 мая сообщает РИА Новости со ссылкой на источник в правоохранительных органах. (3) "Обнаружили девушку случайно, на обочине дороги недалеко от Талдома, когда растаял снег", - заявил собеседник агентства. (4) В Следственном комитете РФ агентству "Интерфакс" подтвердили, что найдено тело молодой женщины. (5) Однако в СК заявили, что официально её личность пока не установлена. (6) Для идентификации тела потребуется генетическая экспертиза. (7) При этом неофициальный источник "Интерфакса" заявил, что при девушке нашли ценности и личные вещи Теслюк. (8) Причиной смерти, по предварительным данным, стал перелом черепа. (9) 16-летняя дочь топ-менеджера "Лукойла" Роберта Теслюк, Виктория Теслюк, пропала 26 марта. (10) Она вышла из дома в деревне Грибки и направилась к репетитору по математике в Москву. (11) Однако до репетитора девушка не доехала. (12) Сначала её телефон не отвечал, а потом оказался вне зоны доступа. (13) В апреле в СМИ появились слухи о том, что расчленённое тело Теслюк якобы нашли в Архангельске. (14) Однако затем эта информация была опровергнута.

**Question**:
О чем сообщает РИА Новости 3 мая?

**Answers**:

- Виктория Теслюк найдена мертвой. (correct)
- Найдено тело дочери топ-менеджера. (correct)
- Растаял снег.
- Репетитор оказался вне зоны доступа.

Figure 7: An original sample from MuSeRC dataset provided in Section 3.1

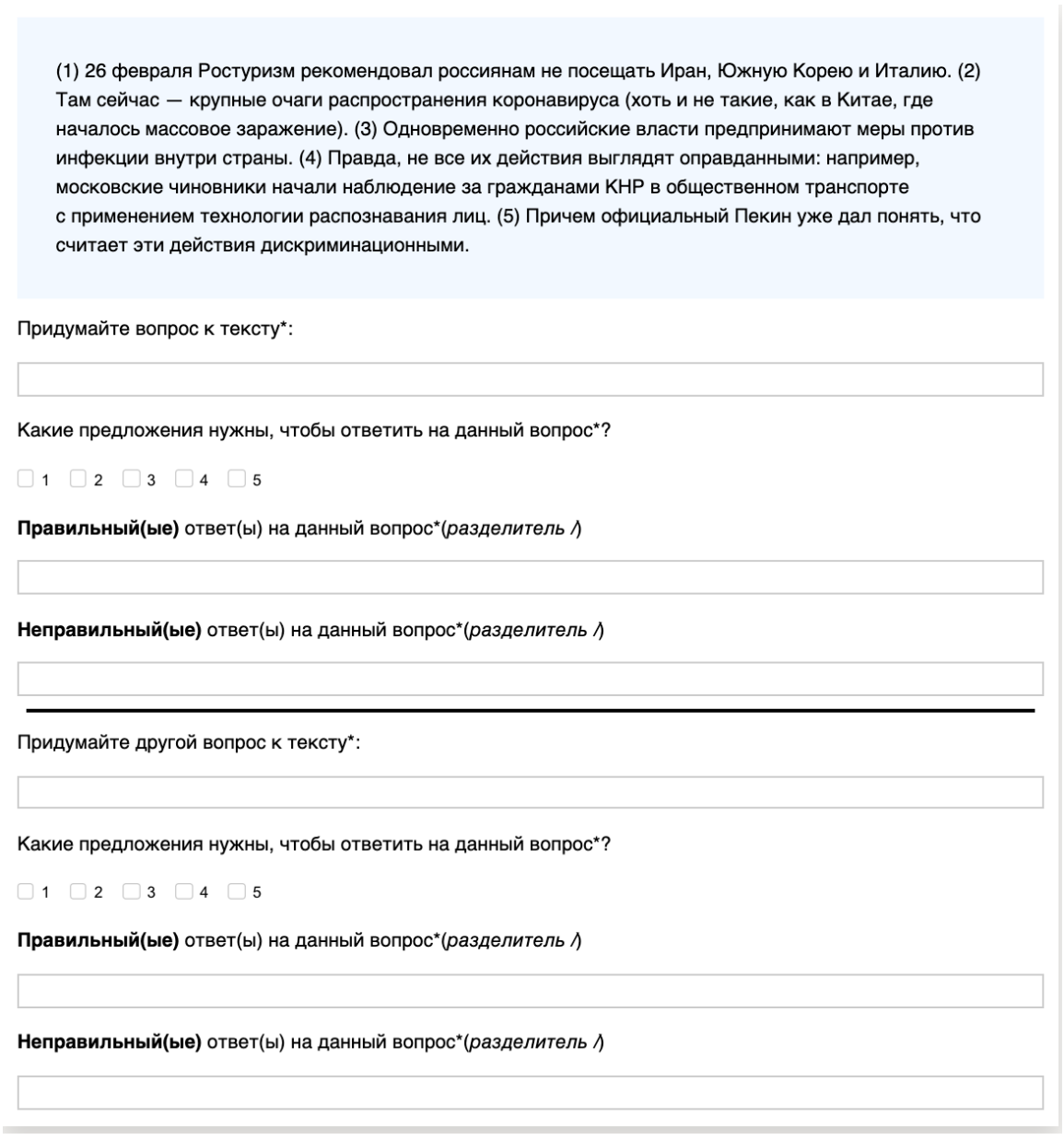


Figure 8: Web interface for the first step of MuSeRC annotation procedure.

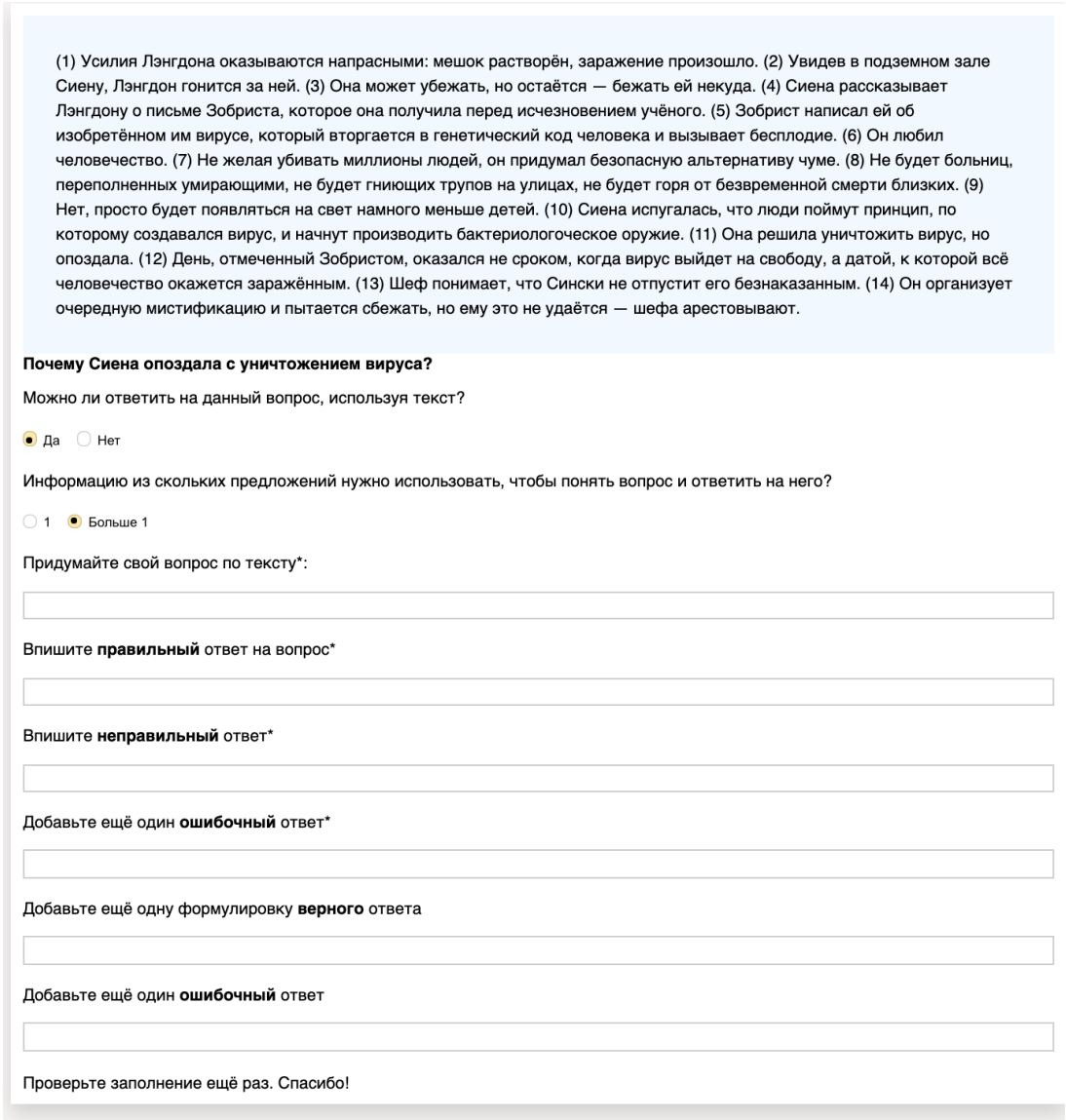


Figure 9: The web interface for the second step of MuSeRC annotation procedure.

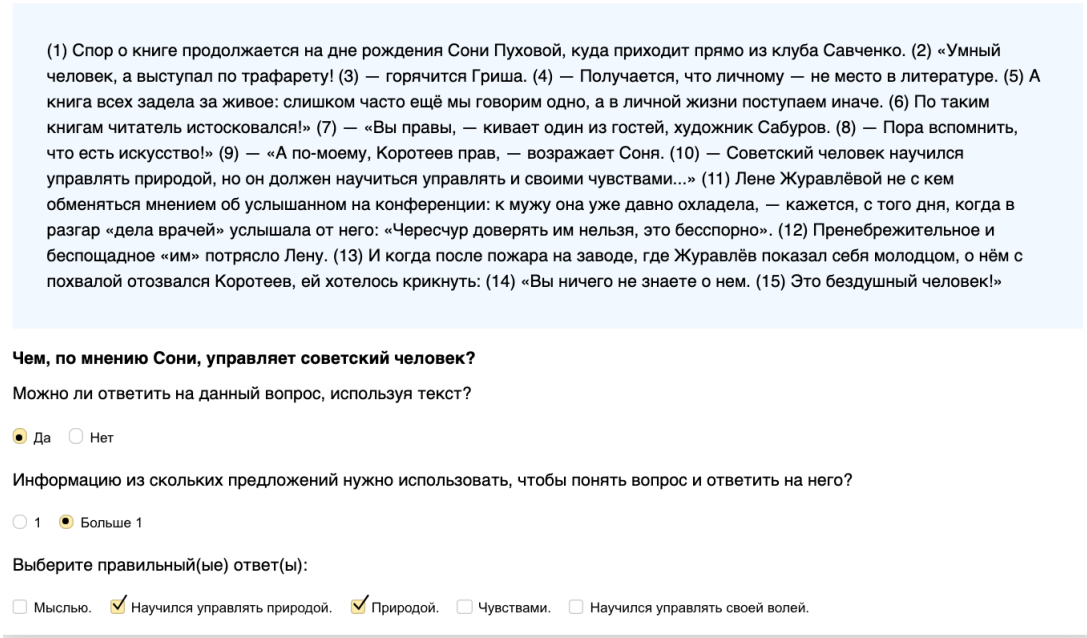


Figure 10: Example of the assignment for the MuSeRC human benchmark task.