Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2023"

June 14–16, 2023

Fact-checking benchmark for the Russian Large Language Models

Anastasia Kozlova
SberDevices
anastasi2510@gmail.com

Denis Shevelev
SberDevices
reddraner@gmail.com

Alena Fenogenova
SberDevices
alenusch@gmail.com

Abstract

Modern text-generative language models are rapidly developing. They produce text of high quality and are used in many real-world applications. However, they still have several limitations, for instance, the length of the context, degeneration processes, lack of logical structure, and facts consistency. In this work, we focus on the fact-checking problem applied to the output of the generative models on classical downstream tasks, such as paraphrasing, summarization, text style transfer, etc. We define the task of internal fact-checking, set the criteria for factual consistency, and present the novel dataset for this task for the Russian language. The benchmark for internal fact-checking and several baselines are also provided. We research data augmentation approaches to extend the training set and compare classification methods on different augmented data sets.

Keywords: fact-checking, factual consistency, lm, nlg, text generation
DOI: 10.28995/2075-7182-2023-22-267-277

Факт-чекинг для улучшения языковых моделей на русском языке

Анастасия Козлова
SberDevices
anastasi2510@gmail.com

Денис Шевелев
SberDevices
reddraner@gmail.com

Алена Феногенова
SberDevices
alenush93@gmail.com

Аннотация

Генеративные языковые модели сейчас стремительно развиваются и используются повсеместно. Однако, у них всё ещё есть ряд лимитов, и упущений, таких как ширина контекста, склонности к галлюцинациям и дегенерациям, логические связи, и изменения фактической информации. В данной работе мы рассматриваем задачу проверки фактов для непосредственно выхода генеративных моделей в классических генеративных задачах, таких как: парафраз, суммаризация, перенос стиля и подобных. В данной работе мы определяем задачу и критерии внутреннего факт-чекинга, впервые представляем новый русскоязычный датасет для этой задачи, а также набор тестов для оценки моделей и их сравнения с базовыми решениями. Мы также рассмотрели несколько методов аугментации данных для тренировочного сета и провели сравнительный анализ методов на разных наборах данных.

Ключевые слова: факт-чекинг, консистентность фактов, большие языковые модели, автоматическая генерация текста

1 Introduction

Large language models are fast developing and excel at producing text. The interest in language models continues to grow as such models are used to solve various downstream tasks, such as paraphrasing, summarization, style transfer, etc. Plenty of these tasks can be defined as generating the text based on some source text, where the model generates new original text, preserving the same sense. For such generative models, one of the main requirements for generated texts is factual correctness and consistency of text with the source data.

Despite progress in the quality of language models and the growth of scientific research in this field, texts generated using language models may contain inaccuracies, hallucinations (Zhou et al.,

2020)(Bender et al., 2021), and misinformations (Kryściński et al., 2019a). Automatic fact-checking can serve as an effective means of identifying inconsistencies in generated text, thereby enhancing the quality and reliability of the output. The significance of factual accuracy cannot be overstated, particularly in the context of news and medical articles, legal documents, and other socially consequential texts. At the same time, an automatic fact-checker can provide a more time-efficient solution to the problem of inaccurate information than manual fact-checking, making it available to a broader group of people. Thus, automatic fact-checking plays a vital role in improving the accuracy and consistency of information, helping to overcome the problem of false or misleading information.

Existing approaches to fact-checking are based on consistency testing of statements against evidence (Thorne et al., 2018a)(Mesgar et al., 2020) but do not consider the original information's completeness. For generative downstream tasks, preserving the consistency and completeness of the data is essential. Thus, the fact-checking systems may also be used as an essential tool for the evaluation of the large language models (Tam et al., 2022), (Chaudhury et al., 2022).

This work focuses on the internal fact-checking task as a fact-preservation problem and defines its criteria. In this paper, we present a new dataset and the factual verification benchmark[1] for the Russian language. The dataset contains tagged examples labeled *consistent* and *inconsistent*; for inconsistent examples, ranges containing violations of facts in the source text and statements are also presented. Various sources were used for data collection, such as texts obtained by the paraphrasing task and summarization data, translations from English to Russian of existing datasets for fact-checking, and text argumentation. We use the obtained dataset to fine-tune and evaluate models, such as ruBERT, ruRoBERTa, and ruGPT3, for the fact-checking task.

The rest of the paper is structured as follows. First, we overview the papers that are related to the field of fact-checking. In section 3, we discuss how we define the internal fact-checking task and what the fact is. Section 4 is devoted to the data we use in our experiments and various approaches to its collection. The methods and models we used and the description of the experiments are presented in Section 5. Finally, section 6 presents the evaluation and discussion.

2 Related work

The general task of fact-checking can be divided into several sequential steps (Guo et al., 2022) — first, the search of the sources and the collection of evidence necessary for verification verdict. Secondly, selecting the most relevant evidence to be used for verification. And finally, issuing a verdict using the collected evidence.

Thus, fact-checking can be separated into internal and external depending on the evidence source type. External fact-checking is the process of checking the actual accuracy of the content generated by a language model using external sources of information and data. This approach aims to determine the consistency of the generated text by comparing it with verifiable facts from some databases or sources such as news articles, academic journals, government reports, and other reliable sources. For internal fact-checking, a reliable source of evidence is predetermined by the downstream task. For example, we are checking the actual consistency of the source text with the content generated by the summarization model. The factual consistency of the summarization task is one of the most frequent cases, discussed in works (Wang et al., 2020) (Fabbri et al., 2021) (Kryściński et al., 2019b). In this case, the model's input text is evidence and aims to preserve the facts in the generated text output. This paper will focus on internal fact-checking for the text-generative downstream tasks and the factual consistency of language models.

2.1 Fact-checking Datasets

The bottleneck for building a fact-checking model is the need for labeled data for various languages. Most of the datasets are presented in English only. The FEVER dataset (Thorne et al., 2018a) is one of the most well-known fact-checking datasets in English, which contains claims extracted from Wikipedia documents. Each claim is assigned one of three labels: *Supported*, *Refuted* or *NotEnoughInfo*.

[1] `https://huggingface.co/datasets/akozlova/RuFacts`

For the first two classes, the annotators recorded the sentences forming the necessary evidence for their judgment. The evidence is texts from Wikipedia, and annotators write claims for verification. Another dataset for fact-checking is the Vitamin C dataset (Schuster et al., 2021) based on texts from Wikipedia. The largest publicly available multilingual dataset is the X-FACT dataset (Gupta and Srikumar, 2021), which includes 31,189 short statements labeled for factual correctness and covers 25 typologically diverse languages, including statements in Russian. As part of the FactRuEval (Starostin et al., 2016) competition, a publicly available corpus was created to evaluate fact extraction systems. The corpus can be used to detect facts of specific types in the texts but is not intended to be used for the fact-checking task. The Russian Commitment Bank dataset that is a part of the Russian SuperGLUE (Shavrina et al., 2020) benchmark can be considered a close variant of the task definition as it also validates the contradiction/entailment of some source premise. However, Natural Language Inference (NLI) is a much broader task and can not be defined as fact-preservation due to the inability of concrete fact selection.

2.2 Fact-checking Methods

There are various approaches to the problem of fact-checking using evidence. Question-answer systems are often used for fact-checking, the main task of which is to check the consistency of named entities in texts. According to previous research, scores based on question-answer systems correlate highly with a human judgment of facts. The approach (Wang et al., 2020) and similar question-answer approaches are based on the intuitive assumption that if we ask the same questions to both the summarized text and its source, we will get similar answers, but only if the generated text matches the source. The authors have shown that this approach significantly outperforms other automatic scoring measures in terms of correlation with human judgments of factual consistency. However, such approaches do not consider the completeness of the presentation of the original information, checking only individual facts.

The most common formulation of the fact-checking problem is to build a binary classifier based on a pre-trained language model, such as BERT, labeled *Supported* or *Unsupported* (Glockner et al., 2022; Guo et al., 2022). The paper is also based on the hypothesis from the FactCC [2] paper (Kryściński et al., 2019b) that errors made by paraphrasing models are most often associated with the use of incorrectly named entities, as well as numbers and pronouns. The authors base their work on the approach for generating training data for fact-checking to reduce manual markup costs. The training data is generated by applying a series of rule-based transformations to the sentences of the source documents. Examples are created by sampling individual sentences, later called claims, from source documents. The claims then undergo a series of text transformations resulting in new sentences with positive and negative labels. The advantage of using a synthetic dataset is that it generates large amounts of data at minimal costs.

The author of the paper (Lee et al., 2021) used a perplexity score from the language model to check the consistency between a claim and evidence. The researchers suggest including evidence in the perplexity calculation, using it as a prefix for a claim since perplexity measures the likelihood of a given sentence regarding a previously encountered text. They assume that unsupported claims have higher perplexity compared to supported claims.

Some approaches (Cao et al., 2020) are devoted to correcting factual errors in generated texts through post-editing. Usually, such text correction models are trained on adversarial examples built using heuristics to introduce errors. However, generating such examples using heuristics often needs to generalize better to actual model errors. In this paper, the authors propose to generate representative non-factual adversarial examples using infilling language models. The authors use a beam search of lower-ranked candidates from the language model to source potentially incorrect facts, creating a set of plausible and probable but incorrect synthetic texts for a particular correct text.

3 Task definition

The task of internal fact-checking can be considered from different perspectives. For example, based on the Named Entity Recognition (NER)/facts span detection in two texts or the classical task of NLI,

[2] https://github.com/salesforce/factCC

determining whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”.

We are to combine such approaches and formulate the fact-checking problem as follows: Given a pair of texts (c, g), where c is a source human-written text, and g is the generated text by some generative model, that needs to be checked for factual consistency with the conditional input c. The fact-checking model must predict one of two labels for the generated output: the facts are ‘consistent‘ or ‘inconsistent‘.

Based on the problem statement, the requirements for a fact-checker include 1) examining the factual inconsistency, looking for the presence of facts that are not contained in the source text, and 2) verifying the completeness of the presentation of the source information. It’s worth mentioning, for instance, not all facts should be presented for the summarization task in the generated abstract, but at the same time, corruption or new facts, in this case, are unacceptable.

A similar definition is used in works that proposed an assessment of factual consistency evaluation methods (Honovich et al., 2022). They require the text to be faithful to its source text, regardless of the "correctness" concerning the "real world". To assess faithfulness, criteria are based on the information presented in the input text, not external knowledge.

Investigating the common errors of factual inconsistency in the corresponding works (Kryściński et al., 2019b) (Tam et al., 2022) we highlight the cases that cover the most frequently encountered contradictions in facts in generated texts. We further use them for the data augmentation procedure. The classes are the following:

- **NER (names, numbers, localizations)**. *Examples: “Lermontov” instead of “Pushkin”; “125.000 roubles” instead of “125 roubles”*
- **relations** *Examples: “grandmother” instead of “grandfather”; “chef” instead of “subaltern”*
- **negotiation** *Examples: “Natasha did not see her boss yesterday” and “Natasha saw her boss yesterday”*
- **gender** *Examples: “Natasha did not see her boss yesterday” and “Natasha did not see his boss yesterday”*
- **states (actions, positions)** *Examples: “Masha has eaten the apple” and “Masha is eating the apple”*

To sum it up, the fact-checking system needs to be based on these typical error cases, and the following conditions need to be complied with: 1) the facts are correct and not corrupted in both texts (source and generated); 2) any additional facts in the generated texts are not included; 3) the generated text includes all the main facts from the source text.

4 Data

4.1 Data Collection

Various data sources and approaches for data generation were used to create the training and test datasets for the fact-checking task. Our approach involves analyzing data at both the sentence level and within smaller texts. The data exhibits an average text length of 198 symbols, with a minimum length of 10 symbols and a maximum length of 3,402 symbols. The final dataset was formed using three main approaches: 1) texts generated by a paraphrase model 2) translations of datasets for fact-checking 3) text augmentation.

Text Generation. The most frequent usage of the fact-checking verification system is some generated output based on the original text. Thus, we take the generation results of the paraphrase model and summarization data for the basis of the dataset. The paraphraser[3] was chosen as it’s a free model that is provided as an API. The model was trained on 7000 examples from different sources of various domains: 1) text level - texts from different domains filtered with Bertscore (Zhang et al., 2019) and Rouge-L) 2) sentence level - the Russian version of Tapaco corpus (Scherrer, 2020) and filtered ParaphraserPlus (Gudkov et al., 2020) corpus. Russian news dataset for summarization[4] was used as the source data for models generation. From each text, a fragment consisting of 1, 2 or 3 sentences were

[3] https://habr.com/ru/company/sberdevices/blog/667106/

[4] https://huggingface.co/datasets/IlyaGusev/gazeta

taken. The collected fragments were used as input for generating statements using the paraphrase model and the evidence for the generated statements. Since the generated data may be factually inconsistent with the source texts, we annotate them manually for future reference.

Datasets Translation. The dataset also included English-language data from the FEVER fact-checking dataset (Thorne et al., 2018a) that was translated into Russian. In the FEVER dataset, the claims are classified as *Supported*, *Refuted* or *NotEnoughInfo*. For the first two classes, the annotators recorded the sentences forming the necessary evidence for their judgment. We use claims labeled *Supported* and *Refuted* and collected evidence in our work. The two NLLB-200 models[5][6] are tested for translation. We sample using the temperature of 0.85, top_k of 100, top_p of 0.8, max_length of 200 as generation parameters. We then choose the best translation using Question-Answering based metrics (Scialom et al., 2019). For each translation to assess, questions are successively generated from a source text by masking each of the named entities in this text. The results are triplets (input, question, answer), where input denotes the claim, the question refers to the sentence containing the masked entity, and the answer refers to this masked entity to retrieve. For each triple, an $F1$ score is calculated. As QA system we use the pre-trained ruBERT-large[7] fine-tuned on the SberQuAD[8] dataset. The resulting dataset included examples with a $F1$ score greater than 0.25.

Text Augmentation. The rule-based transformations (Kryściński et al., 2019b) were proposed as an alternative approach to syntectic data generation. A paraphrase dataset[9] was used as the source data. The original pairs of texts were factually consistent. A series of rule-based transformations were applied to one of the pairs obtaining factually inconsistent pairs, with one paraphrase as evidence and the other as a statement that would go through the transformations. The rule-based transformations consisted of several stages, based on the task definition criteria:

1. a randomly selected named entity in the statement was replaced with a different randomly selected named entity from the evidence text;
2. randomly selected numbers in the statement were replaced with randomly generated numbers;
3. the negative particle не was removed from the statement to change the context.

In the current work, we used the SpaCy library[10] to recognize entities. To generate additional factually inconsistent examples, available Russian corpora[11] were used. We apply the entity swapping transformation for Persons-1000[12] and Collection5[13] datasets annotated with PER, LOC, and ORG tags. For the Persons-1000 dataset, we also apply the number-swapping transformation. We use the sentence negation for the RuADReCT dataset (Tutubalina et al., 2021). We additionally manually annotate the augmented data for the test set; augmented data without manual annotation is used for the training set.

4.2 Test data

The test set consists of examples from all three sources: 26% translations, 6% augmented data, and 68% generated paraphrases. A description of the sources is presented in Section 4.1.

The test data for fact-checking was manually labeled via the crowd-sources platform Yandex.Toloka[14] (Pavlichenko et al., 2021). First, we made a classification task and asked annotators to check whether the facts in the two texts were correct. However, we faced several problems: 1) cheating and 2) misunderstanding the fact definition. It's proved that determining the truthfulness of a fact regarding a general "real world" is subjective and depends on the knowledge, values, and beliefs of the subject (Heidegger, 2005). To decrease these effects, we claim the annotators not just check the fact's coincidence but also highlight exactly the facts span. Human annotation submissions are collected

[5] https://huggingface.co/facebook/nllb-200-distilled-600M
[6] https://huggingface.co/facebook/nllb-200-distilled-1.3B
[7] https://huggingface.co/sberbank-ai/ruBert-large
[8] https://huggingface.co/datasets/sberquad
[9] https://huggingface.co/datasets/merionum/ru_paraphraser
[10] https://spacy.io/
[11] https://github.com/natasha/corus
[12] http://ai-center.botik.ru/Airec/index.php/ru/collections/28-persons-1000
[13] http://www.labinform.ru/pub/named_entities/
[14] https://toloka.ai/tolokers

and stored anonymously via the design presented in Figure 1. Each annotator is warned about potentially sensitive topics in data (e.g., politics, religion, societal minorities, etc.). The annotation details are provided in Table 1.

IAA	**Total**	**Overlap**	N_T	N_{page}	N_C	N_U	**ART**
80.2%	42$	5	8	3	50	74	113

Table 1: Details on the data collection project for the test set. **IAA** (inter-annotator agreement) refers to the IAA confidence scores. **Total** is the total cost of the annotation project. **Overlap** is the number of votes per example. N_T is the number of training tasks. N_{page} denotes the number of examples per page. N_C is the number of control examples. N_U is the number of users who annotated the tasks. **ART** means the average response time in seconds.

IAA	**Total**	**Overlap**	N_T	N_{page}	N_C	N_U	**ART**
75.1%	801$	3-5	8	3	50	181	103

Table 2: Details on the data collection project for the train set. **IAA** (inter-annotator agreement) refers to the IAA confidence scores. **Total** is the total cost of the annotation project. **Overlap** is the number of votes per example. N_T is the number of training tasks. N_{page} denotes the number of examples per page. N_C is the number of control examples. N_U is the number of users who annotated the tasks. **ART** means the average response time in seconds.

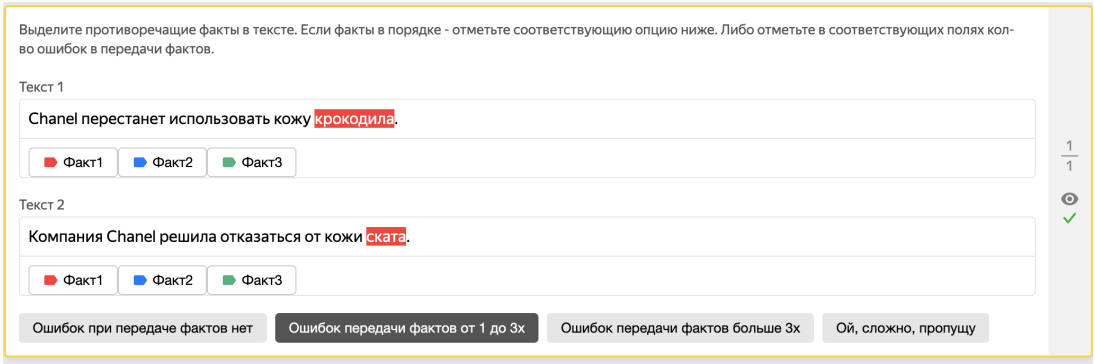

Figure 1: The example of Yandex.Toloka design setup. Two texts are provided, and annotators need to span the inconsistency of the facts. There is a required field with four options to set how many facts the texts contain.

We verify the annotator submissions' quality with control questions and exclude cheaters. The overlap is set to 5 to provide more reliable results and high confidence. We count IAA using majority votes, considering not just classification buttons but also the span overlap of annotators. Due to the complexity of the task, we exclude the examples in the set that contains less than three annotators' votes or has a low IAA. We balance the dataset to save the class distribution; dataset statistics are reported in Table 3.

4.3 Training data

Three training sets were prepared based on data from Section 4.1 to compare various approaches to creating training data for the fact-checking task.

- The first train set **Translated set** consists of translated English-language fact-checking dataset.
- The second train set **Augmented set** contains augmented data.
- The third train set **Labeled set** includes parts of the translations, augmented data and generated data. Translations and generated data were manually labeled via the crowd-sources platform Yan-

dex.Toloka. The annotation project was similar to the golden test set collection setting. The details of the train verification procedure are presented in Table 2.

Data Set	Consistent	Inconsistent	Total
Translated set	2150	2146	4296
Augmented set	1258	1434	2692
Labeled set	2994	3242	6236
Test set	250	250	500

Table 3: Statistics of data sets.

The final statistics of data sets are reported in Table 4. We split all sets into train and validation. For each dataset, we use 75% of the data as the training set and 25% as the validation set.

5 Experiments

Despite the span annotations in our data, in this paper, we define the task as a classification problem and conduct experiments for binary classification. We provide several baselines on the different train sets and fine-tune state-of-the-art models on this task.

5.1 Models

Baselines As baselines, we develop a classifier built on perplexity calculation and a classifier built on the cosine similarity calculation.

The perplexity-based approach (Lee et al., 2021) **ruGPT3-ppl** is based on including evidence in the perplexity calculation, using it as a prefix for a claim: $X = (x_{e_0}, ..., x_{e_E}, x_{c_0}, ..., x_{c_C})$, where E and C denote the number of evidence tokens and claim tokens, respectively. We obtain the perplexity of an input text as follows:

$$PPL(X) = \sqrt[C]{\prod_{i=0}^{C} \frac{1}{p_\theta(x_{c_i}|x_{e_0}, ..., x_{e_E}, ..., x_{c_{i-1}})}} \tag{1}$$

where X is an input text, C is the length of the claim. The ruGPT3-large model[15] is used to calculate perplexity. The ruGPT3 is a Russian adaptation of the autoregressive language model GPT3 (Brown et al., 2020).

The cosine similarity approach **LaBSE-sim** is based on calculating the cosine similarity between embeddings. We use the LaBSE model[16] (Feng et al., 2020) to obtain embeddings of the evidence e and claim c texts, then we calculate the cosine similarity between them:

$$\cos(\theta) = \frac{e \cdot c}{\|e\|\|c\|} \tag{2}$$

Optimal threshold values are determined for baseline models that effectively distinguish between factually consistent and inconsistent claims. The training set is utilized to identify the hyper-parameter value that yields the highest level of performance for the threshold parameter, denoted as th, without requiring any parameter updates to pre-existing language models.

Fine-tuned models We fine-tune pre-trained Transformer-based models on the collected training datasets to build baseline classifiers. Three state-of-the-art models of different size are considered:

- ruBERT-base[17] is a Russian BERT model (Devlin et al., 2019) trained 30 GB Russian filtered dataset (including domains: Wikipedia, news, part of the Taiga corpus, fiction, etc.),
- ruRoberta-large[18] is a RoBERTa model (Liu et al., 2019) trained on 250GB Russian dataset,

[15] https://huggingface.co/sberbank-ai/rugpt3large_based_on_gpt2
[16] https://huggingface.co/sentence-transformers/LaBSE
[17] https://huggingface.co/sberbank-ai/ruBert-base
[18] https://huggingface.co/sberbank-ai/ruRoberta-large

• ruGPT3-small[19] is a small version of ruGPT from the ruGPT-family [20].

We fine-tune ruBERT-base and ruRoberta-large models with a single-layer classifier on top. We concatenate the evidence e and the claim c, insert [SEP] token between them and add [CLS] to make the sequence. This sequence is fed as input to the model for binary classification.

For the ruGPT3-large, the input prompt sequence for the task is written as follows:

Доказательство: [e]

Утверждение: [c]

Доказательство подтверждает утверждение:

We fine-tuned ruGPT3-large to generate the target tokens Да (Yes) or Нет (No).

5.2 Experimental Setup

Evaluation metrics Since we consider the fact-checking task a binary classification problem for a balanced test set, we used accuracy as the primary metric to evaluate models. We also used precision, recall, and F1-score as additional metrics. For fine-tuned models, we report the average results across five runs with different random seeds (the standard deviation is presented in Table 4).

Training Details During our experiments, we set the maximum sequence length to 512 and used a batch size of 16. Models were trained for seven epochs using the Adam optimizer (Kingma and Ba, 2014). For ruGPT3, we used a learning rate of $5e{-}5$, while for ruBERT and ruRoberta we employed the Adam optimizer with a learning rate of $1e{-}5$ was used. The best model checkpoints were selected based on performance on the validation set.

6 Evaluation

Model	Training Set	Accuracy	F1	Precision	Recall
ruGPT3-ppl	Translated set	56.0	57.2	55.7	58.8
ruGPT3-ppl	Augmented set	56.2	63.8	54.4	77.2
ruGPT3-ppl	Labeled set	56.4	62.8	54.8	73.6
LaBSE-sim	Translated set	62.8	55.1	69.5	45.6
LaBSE-sim	Augmented set	51.6	67.3	50.8	99.6
LaBSE-sim	Labeled set	63.2	65.4	61.7	69.6
ruBERT-base	Translated set	57.4 (±0.52)	33.3 (±1.31)	76.8 (±2.59)	21.3 (±1.11)
ruBERT-base	Augmented set	52.4 (±1.07)	63.0 (±0.54)	51.5 (±0.70)	81.1 (±0.59)
ruBERT-base	Labeled set	63.4 (±0.59)	65.7 (±1.12)	61.9 (±0.43)	70.0 (±2.57)
ruRoBERTa-large	Translated set	60.2 (±0.66)	41.8 (±3.27)	**78.0** (±4.77)	28.8 (±3.65)
ruRoBERTa-large	Augmented set	55.3 (±0.83)	63.5 (±1.60)	53.7 (±0.58)	77.8 (±4.54)
ruRoBERTa-large	Labeled set	**66.0** (±1.49)	**68.0** (±1.03)	64.5 (±2.43)	72.2 (±3.73)
ruGPT3-small	Translated set	53.8 (±1.17)	49.3 (±2.38)	54.6 (±1.38)	45.1 (±3.75)
ruGPT3-small	Augmented set	42.2 (±0.58)	56.4 (±1.18)	45.3 (±0.48)	74.7 (±2.83)
ruGPT3-small	Labeled set	54.4 (±1.99)	57.1 (±0.98)	54.2 (±2.44)	60.9 (±4.76)

Table 4: Results of models fine-tuned on each training set and evaluated on the test set. We report the mean and standard deviation (in parentheses) across 5 runs with different random seeds for fine-tuned models.

Our experiments assess the impact of different training datasets on model performance. We report the results in Table 4, which displays the accuracy of the fine-tuned models on *Translated*, *Augmented*, and *Labeled* training sets, evaluated on our manually labeled test set. Based on our accuracy metrics, all models perform best when trained on the *Labeled* set. Specifically, the ruRoBERTa-large model trained on the *Labeled* set achieves the highest accuracy score of 66.0% accuracy and F1-score of 68.0%. These

[19] `https://huggingface.co/sberbank-ai/rugpt3small_based_on_gpt2`

[20] https://sbercloud.ru/ru/datahub/rugpt3family

results can be attributed (i) to the diversity of data sources included in the sample and (ii) to the manual annotation of the collected data, which enhances the quality of data labeling.

Experimental results reveal a decrease in performance metrics when using the *Translated* set for fine-tuning. This can be attributed to the fact that the *Translated* set is composed of automatically translated texts, which may contain mistranslations, especially in the case of named entities and language peculiarities. Therefore, using such translated data may result in poorer model performance compared to the *Labeled* set, which benefits from manual annotation, contains various data sources, and is more reliable.

In our experiments, we observed that using LaBSE-sim on the *Augmented* set resulted in a high F1-score comparable to the best-performing ruRoBERTa-large model, and low, almost random accuracy metrics. This can be attributed to the high recall but low precision of the LaBSE-sim approach. It appears that there is a possibility that the finding of the threshold on synthetic augmented sets can increase model recall in the cases of simple fact contradictions and replacements similar to the FactCC approach. However, this method may not be sufficient for catching more complex fact inconsistencies, as the test set contains more complex cases that cannot be identified solely based on factual inconsistency class replacements.

According to our results, the perplexity-based approach, ruGPT3-ppl, outperforms the ruGPT-small fine-tuned on the classification task. This coincides with the Russian SuperGLUE leaderboard[21], which shows that the ruGPT3-small is not performing well in classification tasks, particularly those based on NLI, perhaps due to its generative pre-training nature. In contrast, the ruGPT3-ppl approach demonstrates consistent results. We suggest that a larger model, such as the ruGPT3 XL, may exhibit more generalization abilities and improve the perplexity-based approach's overall performance.

The experimental results on the proposed datasets demonstrate an overall accuracy close to 70%. This performance level is comparable to that achieved by state-of-the-art models on analogous benchmarks for the English language, such as the FEVER leaderboards (Thorne et al., 2018a) (Thorne et al., 2018b). Moreover, the TRUE benchmark for English also reported comparable F1 scores for a similar task and highlighted that NLI-based models, for example, Adversarial NLI (Nie et al., 2020), outperformed other approaches (Honovich et al., 2022). This observation is not surprising given the complexity of the collected dataset, which requires models to exhibit robust reasoning capabilities. In fact, the nature of factual consistency in the text is more intricate than just simple sentence structures, necessitating more nuanced and sophisticated approaches to capture and classify factual information accurately.

7 Conclusion

This paper investigates the problem of internal fact-checking and the ability of large language models to preserve factual consistency. We introduce a new evidence-based fact-checking dataset and benchmark for the Russian language, which is publicly available[22]. To expand the training set, we utilize data augmentation techniques and compare classification methods on various augmented datasets. Based on our analysis of model performances, we find out that the pre-trained ruRoBERTa-large model fine-tuned on manually annotated data yields the best results. Furthermore, we have launched a competition[23] and present a public leaderboard for the proposed task. In future research, we plan to explore using factual inconsistency spans for model training and treating the task as a token classification problem. Additionally, we aim to address the challenges associated with evaluating factual consistency and explore the integration of NLI-based methods into our current approach.

References

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. // *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, P 610–623.

[21] https://russiansuperglue.com/leaderboard/2

[22] `https://huggingface.co/datasets/akozlova/RuFacts`

[23] `https://www.kaggle.com/competitions/internal-fact-checking-for-the-russian-language`

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.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. *arXiv preprint arXiv:2010.08712*.

Subhajit Chaudhury, Sarathkrishna Swaminathan, Chulaka Gunasekara, Maxwell Crouse, Srinivas Ravishankar, Daiki Kimura, Keerthiram Murugesan, Ramón Fernandez Astudillo, Tahira Naseem, Pavan Kapanipathi, et al. 2022. X-factor: A cross-metric evaluation of factual correctness in abstractive summarization. // *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, P 7100–7110.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. // *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)*, P 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Alexander R Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2021. Qafacteval: Improved qa-based factual consistency evaluation for summarization. *arXiv preprint arXiv:2112.08542*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Max Glockner, Yufang Hou, and Iryna Gurevych. 2022. Missing counter-evidence renders nlp fact-checking unrealistic for misinformation. *arXiv preprint arXiv:2210.13865*.

Vadim Gudkov, Olga Mitrofanova, and Elizaveta Filippskikh. 2020. Automatically ranked russian paraphrase corpus for text generation. *arXiv preprint arXiv:2006.09719*.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. *arXiv preprint arXiv:2106.09248*.

Martin Heidegger. 2005. On the essence of truth. *Truth: Engagements across philosophical traditions*, P 244–260.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. *arXiv preprint arXiv:2204.04991*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019a. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019b. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

Nayeon Lee, Yejin Bang, Andrea Madotto, Madian Khabsa, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. *arXiv preprint arXiv:2103.09535*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mohsen Mesgar, Edwin Simpson, and Iryna Gurevych. 2020. Improving factual consistency between a response and persona facts. *arXiv preprint arXiv:2005.00036*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding.

Nikita Pavlichenko, Ivan Stelmakh, and Dmitry Ustalov. 2021. Crowdspeech and vox diy: Benchmark dataset for crowdsourced audio transcription. // J. Vanschoren and S. Yeung, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Yves Scherrer. 2020. Tapaco: A corpus of sentential paraphrases for 73 languages. // *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association (ELRA).

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. *arXiv preprint arXiv:1909.01610*.

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

Anatoly S Starostin, Victor V Bocharov, Svetlana V Alexeeva, Anastasiya A Bodrova, Alexander S Chuchunkov, SS Dzhumaev, Irina V Efimenko, Dmitry V Granovsky, Viktor F Khoroshevsky, Irina V Krylova, et al. 2016. Factrueval 2016: evaluation of named entity recognition and fact extraction systems for russian.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2022. Evaluating the factual consistency of large language models through summarization. *arXiv preprint arXiv:2211.08412*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The FEVER2.0 shared task. // *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*.

Elena Tutubalina, Ilseyar Alimova, Zulfat Miftahutdinov, Andrey Sakhovskiy, Valentin Malykh, and Sergey Nikolenko. 2021. The russian drug reaction corpus and neural models for drug reactions and effectiveness detection in user reviews. *Bioinformatics*, 37(2):243–249.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.