# RuWikiBench: Evaluating Large Language Models through replication of encyclopedia articles

*D. A. Grigoriev*[1,2], *D. I. Chernyshev*[1,3]

© The Authors 2025. This paper is published with open access at SuperFri.org

In light of the growing interest in using large language models (LLMs) as tools for generating scientific texts, the evaluation of their ability to produce encyclopedic content is becoming increasingly relevant. However, for Russian-language materials this issue has not been sufficiently studied, and existing benchmarks do not cover key aspects of analytical work with sources. This paper presents RuWikiBench - an open benchmark based on Ruwiki for evaluating the ability of large language models to reproduce Wikipedia-style articles, built around three tasks: selection of relevant sources, article structuring, and section generation. The results of testing popular open-source LLMs show that even under ideal conditions, the best models do not always follow the expert logic of composing encyclopedic content: even with a perfect source retrieval system, the models cannot reproduce the reference table of contents, and the quality of section generation shows almost no dependence on the number of parameters.

*Keywords: benchmark, Wikipedia, Ruwiki, large language model.*

## Introduction

Modern large language models demonstrate impressive results in generating texts of various styles and themes. However, their capabilities for working with scientific and encyclopedic materials remain understudied, particularly for Russian-language texts. Existing methods for evaluating model capabilities predominantly focus on standard linguistic tasks, without paying sufficient attention to analytical abilities when working with scientific texts. For the Russian language, this problem is especially relevant due to the limited availability of specialized evaluation tools.

There are many benchmarks covering various linguistic tasks for the Russian language. RussianSuperGlue [1] evaluates general language understanding and basic natural language processing tasks. MERA [2] provides unified testing conditions for models by compiling generation instructions for each task; however, the tasks themselves are oriented towards testing general comprehension. LIBRA [3] focuses on testing a model's ability to retain and retrieve information from a large context but is centered on short answers that do not require deep reasoning. Ru Arena General [4] focuses on pairwise model comparison rather than overall answer quality. Ping-Pong [5] evaluates the dialog abilities of models, which is important for interactive systems, but is not suitable for assessing the ability to conduct research and write coherent scientific-encyclopedic texts. At the same time, an entire class of tasks related to deep text analysis remains uncovered: creating detailed, structured, and factually accurate texts supported by a large number of sources.

The recent development of new agent capabilities, such as the emergence of the "Deep Research" function by OpenAI [6] or the development of the universal Storm algorithm [7], indicates a growing interest in conducting scientific research using large language models. This highlights the need to create new approaches for objectively evaluating the analytical capabilities of models. Existing benchmarks only partially address aspects critical for generating scientific-encyclopedic texts, such as the ability to synthesize information from a set of documents, plan the structure of a future text, maintain coherence and logical sequence of presentation, as well as ensure the

[1]Lomonosov Moscow State University, Moscow, Russia
[2]E-mail: dagrig14@yandex.ru
[3]E-mail: chdanorbis@yandex.ru

accuracy and reliability of facts. One of the closest studies in this area is the ResearchArena [8] benchmark, which formalizes the construction of an academic review; however, it is more aimed at testing the models' ability to select and organize relevant information and does not address their ability to generate coherent scientific-encyclopedic texts.

This paper proposes an approach aimed at creating tools to test how well large language models can work with scientific-encyclopedic texts. Within the framework of this research:

1. A labeled dataset based on the online encyclopedia "Ruwiki" has been collected;
2. An open benchmark, RuWikiBench, has been developed to measure model quality on tasks requiring deep text analysis;
3. The abilities of the best open large language models to generate Wikipedia-style articles have been tested.

The code and data of this work have been made publicly available[4].

## 1. Dataset collection

To build a benchmark aimed at assessing the ability of language models to work with article sources, it is necessary to prepare a corpus of texts that will be used in generation. The choice was made in favor of the Wikipedia style because this genre simultaneously requires factual accuracy, completeness of analysis, and understanding of context, which aligns well with the research direction of this work.

The Russian online encyclopedia "Ruwiki" was chosen as the source. It is distinguished by a large number of references to Russian-language sources, as well as stricter text filtering, which allows it to be relied upon as a reliable benchmark for assessing the quality of generating Russian-language articles.

The data acquisition process included the following steps:

1. **Article Selection**: Articles on diverse topics containing a sufficient number of references to external sources were manually selected;
2. **Source Downloading**: For each article, the available sources it references were automatically gathered;
3. **Splitting into Snippets**: To reproduce real Retrieval Augmented Generation (RAG) conditions, all texts were split into small fragments of approximately $\approx 600$ words in length.

During the data acquisition stage, primary information extraction from the selected article and the collection of its associated sources are performed. Figure 1 shows a brief schematic of the source text extraction process. The sources were downloaded using the Python module `newspaper3k`[5]. A subset of "Ruwiki" articles BB is taken as the initial corpus. The extraction of the article's HTML code is performed using standard Python module tools[6,7]. The obtained text is structured by splitting it into fragments corresponding to nested headings (H1, H2, H3, etc.), which preserves both the substantive part of the article and its hierarchical organization. Next, all external references cited in the "Notes" section are automatically extracted. Invalid links (e.g., 404 error) are excluded from further processing, and the text associated with them is removed, leaving only those sources that are actually accessible.

[4]https://github.com/Nejimaki-Tori/WikiBench
[5]https://github.com/codelucas/newspaper
[6]https://beautiful-soup-4.readthedocs.io/en/latest/
[7]https://requests.readthedocs.io/en/latest/index.html

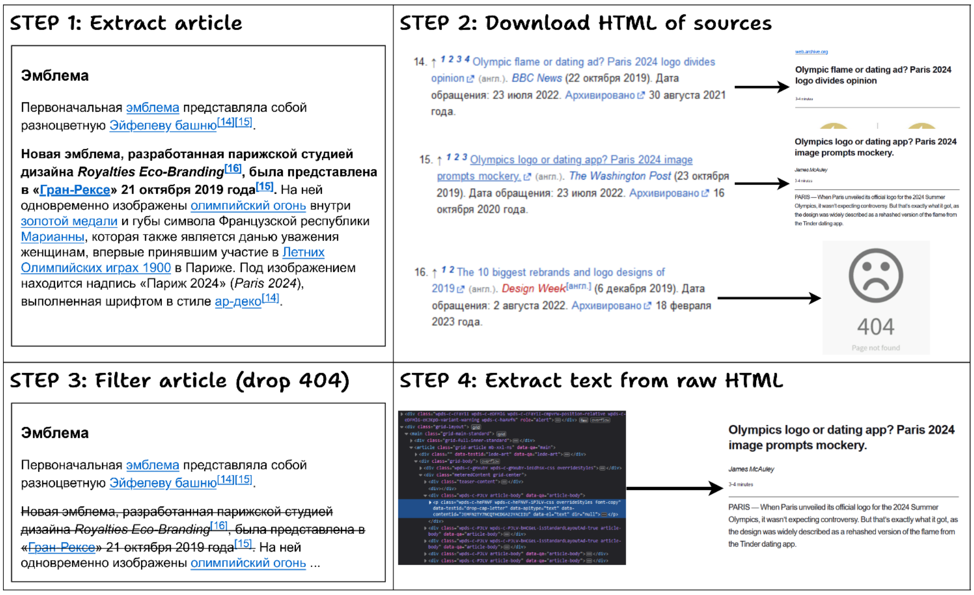


**Figure 1.** Source Extraction

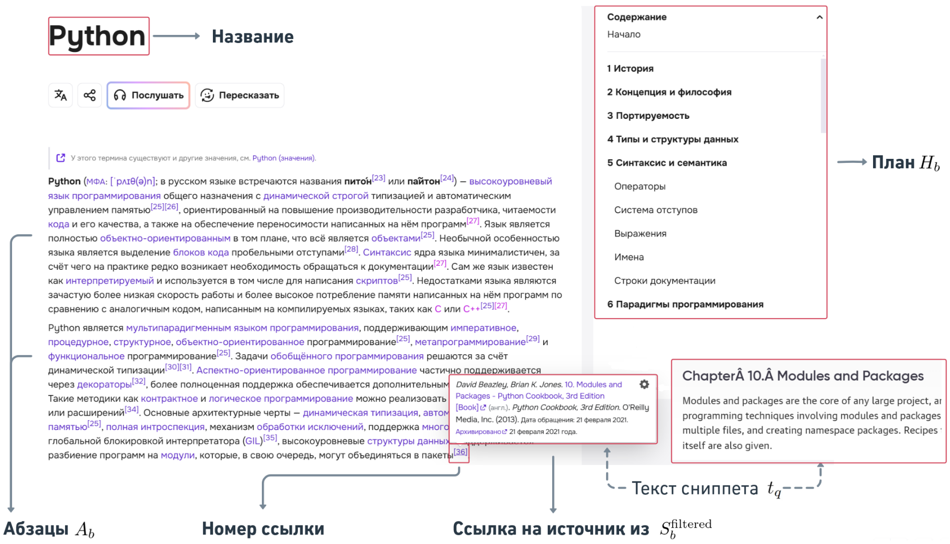


**Figure 2.** Main Article Entities

Figure 2 illustrates the schematic breakdown of an article[8] into key entities used in subsequent processing. During the data processing stage, text filtering is performed to ensure its correct interpretation by the model. Each footnote (e.g., [1], [2]) is matched to a specific link corresponding to one of the available sources. This allows for precise identification of the link's position in the article text and its use for subsequent filtering.

Based on the valid links $S_b^{\text{filtered}}$, filtered sets of paragraphs $A_b^{\text{filtered}}$ and headings $H_b^{\text{filtered}}$ are formed. That is, only content supported by the extracted sources is retained; everything else is removed. Only sources for which a text $t_q$ of at least 1500 characters could be retrieved are kept, to filter out "noisy" responses from HTML pages such as errors (e.g., error 404) or blocking messages. $A_b^{\text{filtered}}$ retains only those paragraphs that contain at least one link to a source for which text was

[8] `https://ru.ruwiki.ru/wiki/Python`

**Table 1.** Key characteristics of the collected dataset

| Metric | RuWikiBench | ResearchArena |
|---|---|---|
| Number of articles | 285 | 7,952 |
| Number of downloaded sources | 15,686 | 12,034,505 |
| Total number of snippets | 36,860 | - |
| Average plan size (number of headings) | 37 | - |
| Average section size (number of words) | 112 | - |

successfully retrieved. Similarly, $H_b^{\text{filtered}}$ is formed—only those headings under which at least one paragraph remains. The characteristics of the collected corpus are presented in Tab. 1.

## 2. Evaluation Methodology

To objectively assess the ability of language models to generate scientific and encyclopedic texts, it is necessary to replicate the real process of preparing encyclopedic content:

1. **Selection of relevant sources**: The model is given the article title and a set of snippets, among which it must identify and rank materials relevant to the topic by their significance;
2. **Article structure construction**: Based on the topic and selected sources, the model creates an outline with main sections in the Wikipedia style;
3. **Section generation**: Article materials are distributed across sections, after which a summary of the relevant materials is generated for each section.

Each stage is evaluated independently of the previous ones, allowing for a quantitative measurement of the quality of each specific subtask.

### 2.1. Selection of Relevant Sources

One of the most effective search strategies [9] is the preliminary generation of an expected result (description) based on the original query (article title) to create an expanded search query. The description is generated in both Russian and English, as the source texts are also available in both languages. The queries in both languages are then combined into a single textual query for a BM25-based search system.

Experiments were conducted with two approaches to query formulation:

1. **Pre-generated query based on the title and second-level headings**: Allows for a clean evaluation of the models' ranking abilities; the LLaMa 3 70b model [10] was used to generate the query;
2. **Query generated from the title by the evaluated model**: Similar to real-world conditions, the LLM is fully responsible for the quality of the results and independently decides which search query to formulate for BM25.

Examples of generated descriptions are shown in Figure 3.

The documents selected by the BM25 query are sequentially passed to the large language model, which must classify each snippet as relevant (answer "yes") or non-relevant (answer "no"). To obtain numerical scores, the titles of the articles to which the retrieved documents belong are compared with the title of the article for which source texts are being selected. The logarithmic probability of the tokens in the model's response is taken: if the response was affirmative, the probability $P(\text{yes})$ is used; if negative, $1-P(\text{no})$ is used. This approach allows ranking the retrieved

| Вариант запроса | Текст |
|---|---|
| **Генерация по заголовкам** | Статья "C++"представляет собой обзор языка программирования C++, его истории, структуры и особенностей. В ней рассматриваются основные аспекты языка, включая его стандартную библиотеку, отличия от языка C и дальнейшее развитие. Кроме того, статья содержит примеры программ на C++, сравнение с альтернативными языками программирования, а также критический анализ и обсуждение влияния C++ на развитие программирования и существующие альтернативы. Статья предназначена для читателей, интересующихся языком C++ и его ролью в современном программировании. |
| **Генерация по названию** | Статья "C++"может быть посвящена языку программирования C++, являющимся одним из наиболее популярных и широко используемых языков программирования в мире. В статье могут быть рассмотрены основы языка, его история, синтаксис и особенности, а также его применение в различных областях, таких как разработка операционных систем, игр и веб-приложений. Кроме того, статья может содержать информацию о стандартах и библиотеках C++, а также о его сравнении с другими языками программирования. Статья может быть полезна как для начинающих программистов, так и для опытных специалистов, которые хотят углубить свои знания о языке C++. Статья также может включать примеры кода и практические советы по использованию C++ в реальных проектах. |

**Figure 3.** Comparison of descriptions for the article «C++» using the two approaches

documents by the model's confidence in their relevance: the higher the probability, the higher the model's confidence in the response, and the higher the document is ranked in the results.

## 2.2. Article structure construction

First, each text fragment (snippet) from the reference article source is converted into a vector representation using a selected embedding model. Then, the snippets are clustered into potential section contents. For determinism, the KMeans algorithm is applied with the number of clusters equal to the number of second-level headings in the reference outline, and the centroids are initialized with the vector representations of these headings.

Next, five snippets closest to the cluster center are selected. This is done to reduce the influence of less relevant snippets on the final outline. The formation of mini-outlines for sections is carried out taking into account two key parameters: the context window size (to account for references and the overall semantics of the document) and two generation modes — directly from the texts and through preliminary generation of a brief cluster description. The two generation modes allow for choosing the level of abstraction: the direct mode preserves details with raw data, while the mode via preliminary cluster description improves consistency of formulations and reduces information duplication. At the final stage, all mini-outlines are combined into the final structured article outline.

## 2.3. Section generation

For each article section, all snippets that were indicated as sources for the reference text of the section are extracted. All snippets are again converted into embeddings, and a pairwise similarity matrix is constructed as the product $E \times E^{\top}$. Elements with a similarity value above the threshold of 0.8 (empirically determined) are considered semantically close and are grouped together to avoid redundant repetitions during generation (e.g., when different sources paraphrase the same information). For each such semantic group, a hierarchical representation is built: the first five texts are taken, and a brief description is generated based on them. This description is then supplemented using the next five texts, and so on, until a complete compressed representation of the group is obtained. Thus, only a set of brief descriptions remains — the most important information without unnecessary repetition. After this, the text of the section is generated based on the obtained group descriptions using a hierarchical summarization method [11].

# 3. Experimental Setup Description

Below is a description of all data, models, hyperparameters, and procedures used to ensure reproducibility and analysis.

## 3.1. Generation Parameters

For all models, unless otherwise specified, the same generation parameters were used: temperature - 0.01, repetition penalty - 1.0, and top_p - 0.9.

## 3.2. Relevant Source Selection

Snippet indexing was performed using BM25[9] across the entire corpus of collected snippets without hyperparameter tuning (default values). For each relevant document, two non-relevant documents were selected (ratio 1:2) — this was done to improve evaluation robustness.

## 3.3. Article Structure Construction

Snippets were converted into vector space using the `sergeyzh/BERTA`[10] model. Two context window options were considered: either a zero window (only the snippet itself) or one neighboring snippet to the left and right to expand the context. Header similarity with reference headers was compared using cosine similarity: semantic correspondence was prioritized over exact wording or header level. The comparison was performed against the cleaned article structure: all headers whose sections consisted entirely of text without downloadable sources were removed from the preprocessed text.

# 4. Evaluation Metrics

Within the benchmark, two groups of metrics are used: ranking metrics, which assess how well the model selects relevant sources, and text similarity metrics, which measure how closely the generated content matches the reference.

## 4.1. Ranking Metrics

To evaluate the quality of the source list, we use **NDCG@K** [12] and R-Precision [13]:

$$\mathrm{NDCG@K} = \frac{\mathrm{DCG@K}}{\mathrm{IDCG@K}} \tag{1}$$

[9]`https://github.com/xhluca/bm25s`
[10]`https://huggingface.co/sergeyzh/BERTA`

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{\text{rel}_i}{\log_2(i+1)} \tag{2}$$

$$\text{IDCG@K} = \sum_{i=1}^{K} \frac{\text{rel}_i^{\text{IDEAL}}}{\log_2(i+1)} \tag{3}$$

$$\text{R-Precision} = \frac{\sum_{i=1}^{R} \text{rel}_i}{R} \tag{4}$$

where $\text{rel}_i \in \{0, 1\}$ is the indicator of relevance for the document at position $i$; $\text{rel}_i^{\text{IDEAL}}$ is the same quantity in the ideal (fully sorted) ranking; $R$ is the total number of relevant documents for the given query.

## 4.2. Text Similarity Metric

The quality of generated sections and headings is evaluated with **BERTScore** [14]:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^{\top} \hat{x}_j \tag{5}$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^{\top} \hat{x}_j \tag{6}$$

$$F_{\text{BERT}} = \frac{2\, P_{\text{BERT}}\, R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \tag{7}$$

where $x$ is the reference text, $\hat{x}$ is the generated text; each sentence is encoded using the model[11], after which cosine similarity is computed.

We also considered ROUGE-L and BLEU for evaluating section generations.

**ROUGE-L** [15] is based on the length of the longest common subsequence (LCS) between the generated summary $S$ and the reference $R$:

$$\text{Precision} = \frac{\text{LCS}(S, R)}{|S|}, \tag{8}$$

$$\text{Recall} = \frac{\text{LCS}(S, R)}{|R|} \tag{9}$$

$$\text{ROUGE-L} = \frac{2\ \text{Precision}\ \cdot\ \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

**BLEU** [16] is an n-gram precision metric with a brevity penalty. The final score is given by Equation (11):

$$\text{BLEU}_N = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{11}$$

where $p_n$ is the precision for $n$-grams, $w_n$ are the weights, and BP is the brevity penalty.

[11] `https://huggingface.co/sergeyzh/BERTA`

## 5. Description of Experimental Results

### 5.1. Models

The experiments used the following large language models: RuadaptQwen2.5-7B-Lite-Beta [17], RuadaptQwen3-32B-Instruct-v2 [17], DeepSeek V3 [18], Qwen3-235B-A22B [19], tpro [20] and yagpt5lite [21]. In all tables, models are ordered by size, and the best results within each parameter group are highlighted.

### 5.2. Results

Tables 2 and 3 present the results of measuring ranking quality. We also include baseline results, represented by BM25 retrieval without model-based reranking. In the first case (Table 2), where a pre-generated search query was used for all models, the best results were achieved by DeepSeek V3, which indicates a strong ability to select relevant documents. In the second experiment (3), where the query was formed solely from the article title, tpro achieved the best results. The experiment showed that BM25's own query generation is not inferior in ranking quality to the reference setup, in which queries are generated by a more "powerful" model and then used by all evaluated models. Presumably, this is because—as shown in Figure 3—the queries turn out to be similar: LLMs have knowledge of Wikipedia's typical article structure from training and can therefore connect relevant concepts in the required format.

**Table 2.** Results of pure ranking ability evaluation

| Model | NDCG | R-Pr |
|---|---|---|
| baseline (bm25) | 88.81 | 62.51 |
| **DeepSeek V3** | **95.42** | **83.86** |
| Qwen3-235B-A22B | 94.49 | 82.42 |
| RuadaptQwen3-32B-Instruct-v2 | 95.25 | 81.81 |
| tpro | 95.42 | 83.53 |
| RuadaptQwen2.5-7B-Lite-Beta | 88.26 | 62.26 |
| yagpt5lite | 90.35 | 77.66 |

**Table 3.** Results of evaluating BM25 query-generation ability

| Model | BM25 | | Rerank | |
|---|---|---|---|---|
| | NDCG | R-Pr | NDCG | R-Pr |
| DeepSeek V3 | 88.39 | 60.65 | 95.67 | 83.07 |
| Qwen3-235B-A22B | 89.17 | 62.98 | 94.90 | 81.96 |
| RuadaptQwen3-32B-Instruct-v2 | 85.39 | 52.80 | 95.82 | 81.62 |
| **tpro** | **90.61** | **65.07** | **96.06** | **83.37** |
| RuadaptQwen2.5-7B-Lite-Beta | 88.81 | 62.51 | 88.23 | 60.96 |
| yagpt5lite | 86.59 | 57.98 | 90.27 | 77.65 |

Overall, the models show fairly high metric values at this stage, which may be due to the fact that an article title reflects its content well. In the best cases, up to 80% of the documents in the sample are relevant, which can be considered a good figure; however, there remains potential for further improvement.

**Table 4.** Results of outline generation

| Model | Mean BERTScore F1 | | |
|---|---|---|---|
| | Direct | | Description |
| | no neighbors | one neighbor | |
| **DeepSeek V3** | **63.51** | **62.93** | **65.50** |
| Qwen3-235B-A22B | 60.86 | 59.06 | 62.66 |
| RuadaptQwen3-32B-Instruct-v2 | 60.12 | 60.04 | 62.91 |
| tpro | 60.32 | 59.09 | 60.75 |
| RuadaptQwen2.5-7B-Lite-Beta | 60.03 | 58.21 | 61.58 |
| yagpt5lite | 59.72 | 60.07 | 60.25 |

Table 4 presents the results of evaluating article-structure construction. Direct — generation from the cluster snippets, with neighbor context as indicated; Description — generation via a preliminary description of all cluster elements. The results show that with preliminary description generation, all models consistently improve in quality. RuadaptQwen3 shows the largest gain, rising to second place and effectively matching the results of the larger model, Qwen3-235B-A22B. DeepSeek V3 remains the leader, showing a substantial margin over the others. At the bottom in quality are RuadaptQwen2.5-7B-Lite-Beta and yagpt5lite. At the same time, yagpt5lite, with only 8 billion parameters, delivers results comparable to a 32-billion-parameter model. Figure 4 shows a comparison of a small excerpt of the reference and generated outlines. The obtained results correlate well with the degree of similarity between the headings and the reference. A common issue across all models was excessive heading hierarchy depth. On "Ruviki", headings were rarely deeper than level three; however, the models often created fourth- and fifth-level headings, implying that all information belongs in one large section, even though it may differ somewhat in meaning and, in the original outline, would correspond to unrelated headings.

| СГЕНЕРИРОВАННЫЙ | ЭТАЛОННЫЙ |
|---|---|
| # Введение в Python | # Python |
| ## Обзор языка | ## История |
| ### История и основные аспекты | ## Концепция и философия |
| #### Ключевые особенности и реализации | ## Портируемость |
| # Основы языка Python | ## Типы и структуры данных |
| ## Синтаксис и семантика | ## Синтаксис и семантика |
| ### Типы данных и структуры | ### Система отступов |
| #### Числа, списки, словари и объектно-ориентированное программирование | ### Выражения |
| # Продвинутые темы Python | ### Имена |
| ## Контроль потока и многопоточность | ### Строки документации |
| ... | ## Парадигмы программирования |
| | ... |

**Figure 4.** Comparison of two article outlines

Table 5 reports the measurements of section-generation quality. The final results are at roughly the same level, but this is due to the sensitivity of the metric used. Sections for which the algorithm did not select a single relevant snippet were excluded from the final metrics. The best overall results were demonstrated by Qwen3-235B-A22B; however, in terms of ROUGE-L and BLEU, RuadaptQwen3-32BInstruct-v2 leads, indicating better structural consistency and greater overlap of wording with the reference. The yagpt5lite model performs above average—especially on BLEU—at a much smaller size, whereas tpro shows the lowest values across all metrics.

**Table 5.** Results of section generation

| **Model** | **Mean F1** | **Mean ROUGE-L** | **Mean BLEU** |
|---|---|---|---|
| DeepSeek V3 | 53.48 | 14.34 | 2.81 |
| Qwen3-235B-A22B | **53.74** | 14.63 | 3.07 |
| **RuadaptQwen3-32B-Instruct-v2** | 53.21 | **15.46** | **3.40** |
| tpro | 53.15 | 13.58 | 2.27 |
| RuadaptQwen2.5-7B-Lite-Beta | 52.99 | 12.29 | 2.11 |
| yagpt5lite | 53.43 | 14.85 | 3.16 |

For a visual comparison of section-generation quality, consider the introductory parts of the article “COVID19” produced by DeepSeek V3 and yagpt5lite, respectively, shown in Figure 5. Despite some semantic inaccuracies (for example, the statement “COVID-19 is a pandemic,” whereas in reality it is a disease), yagpt5lite demonstrates a quite solid result. Its text falls short of DeepSeek V3’s version in terms of coverage and systematic exposition, but contains more numerical data and concrete facts. At the same time, the material generated by DeepSeek V3 reads like an excerpt from an encyclopedic article, whereas the yagpt5lite version is closer in style to a technical report on the disease.

| **Модель** | **Текст** |
|---|---|
| **DeepSeek V3** | **COVID19 инфекционное заболевание**, ... которое **привело к глобальной пандемии**, начавшейся в 2020 году. Первоначально проявляясь респираторными симптомами, такими как кашель, лихорадка и одышка, болезнь может вызывать тяжёлые осложнения, включая поражение нервной, сердечно-сосудистой и других систем из-за наличия рецептора ACE2 в различных тканях организма. Вирус отличается высокой контагиозностью, предположительно имеет зоонозное происхождение и быстро распространился по всему миру. Для контроля пандемии ВОЗ рекомендует вакцинацию, ношение масок, социальное дистанцирование и гигиену рук, при этом эффективность вакцин против исходного штамма достигает 85% и более. Хотя у детей COVID-19 чаще протекает в лёгкой форме, возможны тяжёлые случаи, включая мультисистемный воспалительный синдром. Глобальные усилия по вакцинации направлены на достижение коллективного иммунитета, однако сохраняются проблемы неравномерного доступа к вакцинам и их восприятия населением. |
| **yagpt5lite** | **COVID19 это пандемия**, вызванная новым коронавирусом SARSCoV2. По состоянию на 14 января 2022 года ВОЗ подтвердила около **318 648 834** случаев COVID-19 во всём мире, включая **5 518 343** летальных исхода. Первая вакцина от COVID19 была представлена в декабре 2020 года. 2 декабря 2020 года Великобритания стала первой страной, одобрившей вакцину PfizerBioNTech (BNT162), которую ВОЗ одобрила для экстренного использования. SARSCoV2 считается более заразным, чем SARSCoV, и быстро распространился по всему миру после нескольких случаев заражения в Ухане, Китай. Патогенез SARSCoV2 связан с воспалительными реакциями, которые негативно влияют на лёгкие и вызывают такие симптомы, как кашель, лихорадка, общее недомогание, одышка и дыхательная недостаточность. Вирус может распространяться и на другие органы, вызывая неврологические, сердечно-сосудистые, кишечные и почечные нарушения. |

**Figure 5.** Comparison of two section texts

## Conclusion

This paper proposes and implements the RuWikiBench benchmark for evaluating the analytical capabilities of large language models in generating scientific and encyclopedic texts in Russian. The core of the proposed evaluation system is a three-stage process, consisting of three independent systems that naturally arise when creating articles on a specific topic. Relying on a filtered "Ruwiki" corpus with aligned snippets and a clearly defined evaluation methodology, the proposed benchmark establishes a foundation for further research in the application of language models to the task of generating scientific and encyclopedic text.

Experiments showed that with a fixed search query, DeepSeek V3 demonstrates the best source selection quality, significantly outperforming BM25 without reranking. At the structure construction stage, it was found that adding a preliminary cluster description consistently improves the quality of outlines for all models, including DeepSeek V3, which demonstrated the best understanding of the process. All models showed comparable text generation quality; however, RuadaptQwen3-32B-Instruct-v2 leads in ROUGE-L and BLEU metrics, indicating a text structure more consistent with the reference. The work shows that models possess significant potential, but their reliable application requires further development of methods for analyzing and structuring review materials.

## Acknowledgements

The study was supported by grant No. 25-11-00191 from the Russian Science Foundation. The work was carried out using the supercomputer "MSU-270" of the Lomonosov Moscow State University.

## References

1. *RussianSuperGlue.* Shavrina, Tatiana, et al. "RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark." Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020.

2. *Mera.* Fenogenova, Alena, et al. "MERA: A Comprehensive LLM Evaluation in Russian." Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024.

3. *LIBRA.* Churin I. et al. Long Input Benchmark for Russian Analysis //CoRR. – 2024.

4. VikhrModels. RuLLM Arena: Russian LLM Evaluation Benchmark // GitHub repository. – URL: `https://github.com/VikhrModels/ru_llm_arena` (accessed: 01.08.2025).

5. *Ping-Pong.* Gusev I. PingPong: A Benchmark for Role-Playing Language Models with User Emulation and Multi-Model Evaluation.

6. *OpenAI.* Introducing deep research // OpenAI URL: `https://openai.com/index/introducing-deep-research/` (accessed: 31.07.2025).

7. *Storm.* Shao Y. et al. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models //NAACL-HLT. - 2024

8. *ResearchArena.* Kang, Hao, and Chenyan Xiong. "ResearchArena: Benchmarking LLMs' Ability to Collect and Organize Information as Research Agents." arXiv e-prints (2024): arXiv-2406.

9. *Reranking.* Wang X. et al. Searching for Best Practices in Retrieval-Augmented Generation //CoRR. - 2024.

10. Touvron H. et al. LLaMA: Open and Efficient Foundation Language Models.

11. Wu, Jeff, et al. "Recursively summarizing books with human feedback." arXiv preprint arXiv:2109.10862 (2021).

12. *NDCG.* Järvelin, Kalervo, and Jaana Kekäläinen. "Cumulated gain-based evaluation of IR techniques." ACM Transactions on Information Systems (TOIS) 20.4 (2002): 422-446.

13. *R-Precision.* BUCKLEY C. Evaluating Evaluation Measure Stability //ACM SIGIR 2000 Proceedings. - 2000.

14. *BERTScore.* Zhang T. et al. BERTScore: Evaluating Text Generation with BERT //International Conference on Learning Representations.

15. *ROUGE.* Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." Text summarization branches out. 2004.

16. *BLEU.* Papineni K. et al. BLEU: a Method for Automatic Evaluation of Machine Translation.

17. *RuadaptQwen.* Tikhomirov, Mikhail, and Daniil Chernyshev. "Facilitating large language model russian adaptation with learned embedding propagation." Journal of Language and Education 10.4 (40) (2024): 130-145.

18. *DeepSeek V3.* Liu A. et al. DeepSeek-V3 Technical Report //CoRR. - 2024.

19. *Qwen3-235B.* Yang A. et al. Qwen3 technical report //arXiv preprint arXiv:2505.09388. - 2025.

20. T-Bank has opened access to its own Russian-language language model in the weight category of 7-8 billion parameters / T-Bank URL: `https://www.tbank.ru/about/news/20072024-t-bank-opened-access-its-own-russian-language-language-model-weight-category-of-7-8-billion-parameters/` (accessed: 10.05.2025).

21. YandexGPT 5 with reasoning mode // Yandex URL: https://ya.ru/ai/gpt (accessed: 30.07.2025).