DRAGOn: Designing RAG On Periodically Updated Corpus

Fedor Chernogorskii[2,1], **Sergei Averkiev**[1], **Liliya Kudraleeva**[3],
Zaven Martirosian[1,4,8], **Maria Tikhonova**[1,5],
Valentin Malykh[6,3,7], **Alena Fenogenova**[1,5]
[1]SberAI, [2]MBZUAI, [3]ITMO, [4]MISIS, [5]HSE University, [6]MWS AI, [7]IITU, [8]YSDA
Correspondence: fechernogor@gmail.com

Abstract

This paper introduces **DRAGOn**, method to design a RAG benchmark on a regularly updated corpus. It features recent reference datasets, a question generation framework, an automatic evaluation pipeline, and a public leaderboard. Specified reference datasets allow for uniform comparison of RAG systems, while newly generated dataset versions mitigate data leakage and ensure that all models are evaluated on unseen, comparable data. The pipeline for automatic question generation extracts the Knowledge Graph from the text corpus and produces multiple question-answer pairs utilizing modern LLM capabilities. A set of diverse LLM-as-Judge metrics is provided for a comprehensive model evaluation. We used Russian news outlets to form the datasets and demonstrate our methodology. We launch a public leaderboard to track the development of RAG systems and encourage community participation.

Figure 1: The DRAGOn logo.

1 Introduction

Retrieval-Augmented Generation (RAG) has become a powerful tool for enhancing the domain adaptation and factuality of large language models (LLMs) by incorporating external knowledge retrieved at inference time. This approach enables more up-to-date and grounded responses without the need for costly re-training. As RAG-based systems expand to applications such as open-domain QA, customer support, and enterprise search, their standardized evaluation remains a challenge. It may be unclear whether strong performance of a system is due to the quality of its retriever-generator pipeline or because the underlying LLM has been exposed to portions of the test data during training. It is possible that a static benchmark will become contaminated over time.

Several existing RAG evaluation frameworks (Es et al., 2024; Lyu et al., 2025) provide pipelines for automatic generation of question-answer pairs, typically assuming that users will build their own evaluation datasets. While this enables domain-specific benchmarking, it is labor-intensive and difficult to maintain. Furthermore, when the retrieval corpus is continuously updated during deployment, results may become non-reproducible, as the model may face a different knowledge distribution than the one used during its initial evaluation.

In this work, we introduce **DRAGOn: Designing RAG On Periodically Updated Corpus** a novel methodology that reflects realistic usage patterns by leveraging a regularly updated knowledge base. We also release a benchmark built on current news sources using this methodology, which can be readily adapted to other document domains, such as scientific papers or court decisions. To foster transparency and community engagement, we publicly release an evaluation framework which comprises the codebase for automatic question generation, evaluation scripts, and a dynamic leaderboard to track progress on RAG-based systems in Russian. Although the benchmark targets Russian, the framework is potentially extendable to other languages and multilingual scenarios, making it broadly ap-

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 622–638

plicable. **Our contributions are as follows:**

(i) We propose **DRAGOn**[1] the methodology to develop a RAG benchmark with a regularly updated knowledge base, designed to evaluate RAG systems in a dynamic setup; we develop a benchmark on Russian news corpora as a reference for the proposed methodology.

(ii) We release an open-source evaluation framework[2] comprising a reusable question generation pipeline and evaluation scripts, enabling reproducible experimentation and easy integration of new models and retrieval components. By design, it can potentially be adapted to other languages and multilingual settings, broadening its applicability beyond Russian.

(iii) We launch a regularly updated public leaderboard[3] for recurrent evaluation to support reproducible and community-driven research.

2 Related Work

Evaluating retrieval-augmented generation (RAG) systems poses unique challenges, as it requires datasets that jointly assess both the retrieval and generation components. Constructing such benchmarks is costly and time-consuming because it involves curating large collections of text–question–answer triplets. To alleviate this, several works have explored synthetic data generation to automate question and answer creation (Es et al., 2024; Lyu et al., 2025), often leveraging domain-specific pipelines or knowledge graphs for better control over content and difficulty.

Early RAG benchmarks such as KILT (Petroni et al., 2021) unified multiple English-language datasets over a fixed Wikipedia snapshot, emphasizing source attribution and retrieval grounding. More recent efforts have extended the evaluation to multi-turn conversational and reasoning-intensive scenarios, as seen in mtRAG (Katsis et al., 2025) and RAD-Bench (Kuo et al., 2025). The CRAG benchmark (Yang et al., 2024) further focuses on factual consistency, capturing five key aspects of RAG system behavior. Complementarily, RAGAS (Es et al., 2024) provides a reference-free evaluation framework measuring context relevance, faithfulness, and answer completeness, and offers an open-source API for reproducible benchmarking.

Dynamic and time-sensitive evaluation has emerged as another important dimension. Real-Time QA (Kasai and et al., 2023) introduced a benchmark for evaluating systems on continuously evolving information sources, reflecting real-world deployment settings. The news domain, with its frequent updates and temporal drift, has thus become a popular testbed for such studies (Tang and Yang, 2024; Chen et al., 2024).

Despite these advances, the field still lacks a universal, contamination-free, and continuously updated benchmarks (White et al., 2024). This gap hinders fair and reproducible comparison across RAG systems and motivates the development of dynamic, standardized evaluation resources.

To address this need, we present **DRAGOn** – a methodology to develop dynamic, regularly updated benchmarks for RAG systems based on real-world, shifting corpora.

3 Benchmark Design

Using our methodology, one can develop a benchmark designed to evaluate RAG systems in a dynamically evolving news domain. The benchmark's architecture prioritizes modularity, automation, and reproducibility while addressing the core challenges (Yu et al., 2024) in the RAG evaluation landscape, such as the temporal aspects of information, the vast and dynamic sources of knowledge, and the factuality and faithfulness in generation. The entire pipeline of the benchmark architecture is shown in Fig. 2. Below, each step is described in more detail.

Data Acquisition and Processing: We maintain a dedicated set of parsers which periodically crawl a selection of news sources recognized as popular news websites in Russia on a daily basis. The parsed content is synchronized with our storage. To avoid redundancy and ensure incremental updates, a scheduled automated job identifies differences with the previous dataset revision and extracts updated segments for downstream processing.

This design ensures that the benchmark reflects evolving real-world distributions and mitigates the risks of overfitting to static datasets. The pipeline further ensures that newly surfaced topics and entities from the news stream are constantly incorporated into the benchmark.

QA Dataset Formation: The process of creat-

[1]The video demonstration of the evaluation tool is available on YouTube.

[2]The framework is released under the MIT license: https://github.com/RussianNLP/DRAGON

[3]https://huggingface.co/spaces/ai-forever/rag-leaderboard

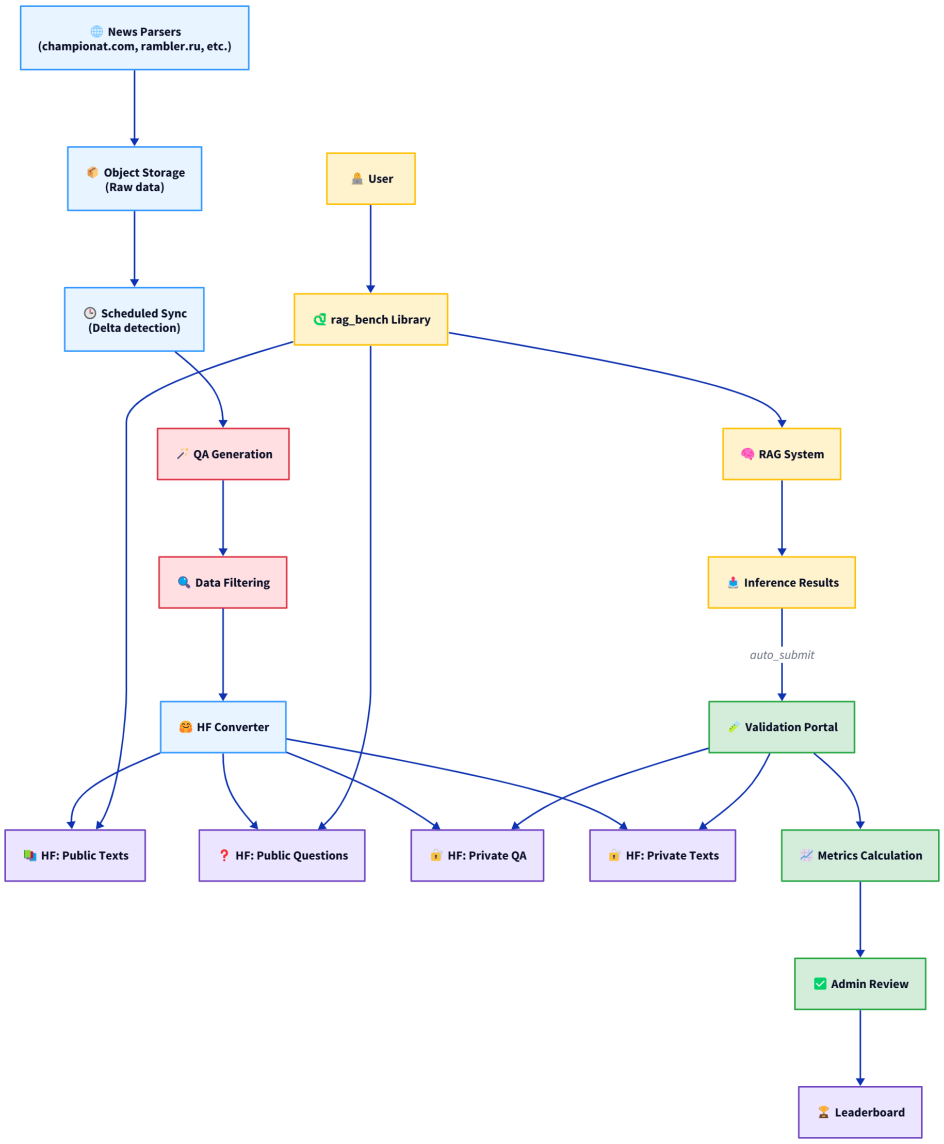


Figure 2: Architecture of the benchmark system based on DRAGOn. All datasets are versioned and uploaded to Hugging Face with incrementally updated revision numbers. This versioning mechanism ensures reproducibility and provides users with stable snapshots for further experimentation.

ing questions and answers based on the updated increment of the news data is described in detail in Sec. 4. The pipeline transforms the generated QA pairs into several HF datasets, which form the core of the benchmark:

- *Public Texts*: Contains cleaned source documents. Each item is assigned a `public_id` to enable matching without exposing the true internal IDs.
- *Public Questions*: Contains only questions, indexed via `public_id` to obfuscate alignment and encourage retrieval.
- *Private Texts Mapping*: Used only for evaluation purposes. It contains internal `ids` and the corresponding `public_ids` to enable accurate mapping during metric computation.
- *Private QA*: Provides canonical ground-truth answers for generative evaluation.

In addition to these main datasets, we provide a separate set of *Sandbox Datasets* with the exact

same structure as the main ones. All four sandbox datasets are fully public. Their purpose is twofold: (1) to transparently demonstrate the full structure and intended usage of the benchmark, and (2) to allow users to validate their RAG systems locally without submitting results to the validation portal.

These sandbox datasets can be evaluated using the `rag_bench` client library, which supports the same retrieval and generative metrics as those used by the official validation portal (except for judgment-based metrics). This enables convenient local experimentation, debugging, and reproducibility.

User Experience To facilitate seamless evaluation for users, we provide a PyPi-hosted Python library `rag_bench`, which offers an interface to: *Fetch* the latest version of the public datasets by dynamically resolving the latest Hugging Face revision; *Observe* the RAG system baseline, which can be adopted for the target one; *Evaluate* RAG system and package results for submission; *Submit* results via API to our evaluation portal; *Calculate* retrieval and generative metrics locally using the sandbox datasets.

User workflow includes loading public data, applying a custom RAG pipeline, and collecting results in the following form:

```
{
    "0": {
        "found_ids": [17, 69, 69, 22, ...],
        "model_answer": "Answer"
    },
    ...,
}
```

These results encode both the retrieved `public_ids` and the generated answers, decoupling the user's model output from any private evaluation artifacts. This separation allows for secure evaluation without exposing ground-truth data.

Validation Portal Submitted results then are sent to the *Validation Portal* — a Flask-based backend with a Single Page Application written in Vue as a frontend that performs secure evaluation using the private datasets. The portal evaluates submissions using private datasets and prepares evaluation results for admin approval before publishing. Importantly, users submit only their results — all ground-truth data remains internal.

Leaderboard and Auto-Evaluation A Hugging Face Gradio Space serves as the public Leaderboard. The results are committed in a version-controlled `results.json` file, automatically updated by the validation portal upon approval.

To reduce latency and improve benchmarking coverage, we support automatic evaluation for selected pre-approved baselines, which include several popular LLMs and retrieval embedding models. The results are computed via the same `rag_bench` client.

3.1 Versioning Strategy

Given the dynamic nature of the benchmark, versioning plays a critical role in ensuring meaningful comparisons. Each evaluation result is tied to a specific dataset revision. On the leaderboard, users can view results for a single dataset version or toggle an "Actual Versions" mode to aggregate results across recent revisions.

Dataset versioning is performed automatically based on the last available version on Hugging Face. The version number follows a semantic format, e.g., `1.10.0`. For each new release, the middle segment of the version is incremented, resulting in a new version such as `1.11.0`, which is then uploaded to Hugging Face. This approach ensures consistent, chronological dataset updates while preserving backward compatibility for previously published results.

Note that sandbox datasets are not updated on a regular basis. They serve as a static reference set for demonstration and local validation purposes.

4 Dataset Generation

The Data Generation pipeline (see Fig. 3) consists of 2 main stages preceded by preliminary data preprocessing: KG Extraction and Question Generation. KG Extraction retrieves factual information from texts and preserves the most specific and fresh facts in the form of a Knowledge Graph. The Question Generation module samples subgraphs of a certain structure to generate a question-answer pair with an LLM.

4.1 Knowledge Graph Extraction

To achieve fine-grained control over automatic question generation, we designed a Knowledge Graph (KG) Extraction module inspired by (Chepurova et al., 2024). This component transforms unstructured news texts into a structured set of factual triplets that later guide question creation.

We use LLaMa 3.3 70B Instruct[4] Grattafiori et al.

[4] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

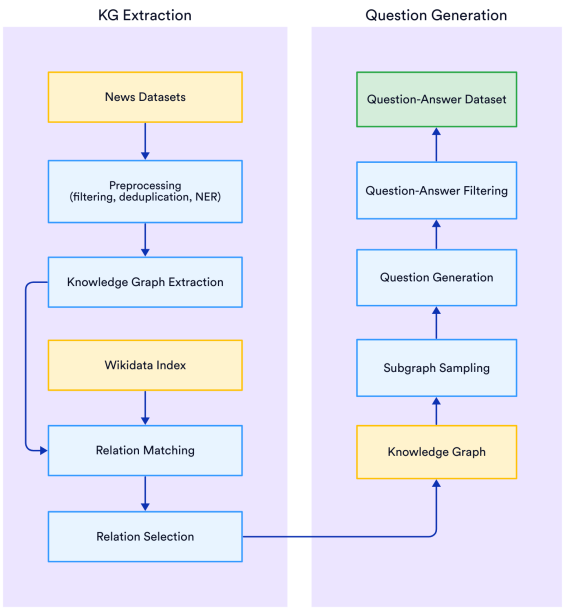


Figure 3: Architecture of the Data Generation pipeline. Before the start of the KG extraction, we perform data deduplication as the news dump could contain multiple edited versions of the same article. We preserve only the latest version of the text with the same URL. Also we extract named entities for further question filtering.

(2024) to extract candidate factual triplets from the corpus. Each triplet has the form (head entity – relation – tail entity), corresponding to the subject, predicate, and object in the original sentence.

The extracted entities are matched with the Russian subgraph of Wikidata (Vrandečić and Krötzsch, 2014). For every entity name identified in the text, we query the Wikidata API to find possible matches. The mapped entities are then vectorized using a sentence-embedding model and stored in a vector database. To handle ambiguity, we keep the five most similar candidates for each extracted entity according to vector similarity.

To ensure consistency across triplets, the same LLaMa 3.3 70B Instruct model is used again to normalize entity and relation names. Given the list of candidate matches from the previous step, the model selects the most appropriate canonical form while taking the full sentence context into account. This process merges spelling variants and aliases referring to the same entity, resulting in a cleaner, unified graph structure.

Our goal is to build a graph that captures new information appearing in the latest news updates. Therefore, we discard any triplets that exactly match existing facts in Wikidata. Triplets absent from the knowledge base are treated as novel facts, as they are more likely to represent fresh events, and thus serve as valuable material for generating time-sensitive questions.

4.2 Question Types

The question generation stage begins with subgraph extraction from the constructed Knowledge Graph. We identify all subgraphs matching one of four predefined structural templates, each representing a distinct question type (Yang et al., 2024):

Simple These type correspond the most simple questions based a single fact mentioned in one or several texts. They are based only on one relation from the graph: the predicate and one of the entities involved in the relation are used to compose the question, and the second entity becomes the answer.

- **relations**: *(Morty Smith | voice | Keisuke Chiba)*
- **question**: *Who voiced Morty Smith?*
- **answer**: *Keisuke Chiba*

Set Set questions test the RAG system's ability to align information from several texts. They are based on a one-to-many subgraphs in which the number of triplets share relation and either object or subject. The question is generated using shared entity and relation. The answer consists of all other entities in the subgraph.

- **relations**: *(Ryan Otter | composed music for | Method), (Ryan Otter | composed music for | Trigger)*
- **question**: *What projects has Ryan Otter composed music for?*
- **answer**: *Trigger, Method*

Multi-Hop Multi-hop questions evaluate the system's ability to reason in a multistage manner. The corresponding subgraph is a pair of triplets, intersecting at a single entity. The question is constructed similarly to a simple question; however, the repeated entity must not be mentioned in the question. It is used as a bridge-entity, which is described in question as a reference extracted from another triplet.

- **relations**: *(FAW | country of origin | China), (FAW | number of cars sold in 2023 | 2139)*
- **question**: *In which country is the company located that sold 2139 cars in 2023?*
- **answer**: *China*

Conditional Conditional questions are the extension of multi-hop questions with the same underlying subgraph of a pair of triplets, intersecting at a single entity. However, for a conditional question, both facts are used to form the question, while the repeated entity becomes an answer.

- **relations**: *(Roman Miroshnichenko | performed at | M-bar), (Roman Miroshnichenko | met with | Dmitry Dibrov)*
- **question**: *Who performed at M-bar and met with Dmitry Dibrov?*
- **answer**: *Roman Miroshnichenko*

Each selected subgraph is then passed to the language model, which generates a natural-language question–answer pair. The question is formulated as a fluent, contextually appropriate sentence, while the answer comprises one or more entities explicitly present in the subgraph.

4.3 QA Filtering

To ensure high-quality and contextually grounded question–answer pairs, we apply a multi-stage **filtering pipeline** combining linguistic validation, entity consistency, graph correspondence check, and LLM-based judgment.

1. Linguistic and Structural Filtering. Firstly, we assess the grammatical correctness and fluency of each question using a RuRoBERTa-large model trained on the RuCoLa dataset[5] (Mikhailov et al., 2022). This step eliminates ungrammatical or poorly formed questions. Next, we perform Named Entity Recognition (NER) on the original source text using the *Natasha* library. The generated questions and answers are checked for the presence of these entities. Samples without explicit named entities are discarded to remove trivial, knowledge-free examples. We further filter out overly simplistic questions by evaluating them with smaller instruction-tuned LLMs (Qwen 2.5 7B (Team, 2024) and LLaMa 3 8B (Grattafiori et al., 2024)) without context; if a model can answer a question directly from prior knowledge, it is excluded.

2. Graph Correspondence Filtering. Each remaining QA pair is verified against the **source subgraph** used during question generation to ensure factual alignment. For every entity in the subgraph, we calculate the Levenshtein distance (Levenshtein et al., 1966) between its label and the text of both the question and answer. Each node in the graph (entity) is assigned 2 coefficients: question presence and answer presence. It is evaluated as the scaled Levenshtein distance between the name of the entity and the closest substring from the question and answer. These values allow us to check that all entities have been mentioned correctly.

In the graphs for **Set** and **Conditional** question types, the positions of every entity are strictly determined. The algorithm averages the presence coefficients of entities implied to be in the same part of the output. If any of these values is lower than the threshold, it indicates incorrect generation. For **Simple** questions each entity can appear in both parts of the output, although the entity must be mentioned once in the question-answer pair. The presence coefficients were averaged over all entities from the subgraph, then 5% highest and lowest values were filtered out. **Multi-Hop** questions inherit the same process for nodes having only one connection in the subgraph. The bridge entity that has two connections should not be mentioned in the model output. A high value for any of the presence coefficients for this entity demonstrates a question-type violation.

3. LLM-as-Judge Evaluation. In the final stage, we apply the *LLM-as-Judge* approach using **POLLUX 7B** (Martynov et al., 2025), a model fine-tuned for fine-grained evaluation in Russian. Each QA pair is automatically rated along **eight generative criteria**: (1) *Question literacy* (grammar and style), (2) *Clarity*, (3) *Naturalness*, (4) *Context sufficiency* (answer can be found in the passage), (5) *Context necessity* (question depends on the passage), (6) *Answer correctness*, (7) *Answer uniqueness*, and (8) *Answer literacy*. More details on these criteria can be found in Appx. D.

Each criterion is transformed into a separate prompt with the specific scoring scale (0-2). For answer criteria (Literacy, Correctness, Uniqueness Based on Context), the prompt contains a news article, a question, and an answer; for other criteria, the answer is omitted. The example is classified as positive according to the particular criterion if the judge model assigns a rating of 1 or higher. This threshold was chosen to imitate the majority vote used in human evaluation.

To validate the reliability of using a language model as an automated evaluator for filtering gen-

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

Criterion	Precision	Recall
Question Literacy	0.96	0.99
Question Clarity	0.99	0.62
Question Naturalness	0.96	0.52
Context Sufficiency	0.94	0.71
Context Necessity	0.93	0.95
Answer Correctness	0.95	0.82
Answer Uniqueness based on context	0.85	0.78
Answer Literacy	0.91	0.97

Table 1: Comparison of the automatic metrics and manual evaluation results. The model achieves high **precision** but moderate **recall** relative to human evaluation. This trade-off is desirable in our setting, where maintaining dataset reliability is more important than exhaustive coverage. Retaining only high-confidence samples ensures that the resulting benchmark consists of the most coherent, contextually valid, and factually grounded question-answer pairs.

erated question-answer pairs, we conducted an empirical comparison against human judgments. A random sample of 532 examples was drawn from the generated dataset and independently assessed by a panel of human annotators (with more than three annotators per example) as well as by a large language model. An example was considered positive by human annotators if half or more of the assessors provided a positive assessment.

The comparison in Tab. 1 reveals that the language model achieves high Precision but moderate Recall relative to the human-labeled data. This trade-off is acceptable in our setting, as the dataset contains a large volume of generated examples. In this context, precision is more critical than recall: retaining only high-quality samples is preferable, even if some potentially acceptable data are discarded. This justifies the use of the language model as an effective filter for selecting the most reliable and contextually appropriate question-answer pairs at scale.

After all filtering stages, **150 high-quality questions per category** are retained for the final benchmark dataset.

5 Experimental Setup

To construct our experimental RAG systems, we used the LangChain framework[6]. All texts from the *Public Texts* dataset are split into chunks of length 500 with an overlap of 100 characters. Each chunk is vectorized using the retrieval model of the evaluating RAG system with the corresponding document prefixes, and the resulting vectors are stored in a vector database.

During the search phase, we use the prompted retrieval model to find five of the most relevant texts that match the user's query. Retrieved chunks are incorporated into a prompt provided to the LLM of the evaluated RAG system. If the total length of the filled-in prompt exceeds the model's maximum context length, the contextual information is truncated to the required size. To accelerate LLM inference, we utilize the vLLM framework[7] (Kwon et al., 2023).

5.1 Experimental Setup Details

Embedding Model Prefixes To vectorize questions and documents, we used embedders with the corresponding prefixes. These prefixes are shown in Tab. 2.

LLM Prompt Template To generate answers for the questions, we used the following template for the user message prompt:

```
``Answer the question using the provided context.
Give me only an answer.
<context> {context} </context>
Question: {question}
Answer: ''
```

Model Configuration For serving models, we used the vLLM framework. The model parameters used are shown in Tab. 3. We set `max_new_tokens` to 1000 for all models to limit the response length of the models.

Metrics The performance of retrieval is measured by 3 metrics:

- **Hit Rate** measures the proportion of queries for which the relevant document appears among the top-k retrieved results.
- **Mean Reciprocal Rank (MRR)** evaluates ranking quality by measuring how highly the first relevant document is ranked, assigning higher scores when a relevant document appears earlier in the ranked list.

We evaluate End-to-end RAG systems with:

- **ROUGE-2** measures bigram overlap between the model output and the reference text, capturing local phrase-level similarity and rewarding matching adjacent word pairs.

[6] https://pypi.org/project/langchain/

[7] https://github.com/vllm-project/vllm

Model	Query prefix	Text prefix
FRIDA	search_query:	search_document:
E5 Mistral$_{\text{7b Instruct}}$	Instruct: Given a web search query, retrieve relevant passages that answer the query. Query:	X
Qwen 3$_{\text{Embedding 8b}}$	Instruct: Given a web search query, retrieve relevant passages that answer the query. Query:	X
mE5$_{\text{Large Instruct}}$	Instruct: Given a web search query, retrieve relevant passages that answer the query. Query:	X

Table 2: Embedder configurations: query and text prefixes

Model	TP	ML
Qwen 2.5$_{\text{32b Instruct}}$	4	32768
Qwen 2.5$_{\text{7b Instruct}}$	1	32768
Ruadapt Qwen$_{\text{32b Instruct}}$	4	32768
Qwen 3$_{\text{32B}}$	4	32768
Gemma 3$_{\text{12b it}}$	1	131072
Gemma 3$_{\text{27b it}}$	4	131072

Table 3: Model configurations. **TP** stands for tensor parallel size, **ML** for maximal context length.

- **ROUGE-L**, measures the longest common subsequence between the model output and the reference text, capturing overlap in overall sentence structure.
- **The Judge Score** is used to evaluate the overall answer quality, is calculated as the average of the automatic scores from Pollux [8] across multiple criteria (e.g., correctness, completeness, and relevance).

6 Experiments

Question Quality Evaluation To assess the quality of the generated question-answer pairs, a human evaluation study is conducted. Each QA pair from *Sandbox Datasets* (Sec. 3) is independently evaluated by 3 expert annotators along the evaluation criteria from Sec. D. Annotators were asked to mark each pair as "Good" or "Not Good" with respect to each criterion. To account for potential subjectivity in judgment, we considered a QA pair to be acceptable with the majority vote. Evaluation results are provided in Tab. 4.

Retrieval Evaluation Retrieval evaluation results presented in Tab. 5 demonstrate consistently strong performance across all evaluated retriever models. Among them, Qwen3$_{\text{Embedding 8B}}$ and E5 Mistral$_{\text{7b Instruct}}$ achieve the highest scores.

[8] https://ai-forever.github.io/POLLUX/

Criterion	Apr	May	Jun
Question Literacy	0.96	0.97	0.99
Clarity	0.99	1.00	1.00
Naturalness	0.98	0.96	0.97
Context Sufficiency	0.98	0.98	0.99
Context Necessity	0.95	0.97	0.98
Correctness	0.95	0.92	0.96
Uniqueness	0.76	0.78	0.80
Answer Literacy	0.79	0.71	0.75

Table 4: The proportion of QA pairs considered good for each dataset version and each evaluation criterion. The results establish the high quality of generated questions and significant context dependency. The answer evaluation proved the prevalence of correct answers, while the answer uniqueness is lower, so the ground truth answer can be substituted with another entity from the text. This fact exhibits the importance of LLM-as-Judge evaluation for RAG systems to avoid rephrasing influence.

mE5$_{\text{Large Instruct}}$ also performs competitively. As for FRIDA, it also demonstrates strong performance but its results are slightly inferior to those of the competitors.

Table 5: Retrieval evaluation results. The best score is in bold, second best is underlined.

Retriever	Hit Rate	MRR
FRIDA	0.932	0.822
Qwen 3$_{\text{Embedding 8b}}$	**0.960**	**0.867**
E5 Mistral$_{\text{7b Instruct}}$	<u>0.956</u>	<u>0.851</u>
mE5$_{\text{Large Instruct}}$	0.949	0.834

End-to-End System Evaluation The results are provided in Table 6. Overall, the results show that classic metrics such as Rouge-L are not objective enough and do not allow evaluating all aspects of the RAG task.

First, it can be seen that the choice of the re-

LLM	Rouge2	RougeL	JS
Retrieval: **FRIDA**			
Gemma 3 12B it	0.14	0.22	0.63
Gemma 3 27B it	0.14	0.22	0.64
Qwen 2.5 32B	0.09	0.16	0.62
Qwen 2.5 7B	0.08	0.13	0.57
Qwen3 32B	0.08	0.14	0.64
Rudadapt Qwen 32B	0.13	0.22	0.72
Retrieval: **mE5**$_{\text{Large Instruct}}$			
Gemma 3 12B it	0.15	0.24	0.67
Gemma 3 27B it	0.15	0.24	0.67
Qwen 2.5 32B	0.10	0.18	0.66
Qwen 2.5 7B	0.09	0.15	0.63
Qwen3 32B	0.11	0.18	0.69
Rudadapt Qwen 32B	0.14	0.21	0.74
Retrieval: **Qwen 3**$_{\text{Embedding 8b}}$			
Gemma 3 12B it	0.16	**0.26**	0.71
Gemma 3 27B it	**0.17**	**0.26**	0.72
Qwen 2.5 32B	0.11	0.19	0.68
Qwen 2.5 7B	0.09	0.16	0.64
Qwen3 32B	0.11	0.19	0.71
Rudadapt Qwen 32B	0.16	0.25	**0.82**
Retrieval: **E5 Mistral**$_{\text{7b Instruct}}$			
Gemma 3 12B it	0.16	0.25	0.68
Gemma 3 27B it	0.16	0.25	0.70
Qwen 2.5 32B	0.12	0.19	0.68
Qwen 2.5 7B	0.09	0.16	0.64
Qwen3 32B	0.11	0.19	0.71
Rudadapt Qwen 32B	0.16	0.25	0.79

Table 6: End-to-end RAG-system evaluation results. Retrieval evaluation results. The judge's score (JS) is computed by averaging the results among the criteria. The best score is in bold, and the second-best score is underlined.

trieval model plays a crucial role. Qwen3$_{\text{Embedding 8B}}$ and E5 Mistral$_{\text{7b Instruct}}$ show the strongest results. Second, it should be noted that the general LLM ranking remains the same with every retrieval, with Rudadapt Qwen 32B heading the list by Judge Score, and Gemma 3$_{\text{12b it}}$ outperforming other competitors by Rouge metrics.

In general, system scores positively characterize DRAGON as being complex enough for modern RAG-systems, allowing researchers to evaluate their capabilities at a high level. In the future, we also plan to complexify Judge Evaluation criteria, thus providing an opportunity for an adequate assessment of more advanced models than those that exist nowadays and avoiding the danger of the benchmark being solved.

7 Conclusion

We presented **DRAGOn**, a method to design RAG benchmark on any periodically updated document source, with it we created the dynamic benchmark for evaluating retrieval-augmented generation systems in Russian. DRAGOn is designed for real-world deployment settings by leveraging a regularly updated knowledge base and focusing on the recurrent evaluation of both retriever and generator components. Our methodology addresses the current lack of standardized RAG evaluation tools; thus, we created a sample benchmark for the Russian language. We release the benchmark, which comprises a question generation pipeline and evaluation scripts, and launch a public leaderboard to support reproducible, transparent, and community-driven research. In the future, with the evolving capabilities of RAG systems, we plan to extend the benchmark by introducing new question types, refining the LLM-as-Judge criteria. In addition, we aim to open-source previous snapshots of the evolving datasets to support reproducibility and foster further community research.

We hope DRAGOn will serve as a foundation for future work on multilingual and dynamic RAG systems.

Limitations

While the proposed benchmark provides a valuable framework for evaluating retrieval-augmented generation (RAG) systems, several limitations should be acknowledged:

Source Diversity The benchmark primarily relies on the available documents from a specific domain (news), which may not fully capture the diversity of real-world information retrieval and generation tasks. Expanding the dataset range could enhance the benchmark's applicability across different domains.

Language Diversity The proposed benchmark consists entirely of Russian language documents and questions. Although the methodology itself could be easily applied to any other language, in its current state, only one language is presented.

Evaluation Metrics The chosen evaluation metrics, such as ROUGE, which is essentially an n-gram precision, predominantly focus on surface-level matching. These metrics may not adequately reflect the semantic and pragmatic aspects of the generated content and have limited correlation with human judgment (Deutsch et al., 2022). The LLM as Judge evaluation is designed to mitigate the semantic gap of n-gram-based metrics. However, the RAG benchmark requires specific criteria to capture details of system performance. Building more adapted judge models can improve the quality of the assessment.

Domain-Specific Challenges RAG systems might perform differently across various domains due to domain-specific complexities and knowledge structures. The benchmark does not currently address these nuances, which could hinder its ability to generalize across distinct fields like medicine, law, or general knowledge.

Retriever-Generator Synergy The interactions between retrieval and generation components are complex and dynamic. Our benchmark does not deeply explore how different configurations and synergistic interactions affect performance, possibly oversimplifying nuances that can significantly impact results.

Human Evaluation The benchmark primarily relies on automated metrics, which may not align perfectly with human judgments of quality and relevance. While we acknowledge the role of human evaluation, it was not feasible to incorporate it extensively into this iteration of the benchmark.

Scalability and Efficiency The computational resources required for comprehensive testing can be substantial, potentially restricting the accessibility of the benchmark to groups with extensive computational infrastructure.

Rapid Technological Advancements The field of RAG systems is rapidly evolving, with new models and techniques emerging frequently. The benchmark may quickly become outdated unless regularly updated to incorporate recent advancements and methodologies.

Addressing these limitations in future work could involve developing more comprehensive, diverse datasets, incorporating a broader range of evaluation metrics, and continuously adapting the benchmark to reflect the state-of-the-art in RAG systems. Additionally, exploring detailed interactions between retrieval and generation components and integrating more human evaluation into the assessment process could provide deeper insights and improve the robustness of the benchmark.

Ethical consideration

In developing and utilizing the retrieval-augmented generation (RAG) systems benchmark, several ethical considerations have been taken into account to ensure responsible and fair use of the technology:

Bias and Fairness Given that RAG systems are influenced by the data they are trained and tested on, it's crucial to address the potential for bias in retrieval and generation processes. Our benchmark highlights these concerns by incorporating evaluation metrics that identify and measure biases in model outputs. Future iterations aim to include datasets specifically designed to stress-test and mitigate bias.

Data Privacy The use of real-world datasets in RAG systems poses privacy risks, particularly concerning personally identifiable information (PII). We ensure that datasets included in the benchmark are sourced following strict privacy regulations and guidelines, and we encourage the anonymization of any PII to safeguard user privacy.

Content Quality and Misinformation RAG systems can potentially generate or propagate misinformation if not properly managed. Our benchmark assesses models on their ability to produce accurate and reliable content, and we emphasize the importance of retrieval sources that are reputable and verifiable to minimize risks associated with misinformation.

Transparency and Explainability Understanding the decision-making process of RAG systems is critical for trust and accountability. The benchmark encourages the development of models that offer insights into their retrieval and generation processes, promoting transparency and explainability.

Unintended Consequences The application of RAG systems can have unintended societal impacts, such as fostering dependency on AI for decision-making or influencing cultural narratives. Researchers and developers are encouraged to consider these broader implications and involve interdisciplinary perspectives in assessing the impact of their systems.

Access and Inequality High computational demands of RAG systems can exacerbate the divide between well-resourced organizations and smaller entities or individuals. Our benchmark advocates for the creation of more efficient models that democratize access and enable wider participation in developing and utilizing RAG technology.

Responsible Usage Educating users and stakeholders about the capabilities and limitations of RAG systems is vital to prevent misuse. Our research promotes guidelines and best practices to ensure that these technologies are used responsibly and ethically.

By acknowledging and addressing these ethical considerations, our aim is to contribute positively to the development and deployment of retrieval-augmented generation systems, ensuring they serve society in a beneficial and responsible manner. Future work will continue to refine these frameworks to address emerging ethical challenges as the field evolves.

Error Analysis A further limitation of our current benchmark release is the lack of a systematic error analysis of model failures. While we report aggregate retrieval and generation scores, we do not yet provide a fine-grained breakdown of common failure modes (e.g., retrieval misses vs. ranking issues, incomplete evidence aggregation in multi-hop questions, hallucinations under partially relevant context). Such analysis is important both for interpreting leaderboard progress and for understanding whether improvements come from better retrieval, better grounding/faithfulness, or exploiting dataset artifacts. In future work, we plan to add structured error taxonomies and identify systematic weaknesses of evaluated RAG pipelines.

AI-assistants Help We improve and proofread the text of this article using Writefull assistant integrated in Overleaf (Writefull's/Open AI GPT models) and GPT-4o[9], Grammarly[10] to correct grammatical, spelling, and style errors and paraphrase sentences. We underline that these tools are used strictly to enhance the quality of English writing, in full compliance with the ACL policies on responsible use of AI writing assistance. Nevertheless, some segments of our publication can be potentially detected as AI-generated, AI-edited, or human-AI-generated.

[9] https://chatgpt.com

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

Acknowledgments

We would like to express our deep appreciation to Ivan Bondarenko for his contribution to our pipeline and generous support of this work. This research partially done by A.F. is an output of a research project implemented as part of the Basic Research Program at the National Research University Higher School of Economics (HSE University).

References

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Alla Chepurova, Yurii Kuratov, Aydar Bulatov, and Mikhail Burtsev. 2024. Prompt me one more time: A two-step knowledge extraction pipeline with ontology-based verification. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 61–77.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022. Re-examining system-level correlations of automatic summarization evaluation metrics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6038–6052.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, and 1 others. 2025. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Patrick Es, Menno van Zaanen, Rob Koeling, and Mark Stevenson. 2024. Ragas: An evaluation framework for retrieval-augmented generation. In *Proceedings of the 2024 Conference of the European Chapter of the Association for Computational Linguistics (EACL): System Demonstrations*, pages 157–166.

Alena Fenogenova, Artem Chervyakov, Nikita Martynov, Anastasia Kozlova, Maria Tikhonova, Albina Akhmetgareeva, Anton Emelyanov, Denis Shevelev, Pavel Lebedev, Leonid Sinev, Ulyana Isaeva, Katerina Kolomeytseva, Daniil Moskovskiy, Elizaveta Goncharova, Nikita Savushkin, Polina Mikhailova, Anastasia Minaeva, Denis Dimitrov, Alexander Panchenko, and Sergey Markov. 2024. MERA: A comprehensive LLM evaluation in Russian. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9920–9948, Bangkok, Thailand. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten,

Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jungo Kasai and et al. 2023. Realtime qa: What's the answer right now? In *Proceedings of the 2023 Conference on Neural Information Processing Systems (NeurIPS)*.

Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. 2025. Mtrag: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *arXiv preprint arXiv:2501.03468*.

Tzu-Lin Kuo, Feng-Ting Liao, Mu-Wei Hsieh, Fu-Chieh Chang, Po-Chun Hsu, and Da-Shan Shiu. 2025. Rad-bench: Evaluating large language models' capabilities in retrieval augmented dialogues. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Industry Track*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Vladimir I Levenshtein and 1 others. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Transactions on Information Systems*, 43(2):1–32.

Nikita Martynov, Anastasia Mordasheva, Dmitriy Gorbetskiy, Danil Astafurov, Ulyana Isaeva, Elina Basyrova, Sergey Skachkov, Victoria Berestova, Nikolay Ivanov, Valeriia Zanina, and 1 others. 2025. Eye of judgement: Dissecting the evaluation of russian-speaking llms with pollux. *arXiv preprint arXiv:2505.24616*.

Vladislav Mikhailov, Tatiana Shamardina, Max Ryabinin, Alena Pestova, Ivan Smurov, and Ekaterina Artemova. 2022. RuCoLA: Russian corpus of linguistic acceptability. pages 5207–5227.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Tim Rocktäschel, and Sebastian Riedel. 2021. Kilt: A benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544. Association for Computational Linguistics.

Artem Snegirev, Maria Tikhonova, Anna Maksimova, Alena Fenogenova, and Alexander Abramov. 2024. The russian-focused embedders' exploration: rumteb benchmark and russian embedding model design. *arXiv preprint arXiv:2408.12503*.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Multihop question answering with retrieval-augmented generation. *arXiv preprint arXiv:2401.15391*.

Gemma Team. 2025a. Gemma 3.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025b. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Mikhail Tikhomirov and Daniil Chernyshev. 2024. Facilitating large language model russian adaptation with learned embedding propagation. *arXiv preprint arXiv:2412.21140*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

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

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

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, and 1 others. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu Jiang, and 1 others. 2024. Crag-comprehensive rag benchmark. *Advances in Neural Information Processing Systems*, 37:10470–10490.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pages 102–120. Springer.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A News Data Sources

For dataset formation, we rely on content from several well-established Russian news websites[11]:

- blog.okko.tv,
- daily.afisha.ru,
- lenta.ru,
- letidor.ru,
- moslenta.ru,
- motor.ru,
- quto.ru,
- tass.ru,
- gazeta.ru,
- ria.ru,
- rg.ru.

B Leaderboard Overview

Fig. 4 shows an overview of the leaderboard.

C Baseline Details

We evaluate open-source LLMs within 70B size[12] which score best on the MERA benchmark[13] (Fenogenova et al., 2024) (see Tab. 7 for their description) and several popular embedding models which show strong results on the retrieval task on ruMTEB or Multilingual MTEB[14] (Snegirev et al., 2024; Enevoldsen et al., 2025).

D Question-Answer Evaluation Criteria

This section describes the question-answer evaluation criteria used on the final question filtering stage. These criteria were developed to assess the general quality and naturalness of the question, its context dependence, and the correctness of the answer. The same set of criteria is used for manual annotation.

Question Literacy *Does the question exhibit correct grammar, spelling, and punctuation?* This criterion assesses the linguistic quality of the question. A well-formed question should be free of typographical errors, contain appropriate punctuation, and follow standard grammatical rules. Additionally, the phrasing should align grammatically with the surrounding context, ensuring the question does not feel syntactically out of place.

Question Clarity *Is the intent of the question clear and unambiguous?* This criterion evaluates how easily a reader can understand what information is being requested. The question should be interpretable either based on the provided context or general knowledge, without requiring additional clarification. Vague, overly broad, or logically inconsistent questions should be penalized.

Question Naturalness *Does the question sound like it could have been written by a human?* This assesses whether the question appears natural and contextually appropriate. It should avoid signs of being artificially generated such as unnatural phrasing, rigid templates, or repetitive structures. A natural question should feel relevant and plausible within the discourse of the text.

Context Sufficiency *Can the answer to this question be found entirely within the provided context?* This criterion determines whether the context passage contains enough information to answer the question. A question should not require external knowledge or assumptions unless that knowledge is very general or trivial. Questions with answers that are clearly present and verifiable in the text should receive high marks.

Context Necessity *Is the provided context necessary to answer the question?* This evaluates whether the question meaningfully engages with the context. Ideal questions should be context-dependent, meaning they cannot be accurately answered without access to the specific passage. Generic or overly broad questions that could be answered independently of the text (without specialized knowledge) are discouraged.

Answer Literacy *Is the answer written in a grammatically correct and readable manner?* This criterion checks for the overall linguistic quality of the answer. It should be free from spelling mistakes, awkward constructions, or inconsistent grammatical structure.

Answer Correctness *Is the answer factually correct and appropriate for the given question?* This criterion gauges the accuracy of the generated answer. It should contain all possible entities that can be mentioned in the answer without omitting any necessary details.

[11] All data is used in full compliance with legal requirements and ethical standards, under a formal agreement with Rambler. The collection process ensures respectful use of content without infringing on the rights of publishers or individuals.

[12] The size limit is introduced to ensure the feasibility of multi-model evaluation under the compute budgets.

[13] https://mera.a-ai.ru/en/text/leaderboard, valid for July 1, 2025.

[14] https://huggingface.co/spaces/mteb/leaderboard valid for July 1, 2025.

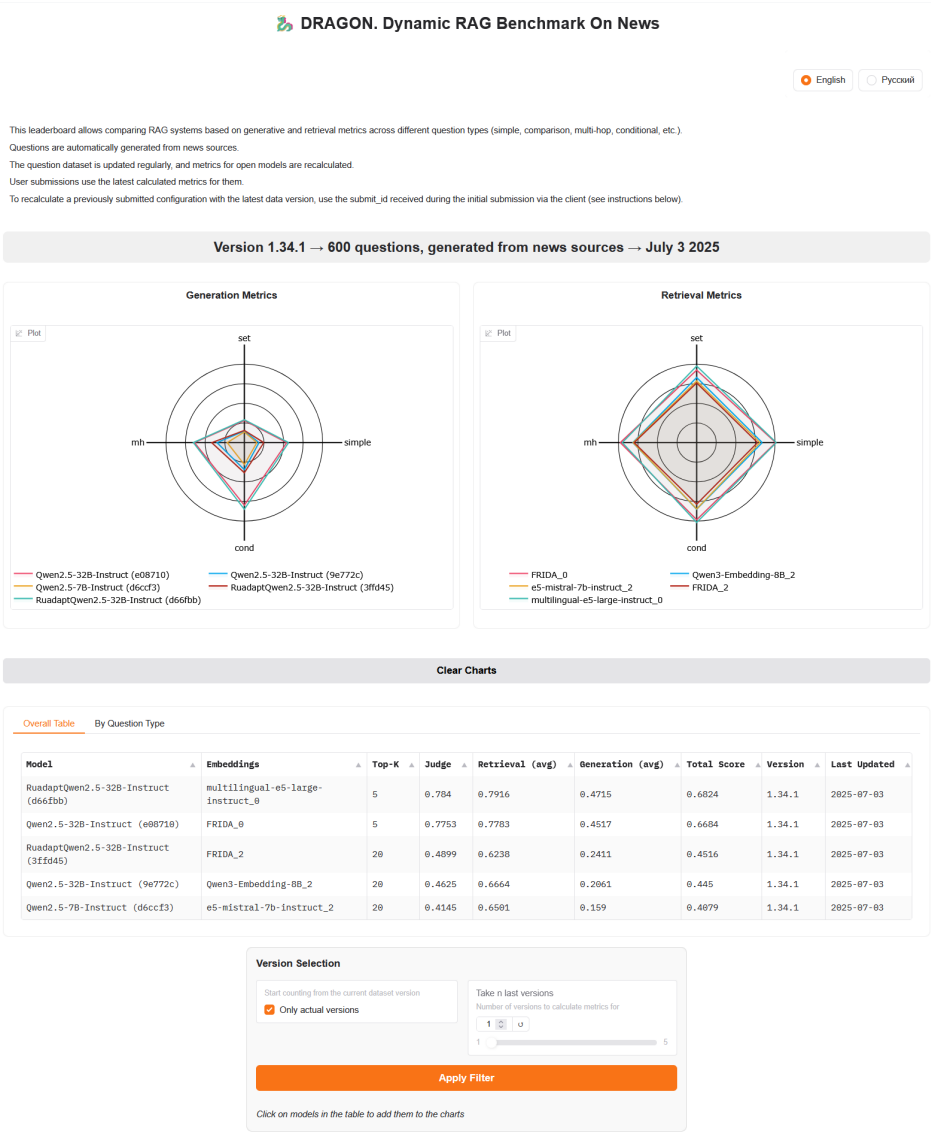


Figure 4: Leaderboard interface.

Answer Uniqueness Based on Context *Is this the only plausible answer that can be given based on the text?* This checks whether the answer is uniquely determined by the information in the context. If the passage contains multiple plausible answers or if ambiguity remains, this checkbox should not be selected. Ideal answers should be both correct and exclusive given the text.

E Human Evaluation Interface

A screenshot of a system used for human evaluation is presented in Fig. 6.

F LLM-as-Judge RAG Evaluation Criteria

This section provides a detailed description of the LLM-as-Judge criteria used to evaluate RAG systems. To build a comprehensive and interpretable set of metrics, Evaluation Targets provided by Yu et al. (2024) are utilized. For each Evaluation Target, we select several criteria from the POLLUX set of criteria:

- **Answer Relevance.** Measures the alignment between the generated response and the content of the initial query.
  - *Absence of unnecessary details. (Fluff)* The LLM's output is relevant and do not contain fluff.
- **Faithfulness.** Estimates the quality of the information extraction from retrieved documents.
  - *Consistency with real-world facts.* The

Model	Size	Hugging Face Hub link	Citation
Qwen $2.5_{\text{7b Instruct}}$	32B	Qwen/Qwen2.5-32B-Instruct	(Team, 2024)
Qwen $2.5_{\text{32b Instruct}}$	32B	Qwen/Qwen2.5-32B-Instruct	(Team, 2024)
Ruadapt $\text{Qwen}_{\text{32b Instruct}}$	32B	msu-rcc-lair/RuadaptQwen-32B-instruct	(Tikhomirov and Chernyshev, 2024)
Qwen 3_{32B}	32B	Qwen/Qwen3-32B	(Team, 2025b)
Gemma $3_{\text{12b it}}$	12B	google/gemma-3-12b-it	(Team, 2025a)
Gemma $3_{\text{27b it}}$	27B	google/gemma-3-27b-it	(Team, 2025a)

Table 7: The evaluated model description. Instruct models are marked with the corresponding suffix.

Model	Size	Hugging Face Hub link	Citation
FRIDA	823M	ai-forever/FRIDA	–
Qwen $3_{\text{Embedding 8b}}$	8B	Qwen/Qwen3-Embedding-8B	(Zhang et al., 2025)
E5 $\text{Mistral}_{\text{7b Instruct}}$	7B	intfloat/e5-mistral-7b-instruct	(Wang et al., 2023)
$\text{mE5}_{\text{Large Instruct}}$	560M	intfloat/multilingual-e5-large-instruct	(Wang et al., 2024)

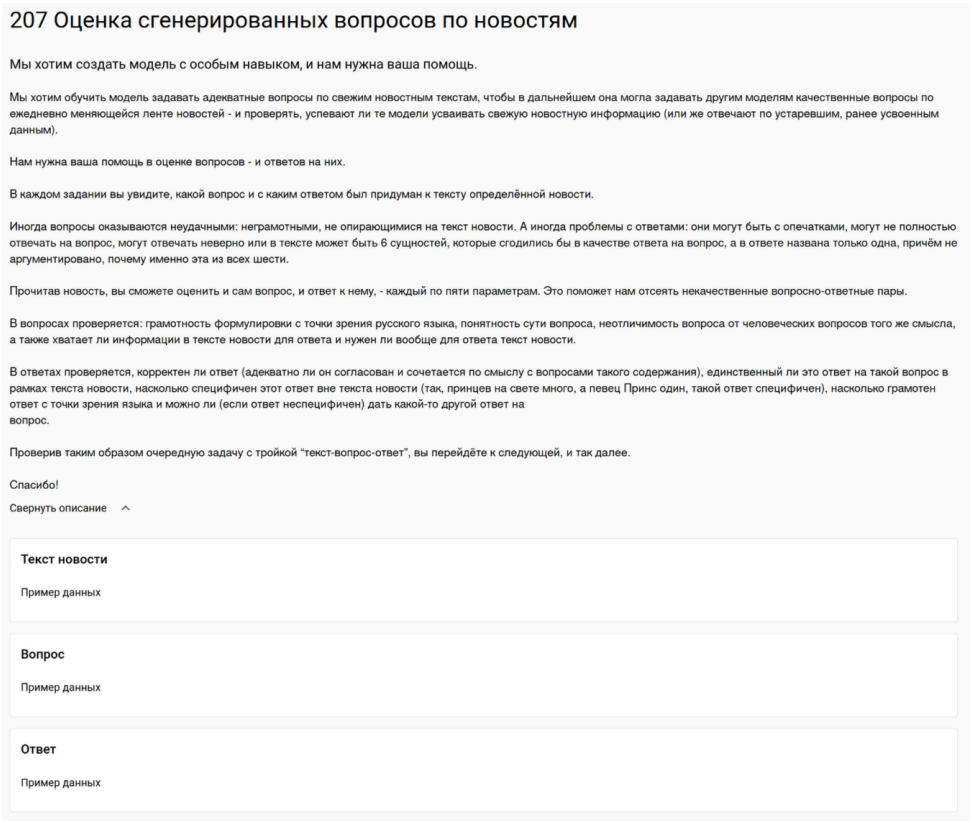
Table 8: The evaluated retriever description. Instruct models are marked with the corresponding suffix.

LLM's output does not contain factual errors.

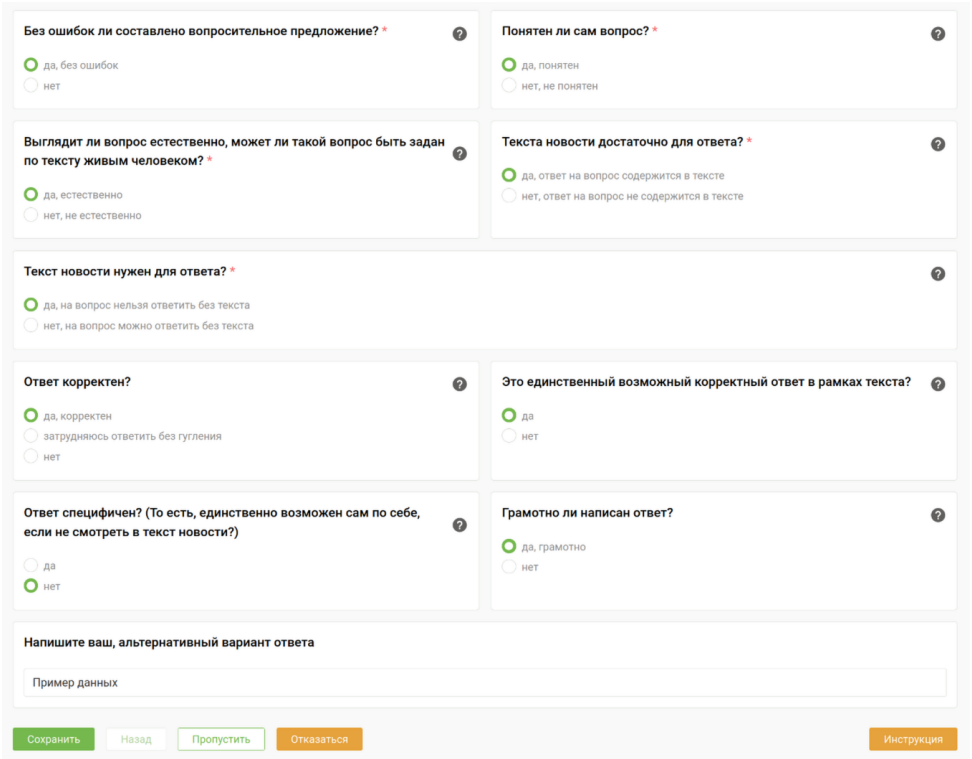
  - *Correctness of results.* The LLM extracted correct information from the text.

- **Correctness** Measures the accuracy of the generated response by comparing it to the ground truth response.
  - *Completeness.* The answer is complete and reaches the goal.
  - *Factual accuracy.* The LLM correctly reproduced the necessary facts and their related context.
  - *Preserving the main idea and details of the original.* The LLM preserves details and main idea.

The fine-grained set of metrics allows for comparing the RAG systems more precisely and improves interpretability. Fig. 6 provides a comparison of different RAG systems built on the basis of different variants of the Qwen 2.5 model combined with FRIDA and Qwen 3 Embedding 8B retrieval models. The results clearly demonstrate that larger language models yield higher-quality responses across all criteria, while the Absence of unnecessary details criterion results are similar for all combinations. Additionally, systems using Qwen3 8B embeddings consistently outperform those using FRIDA, highlighting the critical role of retrieval quality in end-to-end RAG performance. These findings emphasize that both the generative and retrieval components contribute significantly to final system effectiveness.



(a) Interface part 1



(b) Interface part 2

Figure 5: Human evaluation system interface.

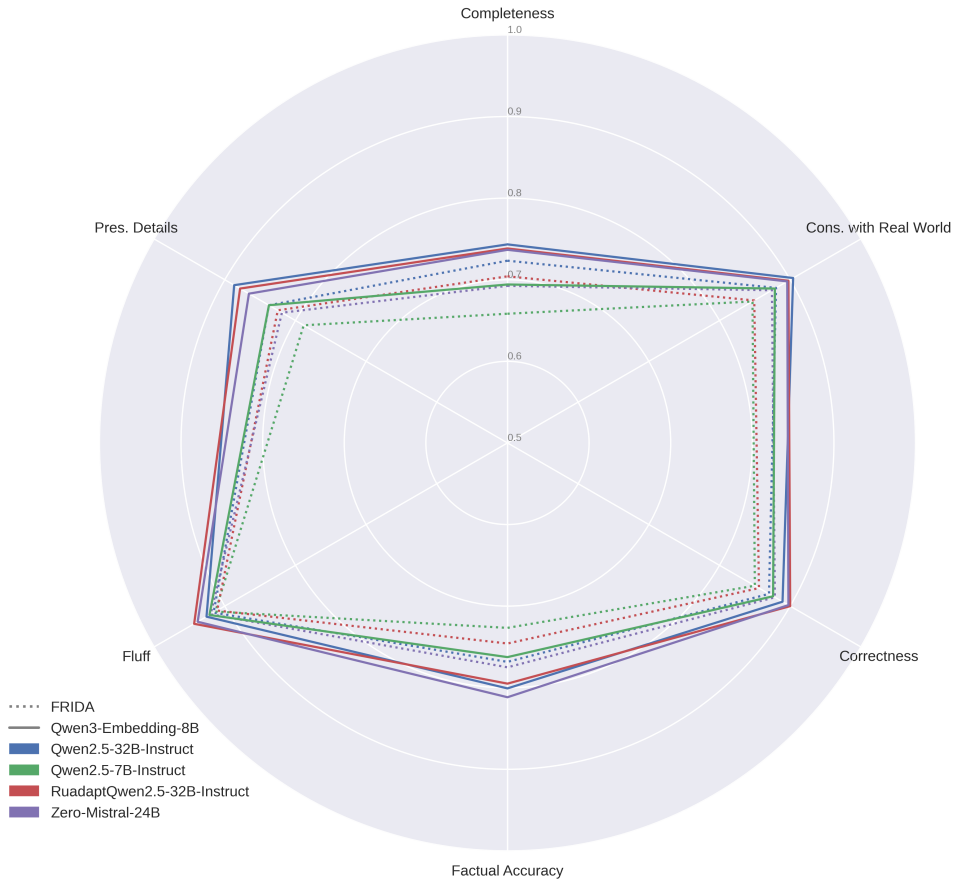


Figure 6: Detailed analysis of the RAG system performance from Tab. 6 along the separate criteria.