

RUWIKIBENCH: ОЦЕНКА БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ПОСРЕДСТВОМ ВОСПРОИЗВЕДЕНИЯ ЭНЦИКЛОПЕДИЧЕСКИХ СТАТЕЙ

© 2025 г. Д. А. Григорьев^{1,*}, Д. И. Чернышев^{1,**}

Представлено ...

В связи с растущим интересом к использованию больших языковых моделей (LLM) в качестве инструментов для генерации научных текстов, оценка их способностей к созданию энциклопедического контента становится все более актуальной. Однако для русскоязычных материалов этот вопрос изучен недостаточно, а существующие бенчмарки не охватывают ключевых аспектов аналитической работы с источниками. В данной работе представлен RuWikiBench - открытый бенчмарк на основе «Рувик» для оценки способностей больших языковых моделей воспроизводить статьи в стиле Википедии, основанный на трех задачах: отбор релевантных источников, построение структуры статьи и генерации секций. Результаты тестирования популярных открытых LLM показывают, что даже в идеальных условиях лучшие модели не всегда следуют экспертной логике составления энциклопедических материалов.

Ключевые слова и фразы: бенчмарк, Википедия, Рувик, large language model

ВВЕДЕНИЕ

Современные большие языковые модели демонстрируют впечатляющие результаты в генерации текстов различной стилистики и тематики. Однако их способности к работе с научными и энциклопедическими материалами остаются малоизученными, особенно для русскоязычных текстов. Существующие методы оценки способностей моделей преимущественно фокусируются на стандартных лингвистических задачах, не уделяя достаточного внимания аналитическим способностям при работе с научными текстами. Для русского языка эта проблема особенно актуальна из-за ограниченной доступности специализированных оценочных инструментов.

Существует множество бенчмарков, охватывающих различные лингвистические задачи для русского языка. RussianSuperGlue [1] оценивает общее языковое понимание и базовые задачи по обработке естественного языка. MERA [2] обеспечивает единые условия тестирования моделей за счет составления инструкций к генерации для каждой задачи, однако сами задачи ориентированы на проверку общего понимания. LIBRA [3] фокусируется на проверке способности модели к удержанию и извлечению информации из большого контекста, но сосредоточен на коротких ответах, не требующих глубоких рассуждений. Ru Arena General [4] фокусируется на парном сравнении моделей, но не на общем качестве ответа. Ping-Pong [5] оценивает диалоговые способности моделей, что важно для интерактивных систем, но не подходит для оценки способности проводить исследования и писать связные научно-энциклопедические тексты. При этом остается неохваченным целый класс задач, связанных с глубоким анализом текстов: создание развернутых, структурированных и фактологически точных текстов, подкрепленных большим количеством источников.

Недавнее развитие новых способностей агентов, например появление функции «Deep Research» у OpenAI [7] или разработка универсального алгоритма Storm [8], свидетельствует о возрастающем интересе к проведению научных исследований с помощью больших языковых моделей, что говорит о необходимости в создании новых подходов к объективной оценке аналитических способностей моделей. Существующие бенчмарки в ограниченной степени затрагивают критически важные для генерации научно-энциклопедических текстов аспекты, такие как умение обобщать информацию из набора документов, планировать структуру будущего текста, соблюдать связанность и логическую последовательность изложения, а также обеспечивать точность и достоверность фактов. Одним из ближайших исследований в этой области является бенчмарк ResearchArena [6], в котором формализуют построение

¹Московский государственный университет им. М. В. Ломоносова, Москва, Россия

*E-mail: dagrig14@yandex.ru

**E-mail: chdanorbis@yandex.ru

академического обзора, однако он больше нацелен на проверку способности моделей отбирать и организовывать релевантную информацию и не затрагивает способность моделей генерировать связные научно-энциклопедические тексты.

В данной работе предлагается подход, направленный на создание инструментов, позволяющих тестировать, насколько большие языковые модели умеют работать с научно-энциклопедическими текстами. В рамках исследования:

- 1. Был собран размеченный набор данных на основе интернет-энциклопедии «Рувики»
- 2. Был разработан открытый бенчмарк RuWikiBench, позволяющий измерять качество модели на задачах, требующих глубокого анализа текста.
- 3. Были протестированы способности лучших открытых больших языковых моделей порождать статьи типа Википедия.

Код и данные работы выложены в открытый доступ¹.

СБОР ДАННЫХ

Для построения бенчмарка, направленного на оценку способности языковых моделей к работе с источниками к статьям, необходимо подготовить корпус текстов, который будет использоваться в генерации. Выбор сделан в пользу стилистики Википедии по той причине, что этот жанр одновременно требует фактологической точности, полноты анализа и понимания контекста, что хорошо соотносится с направлением исследования этой работы.

В качестве источника была выбрана российская википедия «Рувики», которая отличается большим числом ссылок на русскоязычные источники, а также более строгой фильтрацией текстов, что позволяет положиться на нее как на надежный эталон для оценки качества генерации русскоязычных статей.

Процесс получения данных включал следующие шаги:

- 1. **Выбор статей:** вручную были отобраны статьи на разнообразные темы, содержащие достаточное количество ссылок на внешние источники.
- 2. **Загрузка источников:** для каждой статьи были автоматически собраны доступные источники, на которые она ссылается.
- 3. **Разбиение на сниппеты:** для воспроизведения реальных условий Retrieval Augmented Generation (RAG), все тексты были разбиты на небольшие фрагменты длиной ≈ 600 слов.

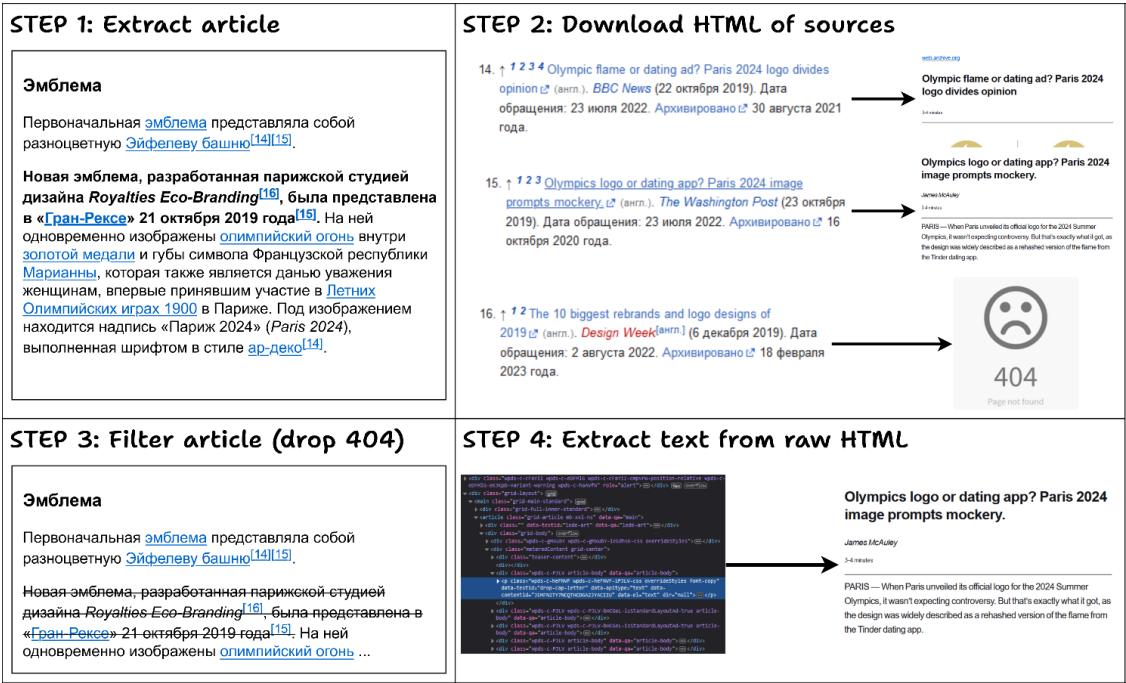


Рис. 1. Извлечение источников

¹<https://github.com/Nejimaki-Tori/WikiBench>

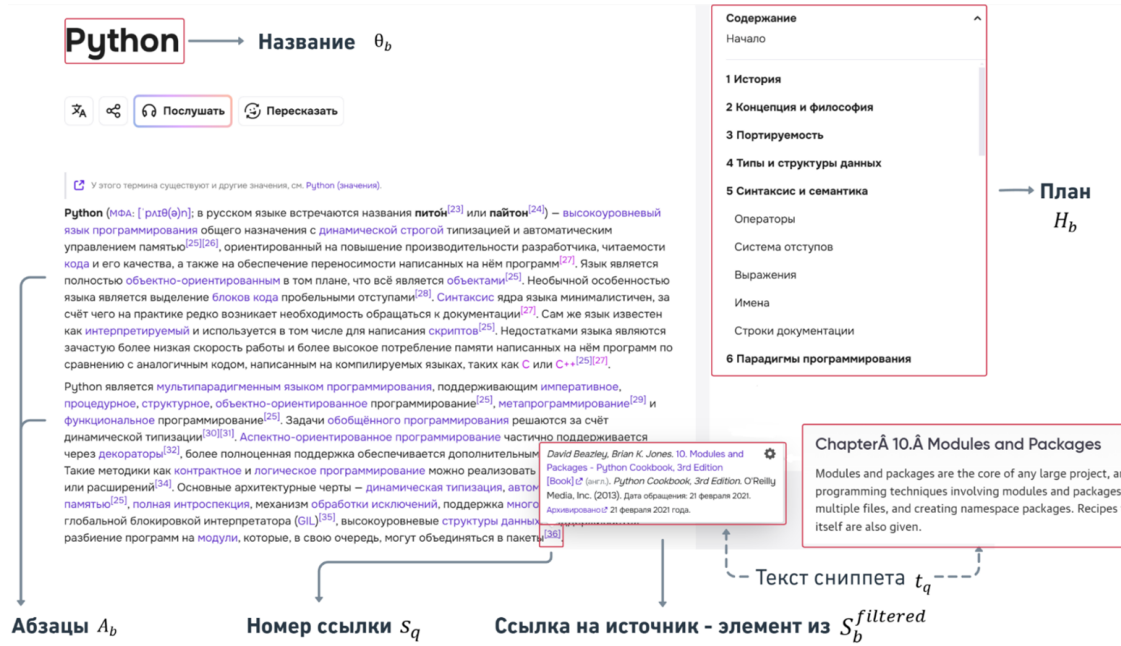


Рис. 2. Основные сущности статьи

На этапе получения данных осуществляется первичное извлечение информации из выбранной статьи и сбор связанных с ней источников. На рисунке 1 показана краткая схема извлечения текстов источников, их загрузка производилась с помощью Python-модуля `newspaper3k`². В качестве исходного корпуса берется подмножество статей «Рувики» B . Извлечение HTML-кода статьи выполняется с помощью стандартных инструментов Python-модулей^{3,4}. Полученный текст структурируется путем разбиения на фрагменты, соответствующие вложенным заголовкам (H_1 , H_2 , H_3 и т.д.), что позволяет сохранить как содержательную часть статьи, так и ее иерархическую организацию. Далее из раздела «Примечания» автоматически извлекаются все внешние ссылки, на которые ссылается статья. Недействительные ссылки (например, код 404) исключаются из дальнейшей обработки, а связанный с ними текст удаляется, оставляя только те источники, которые действительно доступны.

Рисунок 2 иллюстрирует схематичное разбиение статьи⁵ на ключевые сущности, используемые в дальнейшей обработке. На этапе обработки данных выполняется фильтрация текста для обеспечения его корректной интерпретации моделью. Каждая сноска (например, [1], [2]) сопоставляется с конкретной ссылкой, соответствующей одному из доступных источников. Это позволяет точно определить позицию ссылки в тексте статьи и использовать ее для последующей фильтрации.

ТАБЛИЦА 1. Основные характеристики собранного датасета

Показатель	RuWikiBench	ResearchArena
Количество статей	285	7,952
Количество скачанных источников	15,686	12,034,505
Общее число сниппетов	36,860	-
Средний размер плана (число заголовков)	37	-
Средний размер секции (число слов)	112	-

На основании действительных ссылок $S_b^{filtered}$ формируются очищенные множества абзацев $A_b^{filtered}$ и заголовков $H_b^{filtered}$, то есть остается только контент, подкрепленный извлеченными источниками, все прочее удаляется. Сохраняются только источники, для которых удалось получить текст t_q объемом не менее 1500 символов, чтобы отсеять «шумовые» ответы с HTML-страниц вроде ошибок (например, error 404) или сообщений о блокировке. В $A_b^{filtered}$ остаются только те абзацы, в которых присутствует хотя бы одна ссылка на источник, для которой был успешно получен текст. Аналогично формируется

²<https://github.com/codelucas/newspaper>³<https://beautiful-soup-4.readthedocs.io/en/latest/>⁴<https://requests.readthedocs.io/en/latest/index.html>⁵<https://ru.ruwiki.ru/wiki/Python>

МЕТОДИКА ОЦЕНКИ

Для объективной оценки способностей языковых моделей генерировать научно-энциклопедические тексты, необходимо воспроизвести реальный процесс подготовки энциклопедического контента:

- 1. **Отбор релевантных источников:** модель получает заголовок статьи и набор текстовых фрагментов, среди которых необходимо идентифицировать и ранжировать по степени значимости материалы, соответствующие тематике.
- 2. **Построение структуры статьи:** на основании темы и отобранных источников модель формирует план с выделением основных разделов в стиле Википедии.
- 3. **Генерация секций:** материалы статьи распределяются по разделам, после чего для каждого раздела порождается обобщение его релевантных материалов

Каждый этап оценивается независимо от предыдущих, что позволяет количественно измерить качество выполнения конкретной подзадачи.

Отбор релевантных источников. Одной из наиболее эффективных стратегий поиска [9] является предварительная генерация предполагаемого результата (описания) по исходному запросу (названию статьи) для создания расширенного запроса поиска. Описание генерируется на русском и английском языках, так как тексты источников тоже представлены в двух языковых вариантах. Запросы на обоих языках далее объединяются в единый текстовый запрос к системе поиска, основанной на BM25.

Проводились эксперименты с двумя вариантами составления запроса:

- 1. **Заранее сгенерированный запрос по названию и заголовкам второго уровня:** позволяет провести чистую оценку способностей ранжирования моделей; для генерации запроса применялась модель LLaMa 3 70b;
- 2. **Запрос, сгенерированный по названию посредством оцениваемой модели:** подобно реальным условиям, LLM полностью отвечает за качество выдачи и самостоятельно решает, какой поисковый запрос лучше сформулировать для BM25.

Примеры порождаемых описаний приведены на рисунке 3.

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.

Рис. 3. Сравнение описаний статьи «C++» в двух вариантах

Отобранные по запросу ВМ25 документы последовательно передаются большой языковой модели, которая должна определить каждый сниппет как релевантный (ответ «да») или нерелевантный (ответ «нет»). Для получения численных оценок сравниваются названия статей, к которым относятся документы из выдачи и название статьи, для которой происходит отбор текстов-источников. Берется логарифмическая вероятность токенов в ответе модели: если это был утвердительный ответ, то берется сама вероятность $P(\text{да})$, если отрицательный, то $1 - P(\text{нет})$. Такой подход позволяет ранжировать выдачу документов по уверенности модели в релевантности: чем выше вероятность, тем выше степень уверенности модели в ответе, тем выше документ будет в выдаче.

Построение структуры статьи. Сначала каждый текстовый фрагмент (сниппет) эталонного источника статьи преобразуется в векторное представление с использованием выбранной модели эмбедингов. Затем фрагменты разбиваются на кластеры - потенциальное содержание секций. Чтобы гарантировать детерминированность, применяется алгоритм Kmeans с числом кластеров равным числу заголовков 2го уровня эталонного плана и инициализацией центроидов векторными представлениями этих заголовков.

Далее отбираются пять сниппетов, наиболее близких к центру кластера. Это делается с целью снизить влияние менее релевантных фрагментов текста на итоговый план. Формирование мини-планов секций осуществляется с учетом двух ключевых параметров: размера окна контекста (для учета отсылок и общей семантики документа) и двух режимов генерации - напрямую по текстам и через предварительную генерацию краткого описания кластера. Два режима генерации позволяют выбирать уровень абстракции: прямой режим сохраняет детали при необработанных данных, а режим через предварительное описание кластера повышает согласованность формулировок и уменьшает дублирование информации. На заключительном этапе происходит объединение всех мини-планов в итоговый структурированный план статьи.

Генерация секций. Для каждой секции статьи извлекаются все сниппеты, которые указывались в качестве источников к эталонному тексту секции. Все сниппеты снова переводятся в эмбединги и строится матрица попарных сходств как произведение $Emb \times Emb^T$. Элементы со значением сходства выше порога 0.8 (значение подобрано эмпирически) считаются близкими по смыслу и объединяются в группы, чтобы избежать избыточных повторов при генерации (например, когда разные источники перефразируют одно и то же). Для каждой такой смысловой группы строится иерархическое представление: берутся первые **пять** текстов, по ним генерируется краткое описание, затем это описание дополняется на основе следующих пяти и так далее, пока не получено полное сжатое представление группы. Таким образом остается только некоторый набор кратких описаний - самая важная информация без лишних повторов. После этого по полученным описаниям групп генерируется текст секции с использованием иерархического метода реферирования [10].

ОПИСАНИЕ ПАРАМЕТРОВ ЭКСПЕРИМЕНТА

Ниже приведено описание всех использованных данных, моделей, гиперпараметров и процедуры для обеспечения воспроизводимости и анализа.

Параметры генерации. Для всех моделей, если не указано иное, использовались одинаковые параметры генерации: температура - 0.01, коэффициент штрафа за повторения - 1.0 и значение `top_p` - 0.9.

Отбор релевантных источников. Индексирование сниппетов производится с помощью ВМ25⁶ по всему корпусу собранных сниппетов без настройки гиперпараметров (стандартные значения). Для каждого релевантного документа выбираются два нерелевантных (соотношение 1:2) - это сделано для повышения устойчивости оценки.

Построение структуры статьи. Сниппеты переводились в векторное пространство с помощью модели `sergeyzh/BERTA`⁷. Рассматривались два варианта контекстного окна: используется либо нулевое окно (только сам сниппет), либо по одному соседнему сниппету слева и справа для расширения контекста.

Схожесть заголовков с эталонными сравнивалась при помощи косинусной близости: учитывалось именно смысловое соответствие, а не точная формулировка или уровень заголовка. Сравнение проводилось с очищенной структурой статьи: из предобработанного текста удалялись все заголовки, секции которых полностью состояли из текста без доступных для скачивания источников.

⁶<https://github.com/xhluca/bm25s>

⁷<https://huggingface.co/sergeyzh/BERTA>

В рамках бенчмарка применяются две группы метрик: метрики ранжирования, оценивающие, насколько хорошо модель отбирает релевантные источники, и метрики схожести текста, измеряющие соответствие сгенерированного содержания эталонному.

Метрики ранжирования. Для оценки качества списка источников используются **NDCG@K** [11], и **R-Precision** [12]:

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \quad (1)$$

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

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

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

где $\text{rel}_i \in \{0, 1\}$ - индикатор релевантности документа на позиции i ; $\text{rel}_i^{\text{IDEAL}}$ - та же величина в идеальной (полностью отсортированной) выдаче; R - общее число релевантных документов для данного запроса.

Метрика схожести текста. Качество сгенерированных секций и заголовков оценивается **BERTScore** [13]:

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

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

$$F_{\text{BERT}} = \frac{2 P_{\text{BERT}} R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \quad (7)$$

где x - эталонный текст, \hat{x} - сгенерированный; каждое предложение кодируется с помощью модели⁸, после чего вычисляется косинусное сходство.

Для оценки генераций секций также рассматривались **ROUGE-L** и **BLEU**.

ROUGE-L [14] - основана на длине наибольшей общей подпоследовательности (LCS) между сгенерированным рефератом S и эталонным R . Вычисляется по формуле (10) с использованием формул (8) и (9):

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

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

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

BLEU [15] - метрика n -граммной точности с учетом штрафа за краткость. Итоговый счет определяется по формуле (11):

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

где p_n - точность для n -грамм, w_n - веса, BP - штраф за краткость.

⁸<https://huggingface.co/sergeyzh/BERTA>

ОПИСАНИЕ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА

Используемые модели. В экспериментах использовались следующие большие языковые модели: RuadaptQwen2.5-7B-Lite-Beta [16], RuadaptQwen3-32B-Instruct-v2 [16], DeepSeek V3 [17], Qwen3-235B-A22B [18], tpro [19] и yagpt5lite [20]. Во всех таблицах модели упорядочены по размеру, а наилучшие результаты внутри каждой весовой группы выделены.

Полученные результаты. В таблицах 2 и 3 представлены результаты измерения качества ранжирования. Также были добавлены результаты бейзлайна, в качестве которого выступала выдача BM25 без переранжирования с помощью моделей. В первом случае (таблица 2), когда для всех моделей использовался заранее сгенерированный поисковый запрос, лучшие результаты показала модель DeepSeek V3, что свидетельствует о ее высокой способности отбирать релевантные документы. Во втором эксперименте (таблица 3), где запрос формировался только на основе названия статьи, tpro показала лучшие результаты. Стоит отметить, что в данной конфигурации модели тратят немного больше времени на отбор документов, так как к ранжированию еще добавляется генерация запроса на двух языках. Эксперимент показал, что самостоятельная генерация запроса BM25 не уступает по качеству ранжирования эталонному варианту, в котором запросы генерируются более «сильной» моделью и затем используются всеми оцениваемыми моделями. Предположительно, это связано с тем, что, как показано на рисунке 3, запросы получаются похожими, потому что LLM имеют представления о типовой структуре статьи Википедии из обучения и поэтому умеют связывать релевантные концепты в нужном формате.

Таблица 2. Результаты чистой оценки навыков ранжирования

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

Таблица 3. Результаты оценки навыков генерации запроса BM25

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

В целом, модели демонстрируют достаточно высокие значения метрик на данном этапе, что может быть обусловлено тем, что название статьи хорошо отражает ее содержание. В лучших случаях до 80% документов в выборке являются релевантными, что можно считать хорошим показателем, однако остается потенциал для дальнейшего улучшения.

В таблице 4 представлены результаты оценки качества построения структуры статьи. Результаты показывают, что при предварительной генерации описания (режим Description) качество работы всех моделей стабильно повышается. Наибольший прирост демонстрирует модель RuadaptQwen3, поднимаясь на второе место, фактически сравниваясь по результатам с более крупной моделью - Qwen3-235B-A22B. Лидером остается DeepSeek V3, показывая значительный отрыв от остальных. На последнем месте по качеству находятся модели tpro и yagpt5lite. При этом yagpt5lite, имея всего 8 млрд., показывает результаты сопоставимые с моделью объемом 32 млрд. параметров. На рисунке 4 приведено сравнение небольшого отрывка эталонного и сгенерированного планов. Получившиеся результаты хорошо коррелируют со степенью сходства заголовков с эталонными. Общей проблемой всех моделей была чрезмерная глубина иерархии заголовков. На «Рувики» заголовки редко были глубже третьего

Таблица 4. Результаты генерации планов

Model	Mean BERTScore F1		
	Default	Description	Default with one neighbor
DeepSeek V3	63.51	65.50	62.93
Qwen3-235B-A22B	60.86	62.66	59.06
RuadaptQwen3-32BInstruct-v2	60.12	62.91	60.04
tpro	60.32	60.75	59.09
RuadaptQwen2.5-7B-Lite-Beta	60.03	61.58	58.21
yagpt5lite	59.72	60.25	60.07

уровня, однако модели часто создавали и четвертый, и пятый, подразумевая, что вся информация находится в одной большой секции, хотя она может несколько отличаться по смыслу и в оригинальном плане это были бы не связанные заголовки.

СТЕНЕРИРОВАННЫЙ

```
# Введение в Python
## Обзор языка
### История и основные аспекты
#### Ключевые особенности и реализации
# Основы языка Python
## Синтаксис и семантика
### Типы данных и структуры
#### Числа, списки, словари
и объектно-ориентированное программирование
# Продвинутое темы Python
## Контроль потока и многопоточность
...
```

ЭТАЛОННЫЙ

```
# Python
## История
## Концепция и философия
## Портитруемость
## Типы и структуры данных
## Синтаксис и семантика
### Система отступов
### Выражения
### Имена
### Строки документации
## Парадигмы программирования
...
```

Рис. 4. Сравнение двух планов статей

В таблице 5 приведены результаты замеров качества генераций секций. Итоговые результаты находятся на одном уровне, однако это обусловлено чувствительностью используемой метрики. В конечный метрик не вошли секции, для которых алгоритмом не было отобрано ни одного релевантного снippetа. Лучшие результаты продемонстрировала модель Qwen3-235B-A22B, однако по метрикам ROUGE-L и BLEU лидирует RuadaptQwen3-32BInstruct-v2, что говорит о лучшей структурной согласованности и большем совпадении формулировок с эталоном. Модель yagpt5lite показывает результаты выше среднего, особенно по BLEU, при существенно меньшем размере, тогда как tpro демонстрирует минимальные значения по всем метрикам.

Для наглядного сравнения качества генерации секций можно рассмотреть вводные части статьи «COVID19», созданные моделями DeepSeek V3 и yagpt5lite соответственно, которые представлены на рисунке 5. Несмотря на отдельные смысловые неточности (например, утверждение «COVID-19 - это пандемия», тогда как в действительности речь идет о заболевании), модель yagpt5lite демонстрирует вполне достойный результат. Ее текст уступает варианту от DeepSeek V3 в части полноты охвата темы и системности изложения, но содержит больше числовых данных и конкретных фактов. При этом материал, сгенерированный DeepSeek V3, воспринимается как выдержка из энциклопедической статьи, тогда как версия yagpt5lite ближе по стилю к техническому отчету о заболевании.

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.

Рис. 5. Сравнение текстов двух секций

Таблица 5. Результаты генерации секций

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

ЗАКЛЮЧЕНИЕ

В заключение, в работе предложен и реализован бенчмарк RuWikiBench для оценки аналитических способностей больших языковых моделей при генерации научно-энциклопедических текстов на русском языке. В основу поставленной системы оценки лег трехэтапный процесс, состоящий из трех независимых систем, естественным образом возникающих при создании статей на определенную тему. Опираясь на отфильтрованный корпус «Рувики» с сопоставленными текстовыми фрагментами и четко определенной методикой оценки, предложенный бенчмарк создает основу для дальнейших исследований в области применения языковых моделей к задачам генерации научно-энциклопедического текста.

Эксперименты показали, что при фиксированном поисковом запросе наилучшее качество отбора источников демонстрирует DeepSeek V3, заметно опережая BM25 без переранжирования. На этапе построения структуры обнаружилось, что добавление предварительного описания кластера стабильно улучшает качество планов у всех моделей, включая DeepSeek V3, который продемонстрировал лучшее понимание процесса. Все модели показали сравнимое качество генерации текста, однако по метрикам ROUGE-L и BLEU лидирует RuadaptQwen3-32B-Instruct-v2, что указывает на более согласованную с эталоном структуру текста. Работа показывает, что модели обладают значительным потенциалом, но для их надежного применения требуется дальнейшая проработка методов анализа и структурирования обзорных материалов.

БЛАГОДАРНОСТИ

Исследование выполнено за счет гранта Российского научного фонда № 25-11-00191, <https://rscf.ru/project/25-11-00191/>.

Работа выполнялась с использованием суперкомпьютера «МГУ-270» МГУ имени М.В. Ломоносова.

СПИСОК ЛИТЕРАТУРЫ

- [1] *RussianSuperGlue*. Shavrina T. et al. RussianSuperGLUE: A Russian language understanding evaluation benchmark // EMNLP 2020-2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. – 2020. – С. 4717-4726.
- [2] *Mera*. Fenogenova A. 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. – С. 9920-9948.
- [3] *LIBRA*. Churin I. et al. Long Input Benchmark for Russian Analysis // CoRR. – 2024. Fenogenova
- [4] *Ru Arena General*. Li T. et al. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline // CoRR. – 2024
- [5] *Ping-Pong*. Gusev I. PingPong: A Benchmark for Role-Playing Language Models with User Emulation and Multi-Model Evaluation // arXiv e-prints. – 2024. – С. arXiv: 2409.06820
- [6] *ResearchArena*. Kang H., Xiong C. ResearchArena: Benchmarking LLMs' Ability to Collect and Organize Information as Research Agents // arXiv e-prints. – 2024. – С. arXiv: 2406.10291.
- [7] *OpenAI*. Introducing deep research // OpenAI URL: <https://openai.com/index/introducing-deep-research/> (дата обращения: 31.07.2025).
- [8] *Storm*. Shao Y. et al. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models // NAACL-HLT. – 2024
- [9] *Reranking*. Wang X. et al. Searching for Best Practices in Retrieval-Augmented Generation // CoRR. – 2024.
- [10] Wu J. et al. Recursively Summarizing Books with Human Feedback // arXiv e-prints. – 2021. – С. arXiv: 2109.10862.
- [11] *NDCG*. Järvelin K., Kekäläinen J. Cumulated gain-based evaluation of IR techniques // ACM Transactions on Information Systems (TOIS). – 2002. – Т. 20. – №. 4. – С. 422-446.
- [12] *RPrecision*. BUCKLEY C. Evaluating Evaluation Measure Stability // ACM SIGIR 2000 Proceedings. – 2000.
- [13] *BERTScore*. Zhang T. et al. BERTScore: Evaluating Text Generation with BERT // International Conference on Learning Representations.
- [14] *ROUGE*. Lin C. Y. Rouge: A package for automatic evaluation of summaries // Text summarization branches out. – 2004. – С. 74-81.
- [15] *BLEU*. Papineni K. et al. BLEU: a Method for Automatic Evaluation of Machine Translation.
- [16] *RuadaptQwen*. Tikhomirov M., Chernyshev D. Facilitating large language model russian adaptation with learned embedding propagation // Journal of Language and Education. – 2024. – Т. 10. – №. 4 (40). – С. 130-145.
- [17] *DeepSeek V3*. Liu A. et al. DeepSeek-V3 Technical Report // CoRR. – 2024.
- [18] *Qwen3-235B*. Yang A. et al. Qwen3 technical report // arXiv preprint arXiv:2505.09388. – 2025.
- [19] Т-Банк открыл доступ к собственной русскоязычной языковой модели в весовой категории 7-8 млрд параметров
Т-Банк 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/> (дата обращения: 10.05.2025).
- [20] YandexGPT 5 с режимом рассуждений // Яндекс URL: <https://ya.ru/ai/gpt?ysclid=ma19jrssc8906806775> (дата обращения: 30.07.2025).

RUWIKIBENCH: EVALUATING LARGE LANGUAGE MODELS THROUGH REPLICATION OF ENCYCLOPEDIA ARTICLES

D. A. Grigoriev^{a,*}, D. I. Chernyshev^{a,**}

^aLomonosov Moscow State University, Moscow Center for Fundamental and Applied Mathematics,
Moscow, Russian Federation

Presented by ...

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.

Keywords: benchmark, Wikipedia, Ruwiki, large language model