ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2025, том 505, с. 46–49

МАТЕМАТИКА

УДК 517.54

ОЦЕНКА ОБЩИХ И СПЕЦИАЛЬНЫХ ЗНАНИЙ В БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЯХ ДЛЯ РУССКОГО ЯЗЫКА ПОСРЕДСТВОМ ВОСПРОИЗВЕДЕНИЯ ЭНЦИКЛОПЕДИЧЕСКИХ СТАТЕЙ

© **2025 г.** **Д. А. Григорьев**[1,*], **Д. И. Чернышев**[1,**]

Представлено кем-то
Поступило 16.08.2025
После доработки 20.08.2025
Принято к публикации 31.08.2025

В данной работе предложен и реализован бенчмарк WikiBench для оценки аналитических способностей больших языковых моделей при составлении научно-энциклопедических текстов на русском языке.

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

DOI: 10.31857/S2686954322040117

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. Традиционные методы создания научных статей требуют значительных временных затрат на поиск и анализ информации. Если большая языковая модель сможет самостоятельно проводить такие "глубокие"исследования по темам, не входящим в ее первоначальные тренировочные данные, это позволит отказаться от постоянного дообучения моделей и предложить более эффективный и масштабируемый подход к работе с постоянно обновляющимся научными знаниями. 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 на русском языке, охватывающих различные лингвистические задачи для русского языка. RussianSuperGlue [1] оценивает общее языковое понимание и базовые задачи по обработки 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 фокусируется 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.

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 ResearchArena [6], в которой формализуют построение академического обзора с помощью трех этапов: обнаружение релевантной литературы, отбор по значимости и

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

[*] *E-mail: dagrig14@yandex.ru*

[**] *E-mail: chdanorbis@yandex.ru*

организация знаний. Однако этот бенчмарк больше нацелен на проверку способности моделей отбирать и организовывать релевантную информацию и не затрагивает способности модели генерировать связные научно-энциклопедические тексты. Также развиваются в направлении похожих задач такие методы как Storm [7] - он подготавливает статью через создание множества перспектив и диалогов между ними. Однако это не позволяет в полной мере отследить каждый этап генерации, и дает меньший контроль над параметрами генерации. Недавнее развитие новых способностей агентов, например появление функции «Deep Research» у OpenAI [8], свидетельствует о возрастающем интересе к проведению научных исследования с помощью больших языковых моделей, что говорит о необходимости в создании новых подходов к объективной оценке аналитических способностей моделей.

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 и протестирован открытый бенчмарк WikiBench, позволяющий измерять качество модели на задачах, требующих глубокого анализа текста.

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 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.

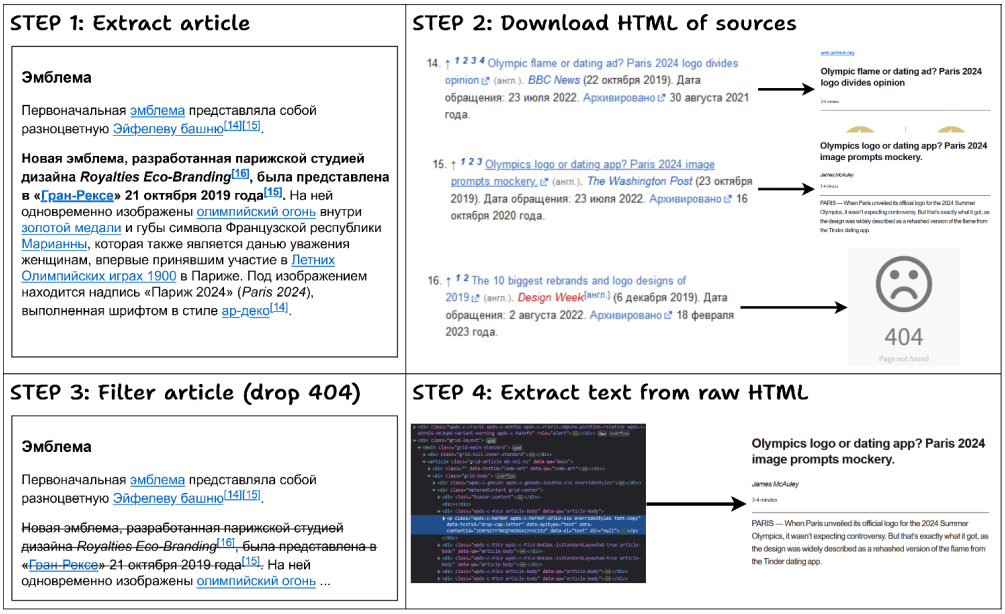

Рис. 1. Извлечение источников

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

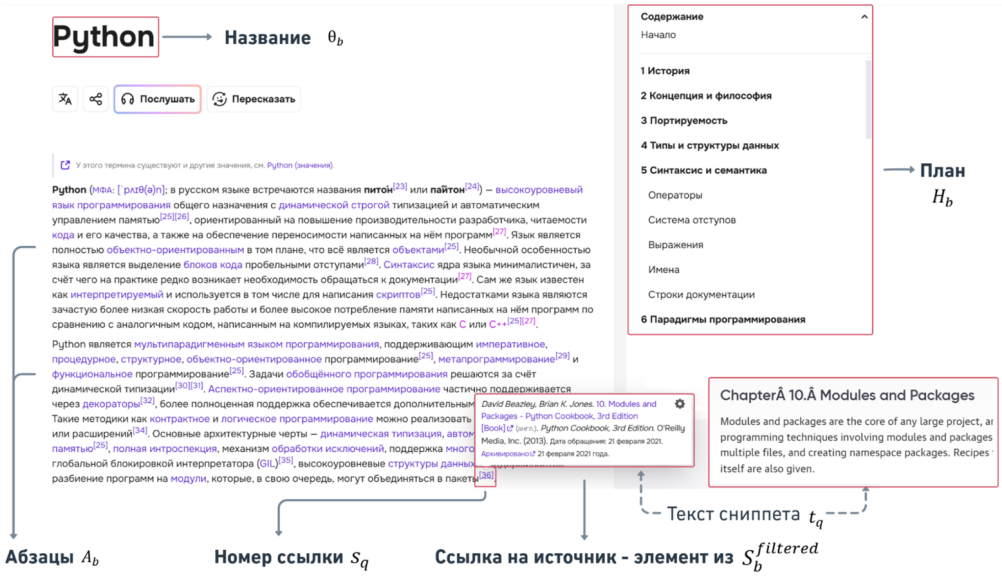

Рис. 2. Основные сущности статьи

источников, их загрузка производилась с помощью Python-модуля `newspaper3k`[1]. В качестве исходного корпуса берётся подмножество статей Википедии B. Извлечение HTML-кода статьи выполняется с помощью стандартных инструментов Python-модулей[2],[3]. Полученный текст структурируется путем разбиения на фрагменты, соответствующие вложенным заголовкам (H1, H2, H3 и т.д.), что позволяет сохранить как содержательную часть статьи, так и её 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 статьи[4] на ключевые сущности, используемые в дальнейшей обработке. На этапе обработки данных выполняется фильтрация текста для обеспечения его корректной интерпретации моделью. Каждая сноска, обозначенная цифрами в квадратных скобках (например, [1], [2]), сопоставляется с конкретной ссылкой, соответствующей одному из доступных источников.

Это позволяет точно определить позицию ссылки в тексте статьи и использовать её для последующей фильтрации. На основании S_b^{filtered} формируются очищенные множества абзацев A_b^{filtered} и заголовков H_b^{filtered}, то есть остаётся только тот контент, подкреплённый извлечёнными источниками; всё прочее удаляется.

Таблица 1. Основные характеристики собранного датасета

Показатель	**WikiBench**	**ResearchArena**
Количество статей	100	7,952
Количество скачанных источников	5,828	12,034,505
Общее число сниппетов	13,704	-
Средний размер плана (число заголовков)	37	-
Средний размер секции (число слов)	112	-

Сохраняются только источники, для которых удалось получить текст t_q объёмом не менее 1500 символов, чтобы отсечь «шумовые» ответы с HTML-страниц вроде ошибок (например, error 404) или сообщений о блокировке. Абзацы очищаются следующим образом: 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 H_b^{filtered} - только те заголовки, под которыми остался поддерживаемый источниками текст (хотя бы один абзац). В результате в статье остаются только те части текста,

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

которые подкреплены достоверными и доступными источниками. Характеристики собранного корпуса представлены в таблице 1.

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 в примере 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 BM25 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(«да»), если отрицательный, то берется обратное значение, то есть 1 − P(«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 эмбеддингов.

Затем фрагменты разбиваются на кластеры - потенциальное содержание секций. Чтобы гарантировать детерминированность применяется алгоритм Kmeans с числом кластеров равному числу заголовков 2го уровня эталонного плана и инициализацией центроидов векторными представлениями этих заголовков.

Далее отбирается 5 сниппетов, наиболее близких к центру кластера. Это делается с целью, чтобы снизить влияние менее релевантных фрагментов текста на итоговый план. Формирование мини-планов секций осуществляется с учётом двух ключевых параметров: размера контекста (определяемого числом соседних сниппетов) и двух режимов генерации - напрямую по текстам и через предварительную генерацию краткого описания кластера. Размер контекста позволяет контролировать баланс между точностью и полнотой: меньшее контекстное окно уменьшает риск отклонений от истинной структуры плана, большее позволяет увеличить покрытие фактов, но также может увеличиться избыточность информации. Два режима генерации позволяют выбирать уровень абстракции: прямой режим сохраняет детали при необработанных данных, а режим через предварительное описание 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 эмбеддинги и строится матрица попарных сходств как произведение $Emb \times Emb^{\top}$, что по сути даёт косинусные близости между векторами. Элементы с 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 BM25[5] 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`[6]. Рассматривались два варианта контекстного окна: используется либо нулевое окно (только сам сниппет), либо по одному соседнему сниппету слева и справа для расширения контекста.

[5]https://github.com/xhluca/bm25s
[6]https://huggingface.co/sergeyzh/BERTA

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.

МЕТРИКИ ОЦЕНКИ КАЧЕСТВА

В рамках бенчмарка применяются две группы метрик: (1) метрики ранжирования, оценивающие, насколько хорошо модель отбирает релевантные источники; (2) метрики текстовой схожести, измеряющие соответствие сгенерированного содержания эталонному.

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

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

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

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

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

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]:

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

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

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

где x - эталонный текст, $\hat{x}$ - сгенерированный; каждое предложение кодируется эмбеддингом с помощью модели[7], 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):

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

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

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

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

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

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

[7] https://huggingface.co/sergeyzh/BERTA

ОПИСАНИЕ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА

Используемые модели. В экспериментах использовались следующие большие языковые модели: RuadaptQwen3-32BInstruct-v2 [16], DeepSeek V3 [17], Qwen3-235B-A22B [18], tpro [19] и yagpt5lite [20].

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 её высокой способности отбирать релевантные документы. При использовании заранее сгенерированного поискового запроса лучшие результаты продемонстрировала модель DeepSeek V3, что говорит о хорошей способности модели отбирать документы. Во втором эксперименте (таблица 3), где запрос формировался только на основе названия статьи, лидирующее качество продемонстрировала модель tpro. 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 в примере 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
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**
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 её содержание. В лучших случаях до 80% документов в выборке являются релевантными, что можно считать хорошим показателем, однако остаётся потенциал для дальнейшего улучшения.

Таблица 4. Результаты генерации планов

Model	**Default**		**Description**	
	Mean F1	**[Min; Max] F1**	**Mean F1**	**[Min; Max] F1**
DeepSeek V3	63.51	[62.88; 64.08]	**65.50**	[64.82; 66.33]
Qwen3-235B-A22B	60.86	[60.20; 61.41]	**62.66**	[61.90; 63.47]
RuadaptQwen3-32B Instruct-v2	60.12	[59.20; 60.93]	**62.91**	[62.32; 63.52]
tpro	60.32	[59.68; 60.87]	**60.75**	[59.89; 61.60]
yagpt5lite	59.72	[58.82; 60.68]	**60.25**	[59.48; 61.00]

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 модель от Яндекса, имея всего 8 млрд., показывает результаты сопоставимые с моделью объемом 32 млрд. параметров. В примере 4 приведено сравнение небольшого отрывка эталонного и сгенерированного планов. Получившиеся результаты хорошо коррелируют со степенью сходства заголовков с эталонными. Общей проблемой всех моделей был слишком сильный уход в «глубину», однако на «Рувики» заголовки редко были выше третьего уровня, 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.

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	### Выражения
и объектно-ориентированное программирование	### Имена
# Продвинутые темы Python	### Строки документации
## Контроль потока и многопоточность	## 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 сниппета. 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.

Например, можно сравнить вводные секции статьи «COVID19», созданные моделями DeepSeek V3 и yagpt5lite соответственно, которые представлены в примере 5. Несмотря на отдельные смысловые неточности (например, утверждение «COVID-19 - это пандемия», тогда как в действительности речь идёт о заболевании), модель yagpt5lite демонстрирует вполне достойный результат. Её 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 отчёту о заболевании.

Таблица 5. Результаты генерации секций

Model	**Mean F1**	**Mean RougeL**	**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
yagpt5lite	53.43	14.85	3.16

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

ЗАКЛЮЧЕНИЕ

В заключение, в работе предложен и реализован бенчмарк WikiBench 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. Процесс включал в себя: отбор и ранжирование источников, построение плана статьи, генерация текстов разделов в стиле Википедии. 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. Работа показывает, что модели обладают значительным потенциалом, но для их надежного применения в академическом контексте требуется дальнейшая проработка методов контроля за достоверностью и структурой создаваемого контента.

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

[8] *OpenAI.* Introducing deep research // OpenAI URL: https://openai.com/index/introducing-deep-research/ (дата обращения: 31.07.2025).

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

[12] *RPrecion.* Järvelin K., Kekäläinen J. Cumulated gain-based evaluation of IR techniques //ACM Transactions on Information Systems (TOIS). – 2002. – Т. 20. – №. 4. – С. 422-446.

[13] *BERTScore.* BUCKLEY C. Evaluating Evaluation Measure Stability //ACM SIGIR 2000 Proceedings. – 2000.

[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=mal9jrssc8906806775 (дата обращения: 30.07.2025).

EVALUATING GENERAL AND SPECIAL KNOWLEDGE IN LARGE LANGUAGE MODELS FOR RUSSIAN LANGUAGE THROUGH REPLICATION OF ENCYCLOPEDIA ARTICLES

D. A. Grigoriev[a,*], D. I. Chernyshev[a,]**

[a]Lomonosov Moscow State University, Moscow Center for Fundamental and Applied Mathematics, Moscow, Russian Federation

Presented by Academician of the RAS B. S. Kashin

In this work we propose and implement WikiBench for evaluating the analytical capabilities of large language models in generating scientific and encyclopedic texts in Russian.

Keywords: benchmark, Wikipedia, Ruwiki, large language model