Literature summarization with large language models

D. A. Grigoriev[1,2], D. V. Khudiakov[1,3], D. I. Chernyshev[1,4]

The work is dedicated to automatic summarization of Russian literary fiction. An open corpus RuBookSum (600+ "book-summary" pairs) was compiled based on texts from LibRuSec and user-generated summaries from «Народный Брифли». Four approaches to summarization are examined: the baseline hierarchical method and the "blueprint" (Text-Blueprint) approach, which builds an outline in the form of question-answer pairs, as well as two new methods proposed in this work: a hierarchical method with node filtering based on cosine similarity of embeddings, and a modification of the blueprint approach with question clustering using the KMeans algorithm. The best quality is achieved by Qwen3-235B-A22B in the blueprint method, while the hierarchical method with filtering provides the best balance between generation time and quality. Keywords: LLM, summarization, literature, books, brief retelling.

Introduction

Automatic text summarization is one of the key tasks in natural language processing. The goal is to create an informative summary of the source text while preserving its main meaning. In recent years, with the advent of large language models (LLMs), interest in automating summarization has increased across many genres, including fiction. Unlike scientific, news, or technical texts, fiction is characterized by high stylistic and semantic complexity. Non-linear storytelling, imagery, metaphor, and stylistic devices make short synopsis writing especially challenging. The limited context window of modern models further complicates processing long works.

At present moment there are not many datasets focusing specifically on summarizing fiction, and the key open datasets concentrate on non-Russian material. BookSum [1] is one of the first and best-known English-language datasets for abstractive summarization of narrative works. It contains books, plays, and short stories paired with summaries of varying granularity (paragraph level, chapter level, book level). Echoes from Alexandria [2] is a multilingual corpus of fiction, including five languages: English, German, French, Italian, and Spanish. FABLES [3] is a hand-curated corpus designed to evaluate factual faithfulness of summaries for book-length fiction. It includes 3,158 claims extracted from LLM-generated summaries for 26 books. Each claim is evaluated across model outputs by experts. According to FABLES, even advanced models (e.g., Claude) commit 20–30% factual errors, including distorted causal relations, incorrect characterization of protagonists, and overemphasis on minor details, judged by three criteria: agreement with original events, logical correctness, and absence of distortions.

In theory, automatic summarization can be performed in two main ways: extractive (selecting key text fragments) and abstractive (generating new text based on the source). For prose, abstractive summarization is typically chosen: key meanings and plot links are distributed throughout the text, so extractive sentence selection yields a fragmented, stylistically uneven result and does not reconstruct the plot, therefore abstractive approach was chosen.

[1]Lomonosov Moscow State University, Moscow Center for Fundamental and Applied Mathematics, Moscow, Russian Federation
[2]E-mail: dagrig14@yandex.ru
[3]E-mail: hydikovv17914@gmail.com
[4]E-mail: chdanorbis@yandex.ru

The topic is motivated by the growing need for tools capable of automatically producing concise, informative, and stylistically appropriate synopses for works of fiction. The goal of this work is to provide such tools, thus, following is proposed:

1. New Russian-language dataset that includes literary works and their synopses;
2. New summarization methods that offer alternatives to existing ones and substantially reduce the time required to produce a short synopsis of a book.

   Code and data are publicly available[5].

1. Dataset

At the start of the study, there were no open and representative corpora designed specifically for summarizing fiction in Russian. To run experiments and evaluate different approaches, we created our own corpus consisting of works of fiction and corresponding short synopses. As the source of synopses we used the "Narodny Briefly" platform [4] where users publish short summaries of literary works.

The synopses are user-generated texts based on the original works. They vary in length—from a few sentences to several paragraphs—and in style: some reproduce key phrases verbatim, while others use freer narration. Some cover the whole work, others split content by chapter. Usually they contain the main facts and conclusions from the source text, but may include the author's commentary.

The book texts were selected from the LibRuSec electronic library [5], one of the largest Russian-language fiction resources. Works were selected for which a synopsis existed on chosen source [4]. Each text underwent automatic preprocessing: meta-information (e.g., titles, chapter descriptions, technical inserts) was removed, then the text was formatted into a unified, standardized form suitable for use with models.

To better link books with their synopses, semantic similarity was used: the author name text from Briefly [4] and from LibRuSec [5] was embedded via SentenceTransformer with the model[6] and compared using cosine similarity.

Таблица 1. Dataset overview

Dataset	Number of documents	Avg. document length (# words)	Avg. summary length (# words)	Compression ratio (summary length / text length)
RuBookSum	634	35052.64	700.77	8.43%
BookSum	405	112885.15	1167.20	0.79%
Gazeta	60964	632.77	41.94	6.99%

The synopses were automatically cleaned of HTML tags, comments, and service markers using LLM Meta-Llama 3-70B-Instruct. Then LibRuSec was searched and a collection of "book text – synopsis" pairs was formed.

The resulting corpus includes:

- 600+ cleaned user synopses from "Narodny Briefly" [4];
- 40+ different genres;

[5]https://github.com/Nejimaki-Tori/BookSum

[6]https://huggingface.co/deepvk/USER-bge-m3

- source works from the LibRuSec electronic library [5].

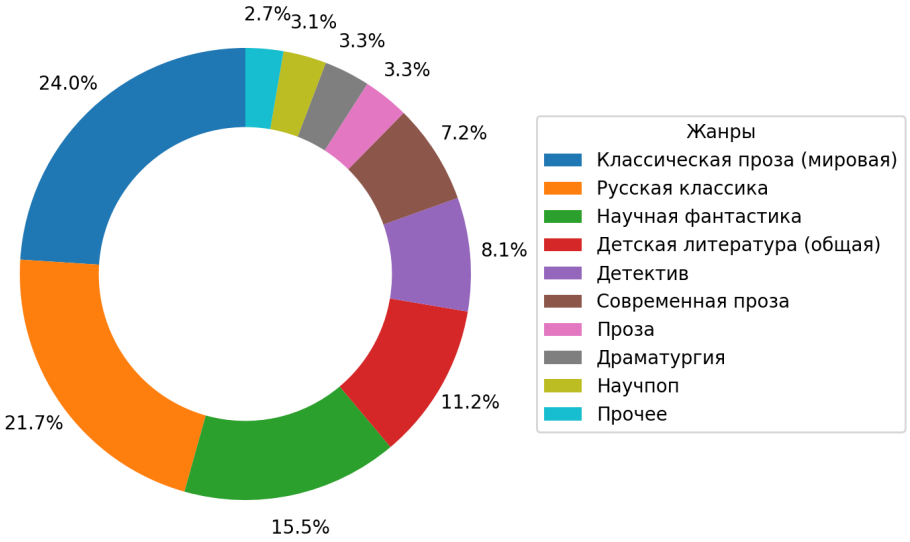

Рис. 1. Distribution of texts by genres (top 10 genres)

Figure 1 shows the genre distribution in the collection. Table Tab. 1 gives dataset statistics versus analogs.

2. Методология

2.1. Базовые и модифицированные стратегии.

2.1.1. Иерархический метод. (Algorithm 1)

Суть этого метода [6] заключается в том, что текст разбивается на фрагменты (чанки), для каждого из которых отдельно генерируется локальный реферат. Эти фрагменты затем объединяются в группы, и из полученных рефератов снова формируется краткое содержание следующего уровня. Последний уровень представляет собой итоговый реферат всего произведения.

2.1.2. Иерархический метод с фильтрацией узлов. (Algorithm 2)

Классический иерархический метод строит итоговый реферат путём многослойного объединения промежуточных рефератов, полученных из отдельных фрагментов текста. Однако в литературных произведениях часто встречаются фрагменты, которые не оказывают большого влияния на развитие сюжета и содержат множество избыточных повторов и второстепенной информации. Эти фрагменты при генерации итогового реферата могут снижать её информативность, а в некоторых случаях даже мешать модели на этапе реферирования отдельных фрагментов.

Чтобы решить эту проблему, в метод была имплементирована фильтрация узлов по семантической близости. Для исключения малоинформативных или дублирующих фрагментов на каждом уровне иерархии выполняется глобальная проверка семантической близости между всеми промежуточными рефератами. Фрагменты, близкие по косинусной мере

с предыдущими, считаются избыточными и не используются при составлении реферата на текущем уровне. Эмбеддинги получаются с помощью SentenceTransformer (модель USER-bge-m3) и при вычислении на GPU обеспечивается высокая скорость обработки. Эта модификация направлена на ускорение генерации за счет удаления потенциально излишних частей информации, что повышает плотность полезной информации в итоговых рефератах.

Algorithm 1 Иерархический метод

Require: W - контекстное окно модели, D - входной текст, длиной $L \gg W$, p_θ - модель, C - длина чанка

Разбить D на чанки $c_1 \dots c_{\lceil \frac{L}{C} \rceil}$
for $c_i = c_1 \dots c_{\lceil \frac{L}{C} \rceil}$ **do**
    $S_0 \leftarrow SummarizeChunk(p_\theta, c_i)$
end for
repeat
    $Groups \leftarrow GroupSummaries(S_l)$
    $\ell \leftarrow \ell + 1$
    for $g \in Groups$ **do**
        $S_l \leftarrow \{MergeGroup(p_\theta, g)\}$
    end for
until $|S_l| = 1$
return $S_l[1]$

Algorithm 2 Иерархический метод с фильтрацией

Require: W - контекстное окно модели, D - входной текст, длиной $L \gg W$, p_θ - модель, θ - порог сходства, C - длина чанка

Разбить D на чанки $c_1 \dots c_{\lceil \frac{L}{C} \rceil}$
$S_0 \leftarrow \{c_1 \dots c_{\lceil \frac{L}{C} \rceil}\}$
repeat
    for $s_i \in S_l$ **do**
        $e_i \leftarrow Encoder(s_i)$
        $M_{ij} \leftarrow \dfrac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \, \|\mathbf{e}_j\|}$ ▷ Матрица эмбеддингов
        Вычисляется максимальное сходство
        с предыдущими рефератами.
        $m_j = \max_{i<j} M_{ji}$
        $S_l \leftarrow \{s_i \mid m_i < \theta \;\text{ or }\; i = 0\}$ ▷ Фильтрация
    end for
    $Groups \leftarrow GroupSummaries(S_l)$
    $\ell \leftarrow \ell + 1$
    for $g \in Groups$ **do**
        $S_l \leftarrow \{MergeGroup(p_\theta, g)\}$
    end for
until $|S_l| = 1$
return $S_l[1]$

2.1.3. «Чертёжный» метод (Text-Blueprint). (Algorithm 3)

Данный метод [7] по сути является модификацией иерархического и ориентирован на построение промежуточного плана реферата перед генерацией текста. План формируется в виде набора вопросно-ответных пар, что повышает управляемость генерации и обеспечивает структурированность результата. Сначала модель формирует список вопросов, отражающих ключевые события, темы и персонажей текста. Далее к каждому вопросу автоматически подбирается краткий ответ. Эта структура служит планом, по которому генерируется итоговый реферат.

2.1.4. «Чертёжный» метод с кластеризацией вопросов. (Algorithm 4)

Базовая реализация «чертёжного» метода предполагает генерацию вопросно-ответного плана для каждого фрагмента текста и каждого уровня объединения рефератов. Однако при работе с художественными текстами вопросы, генерируемые для каждого чанка, могут пересекаться и порождать противоречивые ответы, то в свою очередь сбивает агрегацию текста моделью, делая реферат менее структурированным и содержательно полным. К тому же, генерация плана на каждом шаге алгоритма существенно замедляет его работу и использует дополнительные мощности языковых моделей. Для снижения числа запросов к модели и повышения структурности, была добавлена кластеризация вопросов с использованием SentenceTransformers и алгоритма K-means.

Algorithm 3 «Чертежный» метод

Require: W - контекстное окно модели, D - входной текст, длиной $L \gg W$, p_θ - модель, C - длина чанка, R - ограничение по длине

Разбить D на чанки $c_1 \dots c_{\lceil \frac{L}{C} \rceil}$

for $c_i = c_1 \dots c_{\lceil \frac{L}{C} \rceil}$ **do**

$b_i \leftarrow GenerateBlueprint(p_\theta, c_i)$

$S_0 \leftarrow \{SummarizeWithBp(p_\theta, b_i, c_i)\}$

end for

repeat ▷ Объединение рефератов

$Groups \leftarrow GroupSummaries(S_l)$

$\ell \leftarrow \ell + 1$

for $g \in Groups$ **do**

if $Length(g) > R$ **then**

$b_i \leftarrow GenerateBlueprint(p_\theta, g)$

$S_l \leftarrow \{SummarizeWithBp(p_\theta, b_i, g)\}$

else

$S_l \leftarrow \{g\}$

end if

end for

until $|S_l| = 1$

return $S_l[1]$

Algorithm 4 «Чертежный» метод с кластеризацией

Require: W - контекстное окно модели, D - входной текст, длиной $L \gg W$, p_θ - модель, C - длина чанка, R - ограничение по длине

Разбить D на чанки $c_1 \dots c_{\lceil \frac{L}{C} \rceil}$

for $c_i = c_1 \dots c_{\lceil \frac{L}{C} \rceil}$ **do**

$b_i \leftarrow GenerateBlueprint(p_\theta, c_i)$

$Q \leftarrow \{ExtractQuestions(p_\theta, b_i)\}$

end for

for $q_i \in Q$ **do**

$E \leftarrow \{Encoder(q_i)\}$

$K \leftarrow KMeans(E)$

for $k_i \in K$ **do**

$q_i \leftarrow Generalize(p_\theta, k_i)$

$Q \leftarrow \{q_i\}$ ▷ Собирается общий план

end for

for $c_i = c_1 \dots c_{\lceil \frac{L}{C} \rceil}$ **do**

$S_0 \leftarrow \{SumWithBp(p_\theta, b_i, c_i)\}$

end for

end for

Объединение рефератов аналогично «Чертежному методу» с тем отличием что здесь в качестве чертежа используется один глобальный план Q

Такой подход позволяет уменьшить число обращений к LLM, что позволяет ускорить скорость генераций, как будет показано в таблице 2.

3. Метрики

Для объективного сравнения описанных подходов и моделей в задаче реферирования художественных текстов использовались четыре группы метрик.

ROUGE-L [8] - основана на длине наибольшей общей подпоследовательности (LCS) между сгенерированным рефератом S и эталонным R. Вычисляется по формуле (3) с исполь-

зованием формул (1) и (2):

$$\text{Precision} = \frac{\text{LCS}(S, R)}{|S|}, \tag{1}$$

$$\text{Recall} = \frac{\text{LCS}(S, R)}{|R|} \tag{2}$$

$$\text{ROUGE-L} = \frac{2\,\text{Precision}\,\cdot\,\text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

BERTScore [9] - семантическое качество на уровне токенов. Для каждой пары токенов предсказания и эталона вычисляется косинусное сходство их эмбеддингов в модели USER-bge-m3. Затем:

$$P = \frac{1}{|S|} \sum_{t \in S} \underset{u \in R}{\text{maxsim}}(e_t, e_u), \tag{4}$$

$$R = \frac{1}{|R|} \sum_{u \in R} \underset{t \in S}{\text{maxsim}}(e_u, e_t) \tag{5}$$

$$\text{BERTScore} = \frac{2\,P\,R}{P + R} \tag{6}$$

В формулах (4), (5) и (6) S - эталонный текст, R - сгенерированный; каждое предложение кодируется эмбеддингом модели USER-bge-m3, после чего вычисляется косинусное сходство.

Полнота покрытия ключевых вопросов (Coverage) - доля заранее сгенерированных по эталонному тексту вопросов с помощью модели Qwen3-235B-A22B [10], на которые модель «отвечает» в реферате:

$$\text{Coverage} = \frac{\#\{q_i \colon P(\text{“да”} \mid q_i, S) > 0.75\}}{N} \tag{7}$$

В формуле (7) N - общее число вопросов, а $P(\text{“да”} \mid q_i, S)$ - вероятность наличия ответа на вопрос q_i в тексте S, полученная с помощью LLM (Qwen3-235B-A22B [10]).

Совпадение ответов (AnswerSimilarity) - среднее семантическое сходство между сгенерированными ответами a_i^{pred} и эталонными a_i^{ref} на те же ключевые вопросы:

$$\text{AnswerSimilarity} = \frac{1}{N} \sum_{i=1}^{N} \text{sim}(a_i^{\text{pred}},\, a_i^{\text{ref}}) \tag{8}$$

В формуле (8) sim - косинусное сходство эмбеддингов, полученных через USER-bge-m3.

Использование нескольких метрик, учитывающих как поверхностное совпадение текста, так и глубокое семантическое сходство (BERTScore, AnswerSimilarity), а также степень охвата заранее заданных вопросов (Coverage), обеспечивает всестороннюю и устойчивую оценку качества рефератов.

4. Параметры экспериментов

Все представленные в работе измерения выполнены на тестовой части датасета, отобранных так, чтобы исходные тексты не превышали по длине 800 000 символов. Для всех методов генерируемые рефераты ограничивались максимумом в 500 слов.

Текст на вход разбивался на чанки фиксированного размера в 2000 токенов. Токенизация выполнялась с помощью AutoTokenizer модели DeepPavlov/rubert-base-cased в стандартном режиме. Для воспроизводимости всех случайных процедур использовался фиксированный seed ($random\_seed = 42$).

В иерархическом методе с фильтрацией узлов для оценки избыточности промежуточных рефератов на каждом уровне вычислялась матрица косинусных сходств между их эмбеддингами. Порог схожести был установлен равным $\theta = 0.85$: если для реферата S_j существует предыдущий S_i с косинусным сходством выше этого порога, то S_j отбрасывается как избыточный. Такой выбор порога обеспечивает компромисс между сохранением значимой информации и устранением дублирования, что эмпирически привело к заметному уменьшению объёма промежуточных представлений без существенной деградации качества.

В чертёжном методе с кластеризацией вопросов количество кластеров для K-means выбирается по эвристике, подобранной эмпирически, представленной в формуле (9):

$$n_{\text{clusters}} = \max\Big(2,\ \Big\lceil\sqrt{N_{\text{questions}}}\Big\rceil\Big) \tag{9}$$

где $N_{\text{questions}}$ - общее число сгенерированных вопросов по всем чанкам до кластеризации. Гарантируется минимум в два кластера, что позволяет даже при небольших наборах вопросов получать структурированное чертёжное представление.

Временные показатели измерялись как среднее значение (в секундах) времени генерации одной книги по каждому методу для 100 книг. В случае всех четырех методов учитывалось суммарное время всех этапов (включая генерацию промежуточных рефератов / планов, фильтрацию и финальную агрегацию).

5. Результаты

5.1. Используемые модели.

В экспериментах использовались следующие большие языковые модели: RuadaptQwen2.5-7B-Lite-Beta [11], RuadaptQwen3-32BInstruct-v2 [11], DeepSeek V3 [12], Qwen3-235B-A22B [10], tpro [13] и yagpt5lite [14].

5.2. Полученные результаты.

В таблице 2 представлены сравнительные метрики качества автоматического пересказа книг разными моделями и методами обработки. Для каждой комбинации модели и метода измерялись BERTScore, ROUGEL, Answer Coverage и Similarity, а также время генерации (среднее) на 100 примерах, одинаковых для всех замеров. Лучше всего себя показала модель Qwen3-235B-A22B: она продемонстрировала самые высокие показатели в покрытие вопросов и сходстве ответов. В то же время важно отметить, что среди всех методов лучшим образом в соотношение качество и время обработки себя показывает иерархический метод с фильтрацией узлов. Он позволяет существенно ускорить время обработки (например, почти в два раза для модели DeepSeek V3), и по сравнению с чертежным методом, который в среднем показывал лучшие результаты, не сильно отстает по показателям. Исключением стала лишь модель Qwen3-235B-A22B, так как она показала лучший результат среди всех моделей на базовом чертежном методе. Эксперименты показали, что иерархический метод

Таблица 2. Результаты по методам и моделям

Модель	Метрики	Иерархический	Чертежный	Иерархический с фильтрацией	Чертежный с кластеризацией
DeepSeek V3	bertscore	60.0 ± 3.1	58.0 ± 4.0	60.0 ± 2.9	58.4 ± 3.6
	rouge-l	13.7 ± 3.9	12.6 ± 4.6	13.5 ± 3.7	11.2 ± 3.9
	coverage	53.57 ± 21.66	40.19 ± 23.68	45.00 ± 23.03	34.68 ± 23.77
	similarity	42.38 ± 17.73	32.31 ± 19.33	35.64 ± 18.88	27.76 ± 19.75
	time	196.77 ± 187.85	315.67 ± 321.89	147.21 ± 146.4	132.60 ± 197.25
Qwen3-235B-A22B	bertscore	61.2 ± 3.0	61.6 ± 3.3	60.9 ± 2.7	59.3 ± 3.4
	rouge-l	14.9 ± 4.0	15.8 ± 4.5	14.8 ± 3.7	12.2 ± 3.6
	coverage	52.48 ± 20.79	54.78 ± 21.16	44.54 ± 23.03	30.19 ± 21.96
	similarity	41.68 ± 17.18	43.99 ± 17.54	35.67 ± 18.87	24.10 ± 17.62
	time	103.49 ± 97.30	230.35 ± 271.03	83.06 ± 102.05	158.30 ± 196.35
RuadaptQwen3-32B Instruct-v2	bertscore	57.3 ± 2.9	58.9 ± 3.6	57.7 ± 3.3	55.3 ± 3.3
	rouge-l	11.0 ± 2.4	10.6 ± 3.2	10.7 ± 2.4	7.8 ± 2.1
	coverage	33.12 ± 21.50	33.18 ± 22.83	32.19 ± 22.52	17.72 ± 15.23
	similarity	25.25 ± 16.94	26.21 ± 18.22	24.82 ± 17.74	13.97 ± 12.39
	time	218.30 ± 195.16	379.24 ± 500.40	166.79 ± 164.61	286.35 ± 395.97
tpro	bertscore	59.4 ± 3.0	59.0 ± 4.9	59.5 ± 3.3	58.2 ± 3.7
	rouge-l	13.8 ± 3.1	14.7 ± 4.9	13.5 ± 3.0	11.8 ± 3.9
	coverage	40.27 ± 20.23	40.83 ± 22.42	37.13 ± 20.72	26.03 ± 18.44
	similarity	31.77 ± 16.63	32.60 ± 18.57	29.44 ± 16.83	20.83 ± 15.26
	time	367.32 ± 324.49	592.39 ± 772.19	267.73 ± 253.34	247.59 ± 361.20
RuadaptQwen2.5-7B Lite-Beta	bertscore	55.4 ± 2.9	56.1 ± 4.9	55.8 ± 2.9	54.0 ± 4.0
	rouge-l	8.6 ± 2.5	10.1 ± 3.9	8.7 ± 2.5	7.7 ± 2.8
	coverage	19.66 ± 17.77	24.94 ± 21.08	20.31 ± 17.95	15.51 ± 14.83
	similarity	15.16 ± 14.11	20.03 ± 17.50	15.94 ± 14.39	12.23 ± 12.30
	time	68.86 ± 64.85	126.84 ± 145.74	53.59 ± 47.28	76.66 ± 91.78
yagpt5lite	bertscore	62.5 ± 3.5	61.1 ± 3.8	62.1 ± 3.2	61.5 ± 3.3
	rouge-l	16.9 ± 5.1	15.8 ± 5.1	16.4 ± 4.7	14.3 ± 4.4
	coverage	36.85 ± 19.40	33.17 ± 21.58	31.75 ± 20.06	24.28 ± 16.95
	similarity	29.69 ± 16.43	26.58 ± 18.13	25.60 ± 16.85	19.70 ± 14.29
	time	31.02 ± 28.51	113.34 ± 123.78	27.39 ± 28.05	42.15 ± 56.50

с фильтрацией узлов обеспечивает наилучший компромисс между скоростью и качеством генерации.

5.3. Анализ и сравнение результатов.

Разброс значений метрики QA можно проиллюстрировать на примере работы одной и той же модели (DeepSeek V3) в рамках иерархического метода. В качестве иллюстрации взяты два реферата к произведениям «И грянул гром» и «Кастрюк». В первом случае модель получила высокий результат, ответив на все, кроме одного вопроса; во втором реферате содержались ответы только на два вопроса из одиннадцати, что привело к низкому показателю. На рисунке 2 показаны два реферата. Для краткости в них выделены только основные моменты, которые повлияли на итоговую метрику. Сравнение показывает возможную причину столь значительного расхождения: реферат к рассказу «Кастрюк»

содержит большое количество лирических отступлений и художественных деталей, из-за чего суть произведения сложно уловить и модель отвлекается от фиксации главных фактов, тогда как в "И грянул гром"события изложены последовательно и структурировано, а основные элементы сюжета чётко перечислены, что существенно упрощает задачу поиска важной информации. В текстах выделены жирным шрифтом фрагменты, которые несут в себе важную сюжетную информацию, а подчеркнутый текст - то, что можно было бы опустить.

Название	Текст
И грянул гром	... **Главный герой, Экельс**, азартный и самоуверенный охотник, **платит** огромную сумму **за возможность отправиться на 60 миллионов лет назад, чтобы убить тираннозавра**. Перед путешествием гид Тревис строго предупреждает его о правилах: **ни в коем случае нельзя сходить с антигравитационной Тропы** или вмешиваться в естественный ход событий, так как малейшее нарушение может катастрофически изменить будущее.... Тревис объясняет **хрупкость временного баланса: даже гибель одной мыши способна уничтожить целые виды**, а значит, и изменить историю человечества. **Группа выслеживает тираннозавра**, помеченного красной краской — это знак, что **его убийство не повлияет на будущее**. Однако при виде гигантского хищника **Экельс впадает в панику, сходит с Тропы и случайно раздавливает бабочку**... По возвращении в 2055 ... **мир изменился до неузнаваемости: язык стал грубым, атмосфера — тяжёлой, а вместо умеренного президента Кейта у власти стоит жестокий диктатор Дойчер.** Экельс осознаёт, что **его неосторожность спровоцировала «эффект бабочки»** — раздавленное насекомое вызвало цепь событий, исказивших историю. В отчаянии он умоляет исправить ошибку, но **Тревис, понимая необратимость последствий, поднимает ружьё.** ...
Кастрюк	... Действие рассказа разворачивается в русской деревне ранней весной, где природа пробуждается, но жизнь людей остаётся тяжёлой и однообразной. **Главный герой — старик Семён, прозванный Кастрюком**, — доживает свои дни в одиночестве, **терзаемый воспоминаниями о былой силе** и сожалениями о нынешней немощности. **Когда-то он славился как лучший работник в округе**, но теперь, дряхлый и забытый, вынужден оставаться в стороне, пока односельчане трудятся в поле. Его единственная отрада — внучка Дашка, добрая и впечатлительная девочка, которая прибегает к нему, испугавшись барчуков из соседнего имения Залесное. Кастрюк успокаивает её, и они вместе отправляются за деревню, где старик, любуясь весенней природой, пытается отвлечься от гнетущих мыслей. ... **Лишь к вечеру, уговорив сына отпустить его в ночное (пасти лошадей), Кастрюк обретает краткую радость**. На свободе, среди ребятишек и под звёздным небом, он чувствует себя почти молодым. У пруда кобыла пьёт воду, отражая закат, а сам **старик, глядя на Млечный Путь, шепчет молитву — будто вновь обретает связь с миром и утраченную гармонию**. Но это лишь мимолётное утешение: завтра его снова ждёт беспросветное одиночество и осознание собственной ненужности. ...

Рис. 2. Сравнение лучшего и худшего сгенерированного реферата

Переходя к сравнению между моделями, можно отметить, что в целом DeepSeek V3 показывает лучшие показатели, чем модели меньшей категории, однако, если сравнивать чертежный метод, то в 30% случаев модель RuadaptQwen3-32B-Instruct-v2 показывает лучшие результаты, а tpro в 43%. Для сравнения можно взять реферат по произведению «И грянул гром», созданную с использованием чертежного метода, небольшие вырезки которой приведены на рисунке 3. В то время как реферат, созданный моделью DeepSeek V3 больше похожа на перечисление основных событий через нумерованный список, текст у моделей

Модель	Текст
RuadaptQwen3	"Компания «Сафари во Времени» организует платные экскурсии в прошлое для охоты на динозавров, используя машины времени, способные перемещаться между эпохами. Клиенты обязаны соблюдать строгие правила: следовать по металлической тропе...
tpro	"В тексте главный герой, Экельс, отправляется на сафари во времени с целью убить динозавра Tyrannosaurus rex. Компания, организующая сафари, гарантирует только динозавров и строго запрещает охотникам сходить с Тропы ... Мистер Тревис, проводник сафари, объясняет, что даже уничтожение одной мыши может привести к исчезновению всех её потомков ...
DeepSeek V3	"**Краткое содержание по плану:** 1. **Экельс** — охотник ... 2. **Компания «Сафари во времени»** организует охоту в прошлом ... 3. **Тревис** — проводник, контролирующий экспедицию. ...

Рис. 3. Сравнение моделей при генерации рефератов по чертежному методу

RuadaptQwen3-32B-Instruct-v2 и tpro является связным пересказом текста, раскрывающим все основные события сюжета.

Следует отметить, что лучшего результата удалось добиться именно чертежным методом с помощью большой модели Qwen3-235B-A22B, как было показано в таблице 2. Для сравнения качества рефератов можно взять рассказ «Барбос и Жулька» - в иерархическом методе модель Qwen3-235B-A22B посчитала, что «Жулька» - не собака, а лошадь. Также, например, DeepSeek V3 более строго следует шаблону чертежного метода и вместо связного текста пересказа получается нумерованный список пунктов с ключевыми событиями и главными героями. Однако Qwen3-235B-A22B пишет обычный текст, без списков. Таким образом, чертежный метод без модификаций позволил достичь наилучшего результата с использованием лучшей доступной моделью - Qwen3-235B-A22B.

5.4. Замеры времени.

Проводились первоначальные замеры скорости работы методов на небольших текстах, полученные результаты в секундах (среднее по трём запускам) представлены в таблице Tab. 3. Результаты подтверждают, что модификации позволяют повысить скорость генерации.

Таблица 3. Время генерации рефератов (в секундах) для текста размером 81,049 символов (11 чанков). Усреднено по трём запускам.

Модель	Иерархический	Иерархический с фильтрацией	Чертежный	Чертежный с кластеризацией
DeepSeek V3	237.83	72.42	292.80	268.75
Qwen3-235B-A22B	113.24	39.45	215.63	145.20
RuadaptQwen3-32BInstruct-v2	218.23	72.54	420.95	470.4
tpro	472.23	127.38	421.65	185.94
RuadaptQwen2.5-7B-Lite-Beta	84.64	25.70	103.66	78.99
yagpt5lite	34.17	14.08	99.70	27.26

Интересно отметить, что сверхкрупные модели, такие как Qwen3-235B-A22B и DeepSeek V3, продемонстрировали более высокую скорость работы, чем некоторые модели с размером 32B. Ключевая причина этого заключается в использовании архитектуры MoE (Mixture of Experts): во время генерации активна лишь ограниченная часть параметров (например,

порядка 30 млрд вместо всех 600 млрд), кроме того, такие модели, как правило, дополнительно оптимизированы для повышения производительности.

Заключение

В заключение, был создан первый открытый датасет, объединяющий тексты книг и рефератов к ним с открытого ресурса «Народный Брифли» [4]. В работе предложены два улучшенных подхода к реферированию художественных текстов с использованием LLM: иерархический с фильтрацией и чертёжный с кластеризацией. Иерархический метод с фильтрацией позволяет ускорить генерацию при минимальной потере качества, что делает этот метод пригодным для обработки длинных произведений в условиях ограниченного контекста моделей.

Сравнительный анализ показал, что крупные модели, такие как DeepSeek V3 и Qwen3-235B-A22B, в большинстве случаев обеспечивают более высокое покрытие QA и большую полноту рефератов по сравнению с компактными моделями, особенно в иерархическом и чертёжном методах. Однако для некоторых типов текстов и методов (например, базовый чертёжный) более компактные модели, такие как RuadaptQwen3-32B-Instruct-v2, могут демонстрировать конкурентоспособное качество при меньших вычислительных затратах. Таким образом, выбор модели следует определять исходя из баланса между доступными ресурсами, требованиями к качеству и характером обрабатываемых текстов.

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. BOOKSUM: A Collection of Datasets for Long-form Narrative Summarization / Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal et al. // Findings of the Association for Computational Linguistics: EMNLP 2022 / Ed. by Yoav Goldberg, Zornitsa Kozareva, Yue Zhang. - Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. - . - Pp. 6536-6558. https://aclanthology.org/2022.findings-emnlp.488/.

2. Echoes from Alexandria: A Large Resource for Multilingual Book Summarization / Alessandro Scir'e, Simone Conia, Simone Ciciliano, Roberto Navigli // Findings of the Association for Computational Linguistics: ACL 2023 / Ed. by Anna Rogers, Jordan Boyd-Graber, Naoaki Okazaki. - Toronto, Canada: Association for Computational Linguistics, 2023. - . - Pp. 853-867. https://aclanthology.org/2023.findings-acl.54/.

3. FABLES: Evaluating faithfulness and content selection in book-length summarization

/ Yekyung Kim, Yapei Chang, Marzena Karpinska et al. // First Conference on Language Modeling. - 2024. https://openreview.net/forum?id=YfHxQSoaWU.

4. Народный Брифли. Электронная библиотека кратких пересказов литературных произведений. https://wiki.briefly.ru/ (дата обращения: 30.07.2025).

5. Библиотека художественных произведений. https://librusec.org// (дата обращения: 30.07.2025).

6. Wu J. et al. Recursively Summarizing Books with Human Feedback //arXiv e-prints. - 2021. - C. arXiv: 2109.10862.

7. Text-Blueprint: An Interactive Platform for Plan-based Conditional Generation / Fantine Huot, Joshua Maynez, Shashi Narayan et al. // Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations / Ed. by Danilo Croce, Luca Soldaini. - Dubrovnik, Croatia: Association for Computational Linguistics, 2023. - . - Pp. 105-116. https: //aclanthology.org/2023.eacl-demo.13/.

8. ROUGE. Lin C. Y. Rouge: A package for automatic evaluation of summaries //Text summarization branches out. - 2004. - C. 74-81.

9. BERTScore. BUCKLEY C. Evaluating Evaluation Measure Stability //ACM SIGIR 2000 Proceedings. - 2000.

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

11. 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.

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

13. 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).

14. YandexGPT 5 с режимом рассуждений // Яндекс URL: https://ya.ru/ai/gpt?ysclid=mal9jrssc8906806775 (дата обращения: 30.07.2025).