УДК 004.912; 004.052 Обзорная статья

https://doi.org/10.32603/2071-8985-2025-18-7-60-68

# Сравнительный анализ методов извлечения ключевых слов без предварительного обучения

**П. В. Корытов**

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

thexcloud@gmail.com

**Аннотация.** Рассмотрена задача извлечения ключевых слов из текстов в условиях «холодного старта» – без аннотированных данных для предварительного обучения. Проведен сравнительный анализ трех категорий существующих методов: классические алгоритмы (YAKE, TextRank, SingleRank, TopicRank, PositionRank, FRAKE), методы на основе BERT (KeyBERT, KBIR-Inspec, ансамблевый метод с KBIR-Inspec и WikiNEuRal) и открытые большие языковые модели (LLM: llama3.1, qwen2.5, t-pro). Также предложена методика автоматизированной подготовки бенчмарков для оценки качества извлечения ключевых слов с помощью проприетарной LLM Claude3.5 Haiku. Оценка методов производится по метрикам «жесткого» и «мягкого» $F_1$-score для разного количества ключевых слов. На двух собственных бенчмарках лучшие результаты показала открытая LLM t-pro с 3-shot промптом ($F_1$ = 0.40, $F_1$ = 0.35) даже без проведения дообучения под предметную область, однако это также и самый требовательный по ресурсам метод (~22 Гбайт VRAM). Более «легкие» методы показывают худшие результаты.

**Ключевые слова:** ключевые слова, BERT, подготовка бенчмарков, промпт-инжиниринг, большие языковые модели

**Для цитирования:** Корытов П. В. Сравнительный анализ методов извлечения ключевых слов без предварительного обучения // Изв. СПбГЭТУ «ЛЭТИ». 2025. Т. 18, № 7. С. 60–68. doi: 10.32603/2071-8985-2025-18-7-60-68.

Review article

# Comparative Analysis of Unsupervised Keyword Extraction Methods

**P. V. Korytov**

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

thexcloud@gmail.com

**Abstract.** The paper examines the task of keyword extraction in «cold start» conditions – without annotated data for preliminary training. Three categories of existing methods are compared: classical algorithms (YAKE, TextRank, SingleRank, TopicRank, PositionRank, FRAKE), BERT-based methods (KeyBERT, KBIR-Inspec, ensemble method with KBIR-Inspec and WikiNEuRaL) and open weights large language models (LLM: llama3.1, qwen2.5, t-pro). Additionally, a methodology for automated keyword extraction benchmark preparation is proposed, making use of the proprietary LLM Claude3.5 Haiku. The methods are evaluated using «hard» and «sof» $F_1$-score metrics for different numbers of keywords. On two custom benchmarks, the open LLM t-pro with a 3-shot prompt showed the best results ($F_1$ = 0.40, $F_1$ = 0.35), even without domain-specific fine-tuning; however, it is also the most resource-intensive method (~22 G VRAM). «Lighter» methods show inferior results.

**Keywords:** keywords, BERT, benchmark preparation, prompt engineering, large language models

**For citation:** Korytov P. V. Comparative Analysis of Unsupervised Keyword Extraction Methods // LETI Transactions on Electrical Engineering & Computer Science. 2025. Vol. 18, no. 7. P. 60–68. doi: 10.32603/2071-8985-2025-18-7-60-68.

**Введение.** Задача извлечения ключевых слов (Keyword Extraction) – задача сопоставления строке набора более коротких подстрок, характеризующих исходную строку.

Эта задача важна в обработке неструктурированных наборов текстовых данных (корпусов текстов), так как использование ключевых слов как «тегов» может задать на корпусе определенную структуру. Например, можно найти все документы, в которых упоминается заданное ключевое слово или похожие на него; или приблизительно сравнить два документа, сравнив их ключевые слова.

Кроме того, набор ключевых слов – это более емкое представление исходного текста, которое будет иметь одинаковую форму для любых входных текстов (например, текстов разной длины). Это открывает возможность агрегации разнородных текстов в единое представление.

В данной статье произведен сравнительный анализ методов извлечения ключевых слов, не требующих обучения на аннотированных данных для начала работы. Это ограничение порождает две проблемы. Во-первых, без аннотированных данных сложно оценить качество метода, поэтому здесь предложена методика автоматизированной подготовки бенчмарка. Во-вторых, без обучаемости нет возможность улучшения результатов за счет обратной связи. Поэтому в способе оценки также учтена возможность постобработки результатов метода для улучшения качества.

**Способы оценки извлечения ключевых слов.** Как правило, оценка качества извлечения ключевых слов осуществляется с помощью специальных наборов данных (бенчмарков). Бенчмарки обычно представляют собой набор пар вида:

– исходная строка (документ);

– список ключевых слов, которые нужно извлечь.

Качество работы метода определяется сравнением ключевых слов, извлеченных оцениваемым методом, с целевым набором ключевых слов.

Особенность таких бенчмарков в том, что ключевые слова из одной строки, как правило, можно извлечь разными способами одинаково хорошо. Поэтому этим бенчмаркам присуща определенная неточность, наблюдаемая, например, в [1].

***Методика автоматизированной подготовки бенчмарка.*** Создание бенчмарков вручную требует временных затрат и экспертной оценки. Поэтому предлагается следующая методика автоматизированной подготовки бенчмарка с помощью больших языковых моделей.

*Шаг 1*. Сведение всех частей документа в одну строку естественного текста, удаление переносов строк, спецсимволов вроде перечислений и т. п.

*Шаг 2*. Удаление похожих документов при помощи кластеризации; этот шаг необходим при наличии в корпусе незначительно отличающихся документов, например, с переставленными предложениями или с поправленными опечатками.

Для этого предлагается применение следующей последовательности методов:

• Snowball Stemmer [2].

• TF-IDF (**TF** – term frequency, **IDF** – inverse document frequency) (ngram_range=(1, 1)).

• DBSCAN (Density-based spatial clustering of applications with noise) (min_samples=1, eps=0.5) [3].

Необходимое значение eps нужно подобрать экспериментально в зависимости от требуемой схожести документов.

*Шаг 3*. Извлечение кандидатов в эталонные ключевые слова с помощью LLM (large language model).

Для этого рационально использовать LLM с достаточными способностями (например, > 80 % на бенчмарке MMLU, *Massive Multitask Language Understanding*), но с меньшей стоимостью токена, чем state-of-the-art модели (GPT-4o, Claude 3.5 Sonnet). В данной статье выбрана модель Claude 3.5 Haiku, в качестве альтернатив можно рассмотреть gpt-4o-mini или DeepSeek-V3. Доступ к LLM в данной работе осуществляется с помощью сервиса OpenRouter (https://openrouter.ai/).

Также, для обеспечения консистентности вывода LLM между разными документами необходимо использовать multi-shot-промпт [4] с тремя-четырьмя примерами. При этом важно, чтобы каждое ключевое слово из примера дословно встречалось в тексте (по условию задачи извлечения ключевых слов).

Multi-shot-промпт для извлечения ключевых слов:

Extract 5–10 keywords from each of the following texts. Each text describes a university subject (in Russian, with occasional English acronyms).

Rules for keyword extraction:

– All keywords must occur in the text, strictly the same form (same declension, etc.).

– Put each keyword on a new line.

– List the most descriptive keywords first.

– Avoid very general keywords (e.g., управление, модели, качество).

– Do not start keywords with a number.

– Output only the keywords, nothing else.

Text N:
<текст>
Keywords N:
<ключевое-слово-1>
...
<ключевое-слово-N>

Now, extract keywords from the following text using the same rules.

*Шаг 4*. Обработка кандидатов в эталонные ключевые слова. Этот шаг необходим по нескольким причинам:

• LLM может отказаться от извлечения ключевых слов при несогласии с тематикой текста;

• LLM иногда выводят комментарии к результатам;

• LLM подвержены галлюцинациям;

• LLM может потерять форму слова при извлечении.

Во всех случаях проблема решается проверкой того, что ключевое слово действительно – подстрока исходного текста. Если в итоге при этом остается меньше, чем $N$ ключевых слов (выбрано $N = 3$), можно попробовать повторить шаг 3 еще $M$ раз (выбрано $M = 3$). Если после $M$ повторений все равно не получается извлечь ключевые слова, данный документ нужно отбросить.

***Используемые бенчмарки.*** Здесь применялись следующие бенчмарки. Во-первых, бенчмарк Inspec [5], доступный в библиотеке ake-dataset (GitHub: boundfl/ake-datasets) в формате Stanford CoreNLP.

Пример одной записи бенчмарка Inspec:

Twenty years of the literature on acquiring out-of-print materials This article reviews the last two-and-a-half decades of literature on acquiring out-of-print materials to assess recurring issues and identify changing practices. The out-of-print literature is uniform in its assertion that libraries need to acquire o.p. materials to replace worn or damaged copies, to replace missing copies, to duplicate copies of heavily used materials, to fill gaps in collections, to strengthen weak collections, to continue to develop strong collections, and to provide materials for new courses, new programs, and even entire new libraries.

Out-of-print materials; recurring issues; changing practices; out-of-print books; library materials; acquisition.

Во-вторых, по описанной методике составлено два бенчмарка из рабочих программ (РП) СПбГЭТУ «ЛЭТИ». Одна РП представляет собой многостраничный документ на русском языке, описывающий содержимое дисциплины – аннотацию, цели и задачи, основные темы, примеры вопросов к экзаменам и т. п. В одном учебном плане (УП) бакалавриата содержится несколько десятков РП.

Бенчмарки составлены из разных разделов документов:

1. Содержание – тезисное описание тем дисциплины. Извлечение ключевых слов из такого текста представляет трудность из-за его сжатости, но ключевые слова часто уже находятся в канонической форме.

2. Название дисциплины, аннотация, цели и задачи – более общее описание дисциплины. Как правило, это менее тезисный текст, но в то же время менее конкретный и более вариативный морфологически.

Сводка по используемым бенчмаркам приведена в табл. 1.

***Мягкая/жесткая оценка $F_1$.*** В данной статье сравнение извлеченных и целевых ключевых слов производится двумя способами:

1. *Жесткий* (strict) – в качестве сравниваемых множеств берутся ключевые слова целиком.

2. *Мягкий* (lenient) – в качестве сравниваемых множеств берутся токены ключевых слов.

Затем по обоим множествам считается матрица ошибок и $F_1$-score:

$$\mathrm{TP} = \left|\mathrm{source} \cap \mathrm{target}\right|,$$
$$\mathrm{FP} = \left|\mathrm{source} \setminus \mathrm{target}\right|,$$
$$\mathrm{FN} = \left|\mathrm{target} \setminus \mathrm{source}\right|,$$

где TP – истинно положительный результат (*true positive*, выделено верное ключевое слово); FP – ложноположительный результат (*false positive*,

*Табл. 1*. Используемые бенчмарки
*Tab. 1*. Used benchmarks

| Бенчмарк | Язык | Количество документов | Средняя длина текста | Среднее количество ключевых слов |
|---|---|---|---|---|
| Inspec | Английский | 1500 | 871.9 | 9.8 |
| РП (содержание) | Русский | 1852 | 3138.8 | 6.2 |
| РП (содержание) | Русский | 1660 | 2007.7 | 4.8 |

выделено неверное ключевое слово); FN – ложно-отрицательный результат (*false negative*, верное ключевое слово не выделено). Тогда

$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}.$$

Запись вида $F_1$@N означает, что из множества извлеченных ключевых слов забрано топ-$N$ лучших (способ оценки зависит от метода).

Таким образом, «жесткая» оценка требует полного совпадения ключевых слов, «мягкая» – частичного. Данный способ выбран исходя из разрабатываемого подхода – большая разница между «жесткой» и «мягкой» оценками показывает, что метод извлекает ключевые слова, близкие к нужным, но немного отличающиеся от них. Теоретически оценку такого метода можно улучшить путем постобработки.

**Обзор существующих методов.** ***Классические методы.*** Классические методы представляют собой методы извлечения ключевых слов, основанные на обработке текстов с помощью эвристик.

Один из таких методов – статистический метод YAKE (*Yet Another Keyword Extraction method*) [1]. Суть метода представлена в следующем уравнении:

$$S(t) = \frac{T_{\mathrm{Rel}} T_{\mathrm{Pos}}}{T_{\mathrm{Case}} + \frac{TF_{\mathrm{Norm}}}{T_{\mathrm{Rel}}} + \frac{T_{\mathrm{Sent}}}{T_{\mathrm{Rel}}}},$$

$$S(kw) = \frac{\prod_{t \in kw} S(t)}{KF(kw)\left(1 + \sum_{t \in kw} S(t)\right)}.$$

Эвристики – это связность терма $t$ с контекстом $T_{\mathrm{Rel}}$, положение терма в документе $T_{\mathrm{Pos}}$, характеристика регистра букв $T_{\mathrm{Case}}$, нормализованная частота терма $TF_{\mathrm{Norm}}$, частота появления в разных предложениях $T_{\mathrm{Sent}}$. Один или несколько термов формируют кандидата в ключевое слово $kw$, который взвешивается указанным образом с учетом частоты встречаемости кандидата в тексте $KF(kw)$.

Также к этой категории относится набор графовых методов, основанных на алгоритме PageRank. Один из них, метод TextRank [6], строит неориентированный граф, где вершины – слова документа, а ребра с бинарным весом отражают совместную встречаемость слов в окне фиксированного размера. SingleRank [7] использует взвешенные ребра, где вес равен частоте совместной встречаемости ключевых слов. TopicRank [8] группирует схожие кандидаты в ключевые фразы в тематические кластеры, строит полный граф с кластерами-вершинами, а затем выбирает по одной фразе из лучших кластеров. PositionRank [9] учитывает позиции всех вхождений слов в документе, модифицируя начальное распределение вероятностей в PageRank пропорционально позициям слов, предполагая, что слова в начале документа более важны. Все методы используют модификацию PageRank для ранжирования вершин и извлечения ключевых слов из текста.

Еще один представитель этой категории – FRAKE (*Fusional Real-time Automatic Keyword Extraction*) [10], метод извлечения ключевых слов, комбинирующий статистические и графовые подходы. Он реализует следующий подход:

1. Предобработка (удаление стоп-слова, токенизация).

2. Параллельно:

а) извлечение графовых признаков –> понижение размерности с PCA (Principal component analysis) –> «графовый рейтинг» слов;

б) извлечение текстовых признаков –> «текстовый рейтинг» слов.

3. Выбор кандидатов в ключевые слова с помощью HUMP (High Utility Pattern Mining).

4. Выбор топ-$N$ кандидатов.

Преимуществами всех методов данной категории является их быстродействие и низкие требования к ресурсам (относительно прочих рассматриваемых методов), а также независимость от предметной области и минимальная зависимость от языка. Общий недостаток – отсутствие возможности дообучения.

В качестве источника реализаций перечисленных методов, кроме FRAKE, выбрана библиотека PKE (*Python Keyword Extraction module*) [11].

***Методы на основе BERT.*** *KeyBERT*. Среди рассмотренных методов на основе BERT (*Bidirectional Encoder Representations from Transformer*) извлечения ключевых слов KeyBERT (GitHub: MaartenGr/KeyBERT) – необучаемый метод, использующий векторные представления текстов, произведенные нейросетями (эмбеддинги). Он реализует следующий подход:

1. Извлечение кандидатов в ключевые слова с помощью метода CountVectorizer.

По сути, это извлечение из документа всевозможных $n$-gram (где $n$ настраивается) с поправкой на стоп-слова.

2. Получение эмбеддингов для всех кандидатов и для исходного текста.

3. Вычисление косинусного расстояния между эмбеддингами кандидатов и исходным текстом.

4. Извлечение топ-*N* кандидатов.

Кандидаты извлекаются одним из трех способов:

1. Выбор топ-*N* с минимальным расстоянием (по умолчанию).

2. MaxSum: выбор топ-*NR* (по умолчанию *NR* = 20) кандидатов, затем выбор подмножества из *N* кандидатов с минимальной суммой попарной схожести между кандидатами.

3. Использование максимальной предельной релевантности (MMR) с исходным текстом в качестве запроса (*q*).

$$\mathrm{MMR} = \arg\max_{d_i \in D \setminus R} \left[\lambda \mathrm{Sim}_1(D - i, q) - \right.$$
$$\left. - (1-\lambda) \max_{d_j \in R} \mathrm{Sim}_2(d_i, d_j)\right],$$

где $D$ – все кандидаты; $R$ – выбранные кандидаты; $q$ – запрос; $\mathrm{Sim}_1$ – схожесть между запросом и кандидатом; $\mathrm{Sim}_2$ – схожесть между двумя кандидатами; $\lambda$ – параметр, балансирующий релевантность (первый член) и разнообразие (второй член); $d_i$ – документ в $D \setminus R$; $d_j$ – документ в $R$.

Данный метод не зависит от конкретного способа извлечения эмбеддингов; по умолчанию предлагается использовать модель all-MiniLM-L6-v2 [12].

«Узкое место» данного метода – шаг 2, так как получение эмбеддингов от всех *n*-gram исходного текста может занимать много времени.

Также отдельный вопрос – выбор такого способа получения эмбеддингов, для которого в пространстве признаков похожие тексты и их ключевые слова будут размещены рядом. Это может быть сложно из-за разницы в размере ключевого слова и текста. Для русского языка в качестве подходящего способа можно использовать paraphrase-multilingual-mpnet-base-v2 [13]. Кроме того, недостатком данного метода служит отсутствие очевидного пути к дообучению при потреблении ресурсов, сравнимом с другими нейросетевыми методами.

*KBIR-Inspec и дообучение BERT*. В противоположность сложной инженерии признаков, предлагаемого KeyBERT, возможен и более простой подход – использование нейросетей для классификации токенов. На вход такой нейросети поступает токенизированный текст, на выходе – отношения токенов к классам (часть ключевого слова или нет).

KBIR-Inspec – пример такой модели. Он представляет собой англоязычную модель KBIR [13], дообученную на наборе данных Inspec, рассматриваемом в данной работе. Код обучения недоступен, но веса модели находятся в открытом доступе.

В карточке модели приведен результат $F_1$@10=0.588, но этот результат явно получен на train-части Inspec, на котором модель проходила обучение. На тестовой части в данной работе эта модель показала $F_1$@10=0.34 (см. далее).

Недостаток данной категории методов в длине контекста, ограничивающий максимальный размер обрабатываемого текста, в данном случае – 512 токенами; также особенностью является зависимость от языка. Для получения лучших результатов такими методами требуется дообучение нейросети на целевом наборе данных; использование без дообучения затруднено.

*Ансамбль KBIR-Inspec и WikiNEuRal*. Метод, описанный в [14], – это ансамбль нескольких открытых нейросетевых методов для извлечения ключевых слов на русском языке:

• упомянутый KBIR-Inspec, обернутый в русско-английский и англо-русский переводчик с обработкой случаев плохого перевода с помощью нечеткого поиска;

• модель WikiNEuRal – модель NER (*Named Entity Recognition*, извлечение именованных сущностей), обученную на Wikipedia [15]. Модель поддерживает русский язык.

Результаты методов объединяются с помощью word2vec [16] и DBSCAN. Извлеченные ключевые слова также предлагается нормализовать с помощью промпт-инжиниринга; для целей сравнения в этой публикации данный шаг опущен.

Преимущество метода состоит в использовании готовых state-of-the-art нейросетевых моделей, что не требует обучения для начала работы. Недостаток – использование модели NER – в рамках задачи распознавания именованных сущностей их можно распознать очень много (например, перечисление персоналий или организаций через запятую) или не распознать ни одной. Кроме того, формат использования моделей затрудняет их дообучение на обратной связи.

***Использование LLM.*** Последняя рассматриваемая категория методов – использование больших языковых моделей (LLM) с помощью промпт-инжиниринга.

LLM были выбраны среди моделей с открытыми весами, которые на момент написания ста-

тьи показывали лучшие результаты и могли работать на оборудовании потребительского уровня. Перечень используемых LLM приведен в табл. 2.

*Табл. 2*. Используемые LLM
*Tab. 2*. Used LLMs

| Модель | Количество весов, байт | Квант | Описание |
|---|---|---|---|
| llama3.1 | 8 | Q4_0 | Базовая модель |
| qwen2.5 | 32 | Q4_K_M | Базовая модель |
| t-pro | 32 | Q4_K_M | Дообучение qwen2.5 на русском языке |

Все модели использованы через решение ollama. Для всех решений использован multi-shot-промпт через сообщения.

Multi-shot-промпт для извлечения ключевых слов с локальными LLM:

Below is a text with the contents a university subject (in Russian, with occasional English acronyms). Extract from 5 to 10 keywords from it.

All keywords must occur in the text, strictly the same form (same declension, etc.). Output the keywords and nothing else. Put each keyword on a new line.

Put the best (i.e. most descriptive) keywords first. Do not write very general keywords (управление, модели, качество, etc.) Do not start keywords with a number.

Для моделей qwen2.5 и t-pro также опробован перевод промпта на русский язык.

Исследование эффективности LLM на Inspec не имеет смысла, так как весь Inspec многократно содержится в обучающем наборе (проблема контаминации бенчмарков [17]). Оценка на собственных бенчмарках, тем не менее, проведена.

LLM, теоретически, можно использовать без дообучения, но для получения лучших результатов рассмотренными моделями и для реализации любой обратной связи дообучение необходимо. В то же время, и использование, и дообучение LLM предъявляют самые высокие требования к ресурсам из рассматриваемых методов.

**Сравнение методов.** Вышеописанные методы оценены способами, указанными в разделе «Способы оценки извлечения ключевых слов». Результаты приведены в табл. 3–5.

*Табл. 3*. Сравнение методов на test-части Inspec
*Tab. 3*. Methods comparison on the test section of Inspec

| Метод | Качество работы метода (раздел «Мягкая/жесткая оценка $F_1$»») | | | | | |
|---|---|---|---|---|---|---|
| | lenient-f1@3 | strict-f1@3 | lenient-f1@5 | strict-f1@5 | lenient-f1@10 | strict-f1@10 |
| Теоретический максимум | 0.844133 | 0.8362 | 0.833845 | 0.827829 | 0.85933 | 0.82699 |
| FirstPhrases | **0.634397** | **0.525867** | **0.597314** | **0.442110** | 0.518554 | 0.27060 |
| YAKE | 0.514524 | 0.329200 | 0.494965 | 0.258810 | 0.454414 | 0.12489 |
| PositionRank | 0.567407 | 0.462667 | 0.545773 | 0.399597 | 0.520984 | 0.27335 |
| SingleRank | 0.547963 | 0.433733 | 0.544652 | 0.382951 | **0.542878** | 0.26812 |
| TextRank | 0.504271 | 0.389067 | 0.509347 | 0.350273 | 0.531813 | 0.29583 |
| TopicRank | 0.500399 | 0.402667 | 0.484969 | 0.345257 | 0.487765 | 0.29877 |
| FRAKE | 0.310381 | 0.078667 | 0.357248 | 0.066867 | 0.42287 | 0.07329 |
| KeyBERT | 0.390640 | 0.099067 | 0.431969 | 0.082959 | 0.47131 | 0.02865 |
| KBIR-Inspec | 0.464349 | 0.365600 | 0.498070 | 0.363543 | 0.534534 | **0.36833** |

*Табл. 4*. Сравнение методов на test-части содержаний РП
*Tab. 4*. Methods comparison on the test section of the subject content benchmark

| Метод | Качество работы метода | | | | | |
|---|---|---|---|---|---|---|
| | lenient-f1@3 | strict-f1@3 | lenient-f1@5 | strict-f1@5 | lenient-f1@10 | strict-f1@10 |
| Теор. макс | 1 | 1 | 1 | 1 | 1 | 1 |
| FirstPhrases | 0.185620 | 0.142189 | 0.155404 | 0.098032 | 0.129637 | 0.059824 |
| YAKE | 0.205432 | 0.120590 | 0.218999 | 0.094831 | 0.242565 | 0.063760 |
| PositionRank | 0.217197 | 0.151008 | 0.196788 | 0.115146 | 0.178398 | 0.078370 |
| SingleRank | 0.160626 | 0.072534 | 0.174700 | 0.066682 | 0.172997 | 0.049818 |
| TextRank | 0.131853 | 0.046976 | 0.151253 | 0.044924 | 0.155119 | 0.034436 |
| TopicRank | 0.264436 | 0.201944 | 0.263531 | 0.177514 | 0.246530 | 0.125882 |
| FRAKE | 0.148182 | 0.065515 | 0.169082 | 0.051296 | 0.188297 | 0.032506 |
| KeyBERT | 0.168484 | 0.039597 | 0.162726 | 0.028354 | 0.175914 | 0.018056 |
| Метод [14] | 0.386004 | 0.349316 | 0.372862 | 0.317378 | 0.317201 | 0.240244 |
| llama3.1:8b | 0.346276 | 0.2946 | 0.344841 | 0.264293 | 0.336083 | 0.233588 |
| qwen2.5:32b, en | 0.512008 | 0.484665 | 0.498874 | 0.450624 | 0.459907 | 0.391896 |
| qwen2.5:32b, ru | **0.530517** | **0.501224** | **0.50038** | **0.465582** | 0.454246 | 0.399559 |
| t-pro | 0.493245 | 0.463067 | 0.479257 | 0.434184 | **0.462031** | **0.404155** |

*Табл. 5*. Сравнение методов на test-части аннотаций РП
*Tab. 5*. Methods comparison on the test section of the subject annotation benchmark

| Метод | Качество работы метода | | | | | |
|---|---|---|---|---|---|---|
| | lenient-f1@3 | strict-f1@3 | lenient-f1@5 | strict-f1@5 | lenient-f1@10 | strict-f1@10 |
| Теор. макс | 1 | 1 | 1 | 1 | 1 | 1 |
| FirstPhrases | 0.285536 | 0.189324 | 0.255510 | 0.133374 | 0.234768 | 0.084191 |
| YAKE | 0.464515 | 0.328798 | 0.401584 | 0.232576 | 0.380643 | 0.148745 |
| PositionRank | 0.436466 | 0.348264 | 0.370380 | 0.257955 | 0.319433 | 0.168897 |
| SingleRank | 0.242640 | 0.075778 | 0.248225 | 0.061274 | 0.235348 | 0.041605 |
| TextRank | 0.211008 | 0.042745 | 0.225633 | 0.037651 | 0.221279 | 0.025958 |
| TopicRank | 0.260466 | 0.152318 | 0.260269 | 0.122649 | 0.266146 | 0.082713 |
| FRAKE | 0.21845 | 0.035737 | 0.238195 | 0.023695 | 0.266341 | 0.014721 |
| KeyBERT | 0.29937 | 0.088622 | 0.293794 | 0.067035 | 0.294250 | 0.044443 |
| Метод [14] | 0.45807 | 0.397917 | 0.430167 | 0.334121 | 0.402811 | 0.267706 |
| llama3.1:8b | 0.303488 | 0.178766 | 0.312828 | 0.166461 | 0.318388 | 0.162300 |
| qwen2.5:32b, en | 0.460902 | 0.396715 | 0.449915 | **0.370786** | 0.443396 | 0.349807 |
| qwen2.5:32b, ru | **0.488651** | **0.402965** | **0.467524** | 0.360064 | **0.466573** | 0.334850 |
| t-pro | 0.427627 | 0.369712 | 0.412781 | 0.354791 | 0.415065 | **0.352109** |

*Табл. 6*. Оценка потребления ресурсов
*Tab. 6*. Resource consumption estimation

| Метод | *S* (Inspec) | *S* (содержание РП) | *S* (аннотации РП) | Макс. RAM, Мбайт | Макс. GPU, Мбайт |
|---|---|---|---|---|---|
| YAKE | 0.050 | 0.102 | 0.059 | 331 | – |
| PositionRank | 0.048 | 0.098 | 0.057 | 323 | – |
| SingleRank | 0.048 | 0.097 | 0.056 | 345 | – |
| TextRank | 0.048 | 0.094 | 0.055 | 345 | – |
| TopicRank | 0.049 | 0.129 | 0.059 | 331 | – |
| FRAKE | 0.328 | 2.278 | 2.856 | 163 | – |
| KeyBERT | 2.542 | 2.597 | 2.592 | 974 | 786 |
| KBIR-Inspec | 0.009 | – | – | 1557 | 2278 |
| Метод [14] | – | 1.847 | 1.010 | 1731 | 2628 |
| llama3.1:8b | – | 2.307 | 1.713 | 384 | 6234 |
| t-pro | – | 9.250 | 7.37 | 359 | 21792 |

В табл. 6 приведено сравнение потребления ресурсов разными методами. Показатель *S* – среднее время обработки одного документа на данном бенчмарке; значения RAM и GPU выбраны как максимальные зафиксированные.

**Обсуждение результатов и выводы.** Из проведенного сравнения можно сделать следующие выводы.

Во-первых, среди не-LLM методов на всех бенчмарках лучшие результаты показали методы, связанные с дообучением BERT (KBIR-Inspec для Inspec и он же, вложенный в переводчик, для РП).

Во-вторых, среди ненейросетевых методов рассмотренные классические методы показывают относительно неплохие результаты по lenient-f1@10 на нетезисных текстах:

• на Inspec SingleRank – лучший по lenient-f1@10;

• на аннотации РП YAKE на 0.08 единиц хуже по lenient-f1@10, чем qwen2.5.

Однако при оценке по strict-f1@10 разница более существенна. Это значит, что данные методы извлекают ключевые слова, похожие на требуемые, но не совпадающие полностью – например, с лишними или отсутствующими токенами.

Таким образом, рационально в дальнейшей работе исследовать возможность дообучения BERT для постобработки выводов более простых методов, дообучение которых невозможно.

В-третьих, методы с LLM весом 32 байт даже без дообучения показывают лучшие результаты на РП, но метод с LLM весом 8 байт уступает методу на основе BERT, при этом потребляя больше ресурсов.

Также интересно, что дообучение LLM на русском языке не оказывает существенного влияния на результаты.

Хороший результат LLM может быть связан с тем, что эталонные ключевые слова сами по себе сгенерированы через LLM большего размера. Возможно, разные LLM генерируют похожие слова за счет внутренних особенностей метода или за счет обучения всех LLM на приблизительно одних и тех же корпусах данных.

Таким образом, в дальнейшей работе разумно провести оценку перечисленных методов на бенчмарках, составленных экспертами-людьми.

## Список литературы

1. YAKE! Keyword extraction from single documents using multiple local features / R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt // Information Sci. 2020. Vol. 509. P. 257–289. doi: 10.1016/j.ins.2019.09.013.

2. Lovins J. B. Development of a stemming algorithm // Mech. Transl. Comput. Linguistics. 1968. Vol. 11, № 1–2. P. 22–31.

3. A density-based algorithm for discovering clusters in large spatial databases with noise / M. Ester, H. Kriegel, J. Sander, X. Xu // Proc. of 2nd Intern. Conf. on Knowledge Discovery and Data Mining (KDD–96). 1996. P. 226–231.

4. Berryman J., Ziegler A. Prompt engineering for LLMs: the art and science of building large language model-based applications. Sebastopol, CA: O'Reilly Media, 2025. 280 p.

5. Hulth A. Improved automatic keyword extraction given more linguistic knowledge // Proc. of the conf. on Empirical methods in natural language processing (EMNLP 2003). Sapporo, Japan: Association for Computational Linguistics, 2003. P. 216–223.

6. Mihalcea R., Tarau P. TextRank: Bringing order into text // Proc. of the 2004 conf. on empirical methods in natural language. Barcelona, Spain: Association for Computational Linguistics, 2004. P. 404–411.

7. Wan X., Xiao J. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction // Proc. of the 22nd Intern. Conf. on Computational Linguistics (Coling 2008). Manchester, UK: Association for Computational Linguistics, 2008. P. 969–976.

8. Bougouin A., Boudin F., Daille B. TopicRank: Graph-based topic ranking for keyphrase extraction // Proc. of the sixth intern. joint conf. on natural language processing (IJCNLP). Nagoya, Japan: Asian Federation of Natural Language Proc., 2013. P. 543–551.

9. RoFormer: Enhanced Transformer with Rotary Position Embedding / J. Su, Y. Lu, S. Pan, A. Murthadha, B. Wen, Y. Liu // Neurocomputing. 2024. Vol. 568. P. 1–12. doi: 10.1016/j.neucom.2023.127063.

10. Zehtab-Salmasi A., Feizi-Derakhshi M.-R., Balafar M.-A. FRAKE: Fusional Real-time Automatic Keyword Extraction. 2021. P. 1–12. URL: https://arxiv.org/abs/2104.04830 (дата обращения: 28.01.2025).

11. Boudin F. PKE: an open source Python-based keyphrase extraction toolkit // Proc. of COLING the 26th Intern. Conf. on computational linguistics: System demonstrations. Osaka, Japan: Association for Computational Linguistics, 2016. P. 69–73.

12. Reimers N., Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation // Proc. of the Conf. on Empirical Methods in Natural Language Proc. (EMNLP 2020). Punta Cana, Dominican Republic: Association for Computational Linguistics, 2020. P. 4512–4525. doi: 10.18653/v1/2020.emnlp-main.365.

13. Learning Rich representation of keyphrases from text / M. Kulkarni, D. Mahata, R. Arora, R. Bhowmik // Findings of the Association for Computational Linguistics: NAACL. Seattle, United States: Association for Computational Linguistics, 2022. P. 891–906. doi: 10.18653/v1/2022.findings-naacl.67.

14. Sorochina M. V., Korytov P. V., Kholod I. I. Application of neural network methods for keyword extraction for compiling student's resume based on work programs // Proc. XXVI Intern. Conf. on Soft Comp. and Measurements (SCM–2023). Saint-Petersburg, Russia: IEEE, 2023. P. 186–189. doi: 10.1109/SCM58628.2023.10159061.

15. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER / S. Tedeschi, V. Maiorca, N. Campolungo, F. Cecconi, R. Navigli // Findings of the Association for Computational Linguistics: EMNLP 2021. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021. P. 2521–2533.

16. Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // Proc. of the Intern. Conf. on Learning Representations (ICLR 2013). Scottsdale, USA: Association for Computational Linguistics, 2013. P. 1–12. doi: 10.48550/arXiv.1301.3781.

17. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark / O. Sainz, J. Campos, I. García-Ferrero, J. Etxaniz, O. Lacalle, E. Agirre // Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore: Association for Computational Linguistics, 2023. P. 10776–10787. doi: 10.18653/v1/2023.findings-emnlp.722.

## Информация об авторе

**Корытов Павел Валерьевич** – аспирант кафедры информационных систем, ассистент кафедры математического обеспечения и применения ЭВМ, ведущий программист отдела разработки цифровых сервисов. СПбГЭТУ «ЛЭТИ».
E-mail: thexcloud@gmail.com
https://orcid.org/0000-0001-5534-5389

## References

1. YAKE! Keyword extraction from single documents using multiple local features / R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt // Information Sci. 2020. Vol. 509. P. 257–289. doi: 10.1016/j.ins.2019.09.013.

2. Lovins J. B. Development of a stemming algorithm // Mech. Transl. Comput. Linguistics. 1968. Vol. 11, № 1–2. P. 22–31.

3. A density-based algorithm for discovering clusters in large spatial databases with noise / M. Ester, H. Kriegel,

J. Sander, X. Xu // Proc. of 2nd Intern. Conf. on Knowledge Discovery and Data Mining (KDD–96). 1996. P. 226–231.

4. Berryman J., Ziegler A. Prompt engineering for LLMs: the art and science of building large language model-based applications. Sebastopol, CA: O'Reilly Media, 2025. 280 p.

5. Hulth A. Improved automatic keyword extraction given more linguistic knowledge // Proc. of the conf. on Empirical Methods in Natural Language Proc. (EMNLP 2003). Sapporo, Japan: Association for Computational Linguistics, 2003. P. 216–223.

6. Mihalcea R., Tarau P. TextRank: Bringing order into text // Proc. of the 2004 Conf. on Empirical Methods in Natural Language. Barcelona, Spain: Association for Computational Linguistics, 2004. P. 404–411.

7. Wan X., Xiao J. CollabRank: Towards a Collaborative approach to single-document keyphrase extraction // Proc. of the 22nd Intern. Conf. on Computational Linguistics (Coling 2008). Manchester, UK: Association for Computational Linguistics, 2008. P. 969–976.

8. Bougouin A., Boudin F., Daille B. TopicRank: Graph-based topic ranking for keyphrase extraction // Proc. of the sixth intern. joint conf. on natural language processing (IJCNLP). Nagoya, Japan: Asian Federation of Natural Language Proc., 2013. P. 543–551.

9. RoFormer: Enhanced transformer with Rotary position embedding / J. Su, Y. Lu, S. Pan, A. Murthadha, B. Wen, Y. Liu // Neurocomputing. 2024. Vol. 568. P. 1–12. doi: 10.1016/j.neucom.2023.127063.

10. Zehtab-Salmasi A., Feizi-Derakhshi M.-R., Balafar M.-A. FRAKE: Fusional Real-time Automatic Keyword Extraction. 2021. P. 1–12. URL: https://arxiv.org/abs/2104.04830 (data obraschenija: 28.01.2025).

11. Boudin F. PKE: an open source Python-based keyphrase extraction toolkit // Proc. of COLING the 26th Intern. Conf. on computational linguistics: System demonstrations. Osaka, Japan: Association for Computational Linguistics, 2016. P. 69–73.

12. Reimers N., Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation // Proc. of the Conf. on Empirical Methods in Natural Language Proc. (EMNLP 2020). Punta Cana, Dominican Republic: Association for Computational Linguistics, 2020. P. 4512–4525. doi: 10.18653/v1/2020.emnlp-main.365.

13. Learning Rich representation of keyphrases from text / M. Kulkarni, D. Mahata, R. Arora, R. Bhowmik // Findings of the Association for Computational Linguistics: NAACL. Seattle, United States: Association for Computational Linguistics, 2022. P. 891–906. doi: 10.18653/v1/2022.findings-naacl.67.

14. Sorochina M. V., Korytov P. V., Kholod I. I. Application of neural network methods for keyword extraction for compiling student's resume based on work programs // Proc. XXVI Intern. Conf. on Soft Computing and Measurements (SCM–2023). Saint-Petersburg, Russia: IEEE, 2023. P. 186–189. doi: 10.1109/SCM58628.2023.10159061.

15. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER / S. Tedeschi, V. Maiorca, N. Campolungo, F. Cecconi, R. Navigli // Findings of the Association for Computational Linguistics: EMNLP 2021. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021. P. 2521–2533.

16. Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // Proc. of the Intern. Conf. on Learning Representations (ICLR 2013). Scottsdale, USA: Association for Computational Linguistics, 2013. P. 1–12. doi: 10.48550/arXiv.1301.3781.

17. NLP Evaluation in trouble: On the need to measure LLM data contamination for each benchmark / O. Sainz, J. Campos, I. García-Ferrero, J. Etxaniz, O. Lacalle, E. Agirre // Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore: Association for Computational Linguistics, 2023. P. 10776–10787. doi: 10.18653/v1/2023.findings-emnlp.722.

## Information about the author

**Pavel V. Korytov** – postgraduate student of the Department of Information Systems, Assistant Professor, Department of Software Engineering and Computer Applications, Lead Developer, Department of Digital Services Development, Saint Petersburg Electrotechnical University.
E-mail: thexcloud@gmail.com
https://orcid.org/0000-0001-5534-5389

Статья поступила в редакцию 06.03.2025; принята к публикации после рецензирования 24.05.2025; опубликована онлайн 29.09.2025.

Submitted 06.03.2025; accepted 24.05.2025; published online 29.09.2025.