УДК 621.396.06 Научная статья

https://doi.org/10.32603/2071-8985-2025-18-9-68-78

Извлечение ключевых слов из текстов в условиях отсутствия аннотированных данных с использованием обратной связи

П. В. Корытов✉, И. И. Холод

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

✉ thexcloud@gmail.com

Аннотация. Рассматривается задача извлечения ключевых слов из неструктурированных текстовых документов при следующих условиях: отсутствие аннотированных данных в начале работы (условия «холодного старта»), возможность улучшения результата с использованием обратной связи, необходимость сопоставления ключевым словам их канонической формы. Приведены формальная постановка задачи, анализ и сравнение существующих методов (классических методов, методов на основе BERT, открытых LLM). Для решения задачи предложен комбинированный метод, сначала использующий необучаемый метод извлечения, а после накопления обратной связи – обучаемый метод постобработки ключевых слов. В качестве необучаемого предлагается использовать классический метод (SingleRank, на Inspec $F_1 = 0.26$); в качестве обучаемого – нейросеть на основе BERT+CRF. Рассмотрены различные стратегии дообучения BERT для постобработки ключевых слов: обработка ключевых слов по одному (отрицательный результат), всех ключевых слов в одной строке ($F_1 = 0.34$), предложений с ключевыми словами по одному ($F_1 = 0.42$), всех предложений с ключевыми словами ($F_1 = 0.50$). Также выполнена оценка метода на собственном русскоязычном бенчмарке (аннотации дисциплин); последний вариант дообучения BERT при добавлении в обучающий набор аугментированных данных показывает $F_1 = 0.33$, что сравнимо с LLM t-pro ($F_1 = 0.33$) при меньших требованиях к VRAM (6 Гбайт против 22.8 Гбайт для LLM). Условие представления ключевых слов в канонической форме выполнено с помощью LLM qwen2.5:3b с $F_1 = 0.68$. Полученные результаты могут быть использованы как самостоятельно для сжатого представления текстовых документов (таких, как рабочие программы дисциплин), так и в качестве входных данных для задач тематического моделирования и сравнительного анализа документов.

Ключевые слова: ключевые слова, холодный старт, BERT, обучение с обратной связью, LLM

Для цитирования: Корытов П. В., Холод И. И. Извлечение ключевых слов из текстов в условиях отсутствия аннотированных данных с использованием обратной связи // Изв. СПбГЭТУ «ЛЭТИ». 2025. Т. 18, № 9. С. 68–78. doi: 10.32603/2071-8985-2025-18-9-68-78.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Финансирование: Работа выполнена при поддержке гранта Российского научного фонда № 25-11-20020 (https://rscf.ru/project/25-11-20020/) и Санкт-Петербургского научного фонда.

Original article

Extracting Keywords from Texts under Conditions of Missing Annotated Data Using Feedback

P. V. Korytov✉, I. I. Kholod

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

✉ thexcloud@gmail.com

Abstract. This work addresses the problem of keyword extraction from unstructured text documents under the following conditions: absence of annotated data at the beginning of work («cold start» conditions), the pos-

sibility of improving results through the use of feedback, and the necessity of mapping keywords to their canonical forms. The paper presents a formal problem statement, analysis and comparison of existing methods (classical methods, BERT-based methods, open LLMs). To solve the problem, a combined method is proposed that first uses a non-trainable extraction method, and after accumulating feedback, uses a trainable keyword post-processing method. The classical method (SingleRank, F_1 = 0.26 on Inspec) is proposed as the non-trainable method; a BERT+CRF-based neural network is used as the trainable method. Various strategies for fine-tuning BERT for keyword post-processing are considered: processing keywords one by one (negative result), all keywords in one line (F_1 = 0.34), sentences with keywords one by one (F_1= 0.42), all sentences with keywords (F_1 = 0.50). The method was also evaluated on a proprietary Russian-language benchmark (course annotations); the last variant of BERT fine-tuning with augmented data added to the training set shows F_1 = 0.33, which is comparable to LLM t-pro (F_1 = 0.33) with lower VRAM requirements (6 GB versus 22.8 GB for LLM). The condition of presenting keywords in canonical form was fulfilled using LLM qwen2.5:3b with F_1 = 0.68. The obtained results can be used independently for the concise representation of text documents (such as course descriptions), as well as input data for topic modelling and comparative document analysis tasks.

Keywords: keywords, cold start, BERT, feedback learning, LLM

For citation: Korytov P. V., Kholod I. I. Extracting Keywords from Texts under Conditions of Missing Annotated Data Using Feedback // LETI Transactions on Electrical Engineering & Computer Science. 2025. Vol. 18, no. 9. P. 68–78. doi: 10.32603/2071-8985-2025-18-9-68-78.

Conflict of interest. The authors declare no conflicts of interest.

Financing: The work was supported by the Russian Science Foundation grant No. 25-11-20020 (https://rscf.ru/project/25-11-20020/) and the St. Petersburg Science Foundation.

Введение. Задача извлечения ключевых слов – традиционная задача обработки текстов на естественном языке (NLP, Natural Language Processing). Неформально ее можно сформулировать как сопоставление строке текста набора более коротких подстрок, по которым можно воспроизвести смысл исходного текста.

Эта задача актуальна в условиях работы с неструктурированными текстовыми документами. Ключевые слова, извлеченные из набора (корпуса) таких документов, можно использовать для:

– тематического моделирования, т. е. выявления тем в корпусе документов;

– анализа трендов использования различных ключевых слов;

– сравнительного анализа документов путем сравнения их ключевых слов и т. п.

В данной работе задача рассматривается в контексте следующих условий:

1. Возможность привлечения начальной экспертизы (создание аннотированных данных) отсутствует.

2. Возможно использование экспертизы в форме обратной связи для улучшения качества работы методов.

3. Извлеченным ключевым словам должна быть сопоставлена их каноническая форма, т. е. форма без морфологических следов контекста (падежей, чисел, ...).

Условие 1 также называется условием «холодного старта», т. е. необходимостью запуска метода на новом корпусе, аннотированные данные для которого отсутствуют. Условие 3 связано со сложной морфологией русского языка.

Примером данной задачи, рассмотренной в статье, служит анализ документов, описывающих дисциплины (рабочие программы, РП). В уже введенных РП ключевые слова отсутствуют, и можно предположить, что авторам будет легче скорректировать автоматически выделенные ключевые слова, чем создавать их «с нуля».

Также для развертывания решения установлено ограничение в 16 Гбайт RAM (*Random Access Memory*) и 8 Гбайт VRAM (*Vîdeo Random Access Memory*), поскольку дообучение решения в п. 2 должно происходить в развернутом виде. Аналогичное ограничение установлено также для обучения.

Постановка задачи. Более формально, задачу извлечения ключевых слов можно представить следующим образом:

$$\text{keywords}: X \to \{k_x \in 2^K \Big| \, |k_x| \le \varepsilon_k,$$
$$\forall w \in k_x : w \in x, \text{length}(w) \le \varepsilon_w\},$$

где X – множество всех документов (текстовых строк); $K \subset X$ – множество всех ключевых слов (ключевые слова – тоже текстовые строки); $k_x \in 2^K$ – множество ключевых слов, извлечен-

ных из документа x (2^K – множество всех подмножеств K); w – одно ключевое слово, которое обязательно служит подстрокой документа x, не длиннее, чем ε_w; ε_k – максимальное количество ключевых слов, которые можно извлечь с помощью keywords.

Иными словами, keywords извлекает из документа x максимум ε_k подстрок максимальной длиной ε_w.

Неформально, качественно извлеченные ключевые слова k_x – такие, которые максимально полно описывают смысл x. Формализация такой оценки представляет понятную сложность; как правило, методы извлечения ключевых слов оцениваются F_1-мерой [1] посредством сравнения с эталоном:

$$f_1 : (x, k_x, k_{xT}) \mapsto [0, 1],$$

где x – документ; k_x – извлеченные ключевые слова; k_{xT} – эталонные ключевые слова; f_1 – функция вычисления F_1-меры, возвращающая число от 0 до 1.

В определении F_1-меры основная сложность состоит в определении множества результатов и обстоятельств их истинности. В данной статье использованы два варианта меры:

• $f_{1\text{-strict}}$ – результатами служат ключевые слова; истинно-положительный результат – полное совпадение извлеченного ключевого слова с эталонным;

• $f_{1\text{-lenient}}$ – результатами являются токены ключевых слов; истинно-положительный результат – совпадение токенов.

Следовательно, «жесткая» оценка $f_{1\text{-strict}}$ требует полного совпадения ключевых слов, «мягкая» $f_{1\text{-lenient}}$ – частичного. Данный способ выбран исходя из разрабатываемого метода – большая разница между «жесткой» и «мягкой» оценкой показывает, что метод извлекает ключевые слова, близкие к нужным, но немного отличающиеся от них. Теоретически, оценку такого метода можно улучшить постобработкой.

Запись $f_{1\text{-strict}}@\varepsilon_k$, $f_{1\text{-lenient}}@\varepsilon_k$ означает, что количество извлекаемых ключевых слов ограничено сверху ε_k.

Бенчмарком для оценки извлечения ключевых слов называется множество пар (x, k_{xT}), т. е. множество документов и эталонных ключевых слов. Набор для обучения метода извлечения ключевых слов определяется идентично. Сводка по используемым в данной статье бенчмаркам приведена в табл. 1.

Inspec – один из существующих бенчмарков для извлечения ключевых слов [2]; остальные бенчмарки составлены с помощью LLM (*Large Language Model*) Claude 3.5 Haiku из разных разделов РП. Содержания РП представляют собой более тезисный текст, аннотации – более естественный. Последний бенчмарк составлен с помощью аугментации текстов LLM при сохранении упоминания ключевых слов, чтобы имитировать изменение текстов со временем.

Таким образом, с учетом условий, поставленных во введении, можно определить требования к целевому методу. Пусть $E = \{x_{iT}, k_{xiT}\}_{i=1}^{n}$ – обратная связь, т. е. набор текстовых документов x_{iT} и сопоставленных им эталонных ключевых слов k_{xiT}.

В таком случае необходимо разработать следующие функции:

• $\text{keywords}_\varnothing : x \mapsto x_k$ – версия метода, работающая в условиях холодного старта (необученная, условие 1);

• $\text{train} : E \mapsto \text{keywords}_E$ – функция обучения метода $\text{keywords}_\varnothing$ на обратной связи E (условие *2*).

При этом ставится условие, что при введении разделения E на E_{train} и E_{test} таким образом, что $E = E_{\text{train}} \cup E_{\text{test}}$, $E_{\text{train}} \cap E_{\text{test}} = \varnothing$, верно следующее:

$$\sum_{j=1}^{|E_{\text{test}}|} f_{1\text{-strict}}\left[x_j, \text{keywords}_\varnothing(x_j), x_{jT}\right] <$$
$$< \sum_{j=1}^{|E_{\text{test}}|} f_{1\text{-strict}}\left[x_j, \text{keywords}_{E_{\text{train}}}(x_j), x_{jT}\right],$$

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 (содержания)	Русский	1852	3138.8	6.2
РП (аннотации)	Русский	1660	2007.7	4.8
РП (аугментированные аннотации)	Русский	500	2019.4	4.8

т. е. среднее качество извлечения ключевых слов должно повышаться при использовании обратной связи. В случае использования одного набора данных E может быть поделен в пропорции 80/20; также E_{train}, может быть получен на предыдущей версии корпуса.

Кроме того, в составе keywords должна существовать функция $\text{canonize}: w \mapsto w_c$, отображающая ключевые слова в каноническую форму (условие *3*).

Обзор существующих методов. Существующие методы извлечения ключевых слов, как и область NLP в целом, можно условно разбить на три группы – классические методы, методы на основе BERT (*Bidirectional Encoder Representations from Transformers*) и на основе больших языковых моделей (LLM).

Классические методы. Классические методы извлекают ключевые слова с помощью эвристик. Например, метод YAKE (*Yet Another Keyword Extraction*) [3] использует положение слова в предложении, положение предложения в тексте, регистр букв и т. п.

Также к этой группе относятся графовые алгоритмы, в основном основанные на алгоритме PageRank (https://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/pagerank.pdf): TextRank, SingleRank, TopicRank, PositionRank; все методы основаны на построении графа из слов тем или иным способом и последующем ранжировании вершин [4]. Метод FRAKE (*Fusional Real-time Automatic Keyword Extraction*, https://arxiv.org/abs/2104.04830), тоже относящийся к этой группе, комбинирует статистический и графовый подходы.

Преимущество этой группы методов заключается в более низких системных требованиях (относительно остальных); основной недостаток – более низкое качество извлечения ключевых слов (см. результаты сравнения далее).

Также к их особенностям относится отсутствие обучаемости методов – с одной стороны, для запуска данных методов не требуются аннотированные данные; с другой стороны, качество работы методов нельзя повысить дообучением.

Методы на основе BERT. В отличие от классических методов, для методов на основе BERT появляется возможность обучения.

Метод KeyBERT – необучаемый, основанный на полном переборе всех n-грамм текста; для каждой n-граммы считается эмбеддинг; ключевыми словами становятся n-граммы с минимальным косинусным расстоянием до эмбеддинга исходного текста [5]. По сути, этот метод есть реализация классического подхода с использованием BERT.

Более распространенный подход – вместо описанной инженерии признаков дообучать непосредственно нейросеть для классификации токенов из текста. Представитель этого подхода – модель KBIR-Inspec, представляющая собой модель KBIR (*Keyphrase Boundary Infilling with Replacement*) [6] со слоем классификации сверху, обученную на упомянутом бенчмарке Inspec. Аналогичный подход можно реализовать и на русскоязычных версиях BERT [7].

В данном случае ситуация с обучаемостью обратна классическим методам – качество работы методов можно повысить дообучением, но для первого запуска нужны аннотированные данные.

Теоретически, необходимость в аннотированных данных можно обойти, используя предобученные модели, как предлагается в [8] – использовать модель KBIR-Inspec и WikiNEuRal, обернутые в переводчик (так как модели англоязычные) и механизм устранения дубликатов. Однако в этом случае необходимо, чтобы предобученная модель обучалась на корпусе и ключевых словах, достаточно похожих на целевые.

Кроме того, в отличие от классических методов, для использования BERT необходимо наличие GPU (*graphics processing unit*), в противном случае работа метода будет занимать много времени (минуты).

Методы на основе LLM. Третья категория методов – использование больших языковых моделей (LLM) и промпт-инжиниринга.

В этом случае основной параметр – количество весов модели:

1. Сравнительно небольшие модели (7 байт, 13 байт).

Например, модели серии llama (https://arxiv.org/abs/2407.21783).

2. Большие модели, которые еще можно запустить на потребительских ресурсах (<= 32 байт с квантованием Q4; например, на неспециализированных для AI видеокартах NVIDIA RTX 4090).

Например, qwen2.5:32b (https://arxiv.org/abs/2407.10671) или ее русскоязычный fine-tune t-pro (https://habr.com/ru/companies/tbank/articles/865582/).

3. Большие модели, предоставляемые как сервис (> 32 байт).

Например, Claude 3.5, DeepSeek R1. Не рассматриваются здесь из-за цены использования и/или необходимости специального оборудования для локального доступа.

Для решения прикладных задач, модели группы 3 можно использовать «как есть», модели группы 1, как правило, нужно дообучать. Если не использовать дообучение, консистентность вывода можно повысить, используя multi-shot-промпты [9].

В любом случае, LLM – самые требовательные по ресурсам по сравнению с другими категориями методов. Например, количество весов KBIR-Inspec – 0.35 байт.

Также нужно отметить, что LLM применимы для отображения слов в каноническую форму, либо указав соответствующую инструкцию в исходном промпте, либо отдельным шагом с обработкой ключевых слов через специальный промпт.

Оценка качества методов. Основные результаты сравнения описанных методов на приведенных бенчмарках приведены в табл. 2.

Оценка метода [8] (ансамбль KBIR-Inspec и WiKiNEuRal) на Inspec невозможна из-за несоответствия языков; оценка t-pro на Inspec бессмысленна из-за проблемы контаминации [10].

Как видно из табл. 2, при оценке по $f_{1\text{-strict}}@10$ методы на основе BERT (кроме KeyBERT) работают лучше классических; локальные LLM работают лучше BERT только с размера 32 байт.

Особенности категорий описанных методов в проекции сформулированных условий (раздел «*Постановка задачи*») и потребление ресурсов в вышеописанном эксперименте приведены в табл. 3.

Как видно из табл. 3, LLM удовлетворяют условиям задачи (1–3), однако не удовлетворяют ограничению по использованию VRAM (16 Гбайт). Таким образом, для решения задачи в заданных условиях необходима комбинация рассмотренных методов.

Разработка комбинированного метода. *Структура метода.* Исходя из постановки задачи и обзора предметной области, предлагается комбинированный метод извлечения ключевых слов, представленный на следующем листинге:

Input: X – набор текстовых документов

/* $x \in X$ – документ; m – обучаемая модель; keywords_extract – извлечение ключевых слов из одного документа */

Function keywords_extract(x, m):

Табл. 2. Основные результаты сравнения существующих методов
Tab. 2. Key results of the existing methods comparison

Метод	Качество работы метода (F_1-мера)					
	$f_{1\text{-lenient}}@10$ Inspec	$f_{1\text{-strict}}@10$ Inspec	$f_{1\text{-lenient}}@10$ Содержания РП	$f_{1\text{-strict}}@10$ Содержания РП	$f_{1\text{-lenient}}@10$ Аннотации РП	$f_{1\text{-strict}}@1010$ Аннотации РП
Теоретический максимум	0.85933	0.82699	1	1	1	1
YAKE	0.45441	0.12489	0.24256	0.06376	0.38064	0.14874
PositionRank	0.52098	0.27335	0.17840	0.07837	0.31943	0.16890
SingleRank	**0.54288**	0.26812	0.17300	0.04982	0.23535	0.0416
TextRank	0.53181	0.29583	0.15512	0.03444	0.22128	0.02596
TopicRank	0.48776	0.29877	0.24653	0.12588	0.26615	0.08271
FRAKE	0.42287	0.07329	0.18830	0.03251	0.26634	0.01472
KeyBERT	0.47131	0.02865	0.17591	0.01806	0.29425	0.04444
KBIR-Inspec (0.3 b)	0.53453	**0.36833**	–	–	–	–
KBIR-Inspec + WikiNEuRaL	–	–	0.31720	0.24024	0.40281	0.26771
llama3.1:8b	–	–	0.33608	0.23359	0.31839	0.16230
qwen2.5:32b, en	–	–	0.45991	0.39190	0.44340	0.34981
qwen2.5:32b, ru	–	–	0.45425	0.39956	**0.46657**	0.33485
t-pro	–	–	**0.46203**	**0.40416**	0.41506	**0.35211**

Табл. 3. Особенности рассмотренных методов
Tab. 3. Features of the considered methods

Требования	Классические методы	BERT	LLM, 32 байт
Холодный старт (условие 1)	+	–	+
Обучение (условие 2)	–	+	+
Канонизация (условие 3)	–	+	+
Использование Гбайт RAM/VRAM	0.4/0	1.7/2.6	0.4/22.8
Обучение Гбайт RAM/VRAM	–	2.5/6.0	?/16.0 (QLoRA)

$k_x \leftarrow$ unsupervised(x);
if $m \neq \varnothing$ **then** k $\leftarrow$ supervised(x, k_x, m);
$k_{xc} \leftarrow$ canonize(k_x);
return $\{(w_j, f_j) \mid w_j \in k_x, f_j \in k_{xc}\}_{j=1}^{|k_x|}$
// E – обратная связь
Function keywords$_E$(x):
$m \leftarrow \varnothing$;
if $E \neq \varnothing$ **then** $m \leftarrow$ train(E);
return keywords_extract(x, m).

Функция keywords_extract – это один проход метода; функция keywords оборачивает keywords_extract в сигнатуру, обозначенную в п. «*Постановка задачи*».

Метод использует следующие функции:

– $\text{unsupervised} : x \mapsto k_x$ – необучаемый метод для извлечения ключевых слов k_x из документа x (условие 1);

– $\text{supervised} : (x, k_x, m) \mapsto k_x$ – обучаемый метод для постобработки ключевых слов с помощью модели m (условие 2);

– $\text{canonize} : k_x \mapsto k_x$ – метод для приведения ключевых слов в каноническую форму (условие 3);

– $\text{train} : E \mapsto m$ обучает модель m по обратной связи E.

При этом нет необходимости каждый раз вызывать train в фактической реализации метода; достаточно сначала использовать необучаемый метод, а при накоплении достаточного количества обратной связи E обучать модель m и переходить к обучаемому методу (символ «//» следует понимать как комментарий):

$m \leftarrow \varnothing$; $E \leftarrow \varnothing$; // E – обратная связь.
for $x \in X$ **do**
$k_x \leftarrow$ keywords_extract(x, m); // извлечение ключевых слов из x.
$k_{xT} \leftarrow$ feedback(x, k_x); // опциональная обратная связь по документу.
// Обратная связь (эталонные ключевые слова k_{xT}) сохраняется.
if $k_{xT} \neq \varnothing$ **then** E $\leftarrow$ E $\cup$ $\{x, k_{xT}\}$;
if need_train(m, E) **then** $m \leftarrow$ train(E);
end

Данный порядок также представлен на рисунке в виде схемы.

Таким образом, основную сложность реализации данного метода представляет выбор конкретных реализаций unsupervised, supervised, train и canonize. В данном исследовании:

– в качестве реализации unsupervised использован один из классических методов, рассмотренных выше, по причине низких требований к ресурсам;

– в качестве supervised и train использованы различные стратегии дообучения BERT таким образом, чтобы постобрабатывать вывод unsupervised, по причине лучших результатов и возможности работы в условиях до 8 Гбайт VRAM;

– для реализации canonize использованы LLM, поскольку это единственный из рассмотренных методов, способный решить данную задачу.

Постобработка ключевых слов. Идея постобработки ключевых слов заключается в наблюдаемой разнице между $f_{1\text{-lenient}}$ и $f_{1\text{-strict}}$ в проведен-

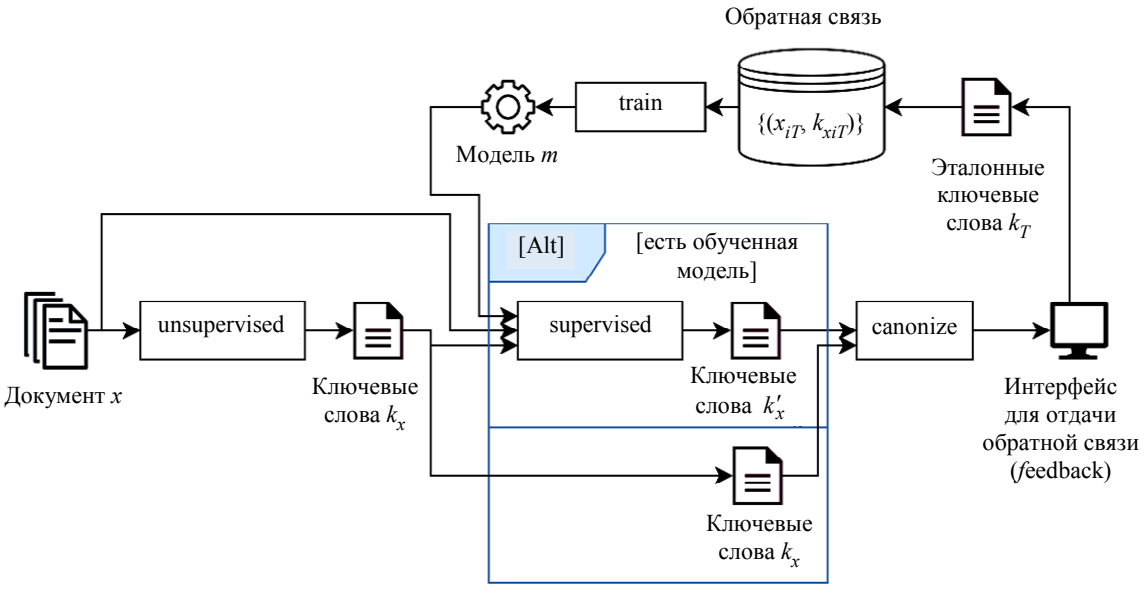

Схема работы метода извлечения ключевых слов
The diagram of keyword extraction method workflow

ном сравнении. Это означает, что качество извлечения ключевых слов можно повысить, исключив из них лишние токены или добавив новые.

Например, для одного из документов набора данных с аннотациями РП метод YAKE извлек следующие ключевые слова (табл. 4, ст. 1).

Табл. 4. Пример обработки ключевых слов
Tab. 4. An example of keywords processing

Исходное ключевое слово	Целевое ключевое слово
Молекулярная физика	Молекулярная физика
Атомная физика	Атомная физика
Физика	Физика
Базовая квантовая физика	Квантовая физика
Дисциплина	
Механические колебания	Механические колебания
Основы квантовой	
Механика	Механика
Динамика	Динамика
2020	

Если заменить их на ключевые слова в столбце 2, удалить дубликаты и исключить ненужные ключевые слова, качество извлечения ключевых слов будет улучшено.

Тогда основным вопросом становится формат представления данных для BERT в методах train supervised, который будет проводить постобработку. Для этого рассмотрены следующие подходы на бенчмарке Inspec:

1. На вход BERT поступают ключевые слова, по одному за раз (отрицательный результат).

2. На вход поступает строка из ключевых слов, соединенных «|» (лучший результат $f_{1\text{-strict}}@10 = 0.34$).

3. На вход поступают предложения, содержание ключевые слова, по одному за раз (лучший результат $f_{1\text{-strict}}@10 = 0.43$).

4. На вход поступает строка из всех предложений, содержащих ключевых слова. Если строка не помещается в контекст, она разделяется по границе предложений (лучший результат $f_{1\text{-strict}}@10 = 0.50$).

Во всех случаях к BERT добавляется дополнительный слой, классифицирующий токены строк. Токены, классифицированные положительно, становятся новыми ключевыми словами.

Из вышесказанного следует, что результаты тем лучше, чем больше контекста передается в нейросеть. Исходя из этого, в качестве основной стратегии выбрана 4-я (supervised_4) с двумя вариациями – выбор предложений с извлеченными ключевыми словами (supervised_{4a}) или с предложениями, которые надо извлечь, исходя из обучающего набора (supervised_{4b}).

Архитектура самой результативной нейросети имеет вид BERT –> Dropout(p=0.1) –> CRF (*Conditional Random Field*), где в качестве BERT ис-

Табл. 5. Оценка предлагаемого метода
Tab. 5. Evaluation of the proposed method

Бенчмарк	Реализация supervised	Реализация unsupervised	Метрики качества	
			$f_{1\text{-lenient}}@10$	$f_{1\text{-strict}}@10$
Inspec	BiLSTM-CRF	∅	**0.69828**	0.49317
Inspec	SingleRank	supervised_{4a}	0.69740	0.49824
Inspec	SingleRank	supervised_{4b}	0.69012	**0.50317**
РП (содержания)	YAKE	supervised_{4a}	0.35880	0.24388
РП (содержания)	YAKE	supervised_{4b}	0.34767	0.24718
РП (аннотации)	PositionRank	supervised_{4a}	0.34293	0.21779
РП (аннотации)	PositionRank	supervised_{4b}	0.47622	0.30822
РП (аннотации)	PositionRank	supervised_{4b} + содержания	0.43138	0.29913
РП (аннотации)	YAKE	supervised_{4b}	0.47401	0.32580
РП (аннотации)	PositionRank	supervised_{4b} + аугментированные аннотации	0.48113	0.32635
РП (аннотации)	YAKE	supervised_{4b} + аугментированные аннотации	**0.51557**	0.34922
РП (аугментированные аннотации)	PositionRank	supervised_{4b}	0.47846	0.32746
РП (аугментированные аннотации)	PositionRank	supervised_{4b} + содержания	0.47846	0.32746
РП (аугментированные аннотации)	YAKE	supervised_{4b}	**0.57275**	**0.40250**

пользуется SpanBERT [11] для английского языка и DeepPavlov/rubert-base-cased [12] – для русского. Результаты оценки предлагаемого метода на всех бенчмарках приведены в табл. 5.

Из этой таблицы следует, что дообучение BERT + CRF – качественный способ для реализации обратной связи постобработкой результатов более простого метода. При этом качество работы тем лучше, чем больше контекста передается на вход нейросети. Следовательно, если тексты достаточно коротки, чтобы полностью влезть в контекст (как в бенчмарке Inspec), рациональнее при появлении достаточного количества обратной связи сразу же использовать нейросеть BERT-BiLSTM-CRF.

На бенчмарке из содержаний РП лучший метод из доступных – LLM. Это может объясняться неоптимальностью выбранного алгоритма, в котором на вход нейросети подаются все предложения, содержащие ключевые слова, – из-за тезисного формата бенчмарка предложения оказываются короткими и контекста оказывается недостаточно.

На бенчмарке из аннотаций РП, состоящем из более «естественного» текста, метод показал результаты, сравнимые с t-pro; на бенчмарке из аугментированных аннотаций (при обучении на обычном бенчмарке) превзошел t-pro.

Таким образом, предложенный метод рационально использовать, когда тексты имеют нетезисный характер и не влезают в контекст BERT, а также когда тексты меняются со временем, сохраняя при этом ключевые слова.

Канонизация ключевых слов. Последний шаг реализации метода – выбор функции приведения ключевых слов в каноническую форму (canonize), т. е. именительный падеж, единственное число и т. п.

Для оценки этой возможности создан бенчмарк на основе бенчмарка с аннотациями РП. Бенчмарк содержит 1660 документов и 5589 уникальных ключевых слов; с помощью LLM Claude 3.5 Sonnet они приведены в каноническую форму, 3917 слов при этом были изменены.

В качестве основы реализации canonize выбрана LLM, удовлетворяющая ограничению на VRAM. LLM преобразовывает слова в каноническую форму путем multi-shot инструкции; преобразованные слова проверяются на корректность с помощью расстояния Джаро–Винклера. Для оценки качества метода составлен бенчмарк с помощью LLM Claude 3.5 Sonnet. Результаты оценки некоторых LLM приведены в табл. 6, где значения с символом «′» определены без этапа проверки на корректность.

Как видно из табл. 6, введение 2-го шага увеличило долю ложноотрицательных результатов, т. е. количество случаев, когда преобразование выполнено не было; но также это привело к увеличению точности (в смысле отношения количества истинно-положительных и истинно-отрицательных случаев к общему количеству случаев).

Среди рассмотренных моделей разумным компромиссом между размером (количеством весов) и качеством работы представляется qwen2.5:3b.

Выводы. В данной статье рассмотрена задача извлечения ключевых слов с тремя условиями: отсутствие данных для обучения в начале работы («холодный старт»), способность к дообучению (обратная связь) и представление ключевых слов в канонической форме.

Для решения был предложен комбинированный метод извлечения ключевых слов, состоящий из трех шагов: *1* – начальное извлечение, *2* – улучшение качества с помощью обратной связи, *3* – приведение слов в каноническую форму.

Табл. 6. Результаты оценки LLM для канонизации ключевых слов
Tab. 6. Results of the evaluation of LLMs for keyword canonization

Модель	Весы	Квант	Метрики качества канонизации					
			f_1'	f_1	Точность' [1]	Точность	Доля ложно-отрицательных результатов' [1]	Доля ложно-отрицательных результатов
llama3.2	1b	Q8_0	0.433	0.476	0.344	0.459	0.042	0.533
llama3.2	3b	Q4_K_M	0.526	0.567	0.416	0.514	0.007	0.389
llama3.1	8b	Q4_0	0.625	0.675	0.5	0.607	0.005	0.22
qwen2.5	0.5b	Q4_K_M	0.426	0.456	0.364	0.447	0.11	0.511
qwen2.5	1.5b	Q4_K_M	0.532	0.564	0.424	0.507	0.017	0.365
qwen2.5	3b	Q4_K_M	0.608	0.628	0.51	0.559	0.014	0.247
qwen2.5	7b	Q4_K_M	0.734	0.741	0.652	0.671	0.015	0.128
qwen3	1.7b	Q4_K_M	0.557	0.564	0.493	0.516	0.066	0.263
qwen3	4b	Q4_K_M	0.61	0.621	0.511	0.545	0.01	0.205

На шаге *1* используется необучаемый классический алгоритм. На наборе данных из аннотаций РП метод YAKE показал результат $f_1 = 0.14$, что является базовым уровнем перед дообучением.

На шаге *2* использована нейросеть BERT+CRF для постобработки результатов шага *1*. Разработанный метод показал разное качество в зависимости от типа входных данных:

• На текстах со связными предложениями (бенчмарк аннотаций РП) метод, дообученный на аугментированных данных, достиг $f_1 = 0.35$. Этот результат сопоставим с качеством LLM t-pro ($f_1 = 0.35$), но требует 6 Гбайт VRAM против 22.8 Гбайт для LLM.

• На аугментированном бенчмарке аннотаций, имитирующем появление новых документов, метод показал $f_1 = 0.40$, что подтверждает способность метода к обобщению.

• На текстах с тезисной структурой (бенчмарк содержаний РП) метод показал более низкий результат $f_1 = 0.25$, уступив LLM t-pro ($f_1 = 0.40$). Это указывает на то, что для эффективной работы метода необходим достаточный контекст в виде полных предложений.

• На англоязычном бенчмарке Inspec предложенный подход достиг $f_1 = 0.50$, что сравнимо с прямым обучением (без предобработки) нейросети с BiLSTM-CRF на этих же данных ($f_1 = 0.49$).

• На шаге *3* задача представления ключевых слов в канонической форме решена отдельно с помощью сравнительно небольшой LLM qwen2.5:3b, справившуюся с задачей $f_1 = 0.68$.

Таким образом, разработанный метод позволяет решать задачу извлечения ключевых слов в заданных ограничениях (холодный старт, обратная связь, требования к VRAM). Он наиболее эффективен для обработки связных текстов, где его качество сопоставимо с рассмотренными LLM.

Дальнейшая работа может включать в себя:

– проверка эффективности способа предобработки, в котором на одно ключевое слово передавалось бы N окружающих предложений, с учетом пересечений;

– сравнение описанных подходов с распознаванием именованных сущностей (NER);

– исследование возможности дообучения LLM для решения задачи извлечения ключевых слов и приведения в каноническую форму, в том числе с помощью QLoRA [13].

Список литературы

1. Manning C. D., Raghavan P., Schütze H. Introduction to information retrieval. Cambridge: Cambridge University Press, 2008. 581 p.

2. 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. doi: 10.3115/1119355.1119383.

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

4. Nomoto T. Keyword extraction: A modern perspective // SN Comp. Sci. 2022. Vol. 4, № 1. P. 92. doi: 10.1007/s42979-022-01481-7.

5. A Comparative study on embedding models for keyword extraction using KeyBERT method / B. Issa, M. B. Jasser, H. N. Chua, M. Hamzah // 2023 IEEE 13th Intern. Conf. on System Engin. and Technol. (ICSET). Shah Alam, Malaysia: Association for Computational Linguistics, 2023. P. 40–45. doi: 10.1109/ICSET59111.2023.10295108.

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

7. Большакова Е. И., Семак В. В. Методы и средства извлечения терминов из текстов для терминологических задач // Программные продукты и системы. 2025. Т. 38, № 1. С. 5–16. doi: 10.15827/0236-235X.149.005-016.

8. Kholod I. I., Korytov P. V., Sorochina M. V. Application of neural network keyword extraction methods for student's CV compilation from discipline work programs // 2023 XXVI Intern. Conf. on Soft Comp. and Measurements (SCM). Saint-Petersburg, RF: IEEE, 2023. P. 143–146. doi: 10.1109/SCM58628.2023.10159061.

9. Berryman J., Ziegler A. Prompt engineering for LLMs: the art and science of building large language model-based applications. Sebastopol, CA: O'Reilly Media, Inc, 2024. 280 p.

10. 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.

11. SpanBERT: Improving pre-training by representing and predicting spans / M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy // Transactions of the Association for Computational Linguistics. 2020. Vol. 8. P. 64–77. doi: 10.1162/tacl_a_00300.

12. Kuratov Yu., Arkhipov M. Adaptation of deep bidirectional multilingual transformers for russian language // Computational Linguistics and Intellectual Technol.: Proc. of the Intern. Conf. «Dialogue 2019». M.: MEPhI, 2019. T. 18. P. 333–339.

13. QLoRA: Efficient Finetuning of Quantized LLMs / T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer // Advances in Neural Inform. Proces. Syst. (NeurIPS). 2023. Vol. 36. P. 10088–10115.

Информация об авторах

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

Холод Иван Иванович – д-р техн. наук, доцент кафедры информационных систем СПбГЭТУ «ЛЭТИ».
E-mail: iiholod@etu.ru
https://orcid.org/0000-0002-7255-5035

References

1. Manning C. D., Raghavan P., Schütze H. Introduction to information retrieval. Cambridge: Cambridge University Press, 2008. 581 p.

2. 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. doi: 10.3115/1119355.1119383.

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

4. Nomoto T. Keyword extraction: A modern perspective // SN Comp. Sci. 2022. Vol. 4. № 1. P. 92. doi: 10.1007/s42979-022-01481-7.

5. A comparative study on embedding models for keyword extraction using KeyBERT method / B. Issa, M. B. Jasser, H. N. Chua, M. Hamzah // 2023 IEEE 13th Intern. Conf. on System Engin. and Technol. (ICSET). Shah Alam, Malaysia: Association for Computational Linguistics, 2023. P. 40–45. doi: 10.1109/ICSET59111.2023.10295108.

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

7. Bol'shakova E. I., Semak V. V. Metody i sredstva izvlechenija terminov iz tekstov dlja terminologicheskih zadach // Programmnye produkty i sistemy. 2025. T. 38, № 1. S. 5–16. doi: 10.15827/0236-235X.149.005-016. (In Russ.).

8. Kholod I. I., Korytov P. V., Sorochina M. V. Application of neural network keyword extraction methods for student's CV compilation from discipline work programs // 2023 XXVI Intern. Conf. on Soft Computing and Measurements (SCM). Saint-Petersburg, RF: IEEE, 2023. P. 143–146. doi: 10.1109/SCM58628.2023.10159061.

9. Berryman J., Ziegler A. Prompt engineering for LLMs: the art and science of building large language model-based applications. Sebastopol, CA: O'Reilly Media, Inc, 2024. 280 p.

10. 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.

11. SpanBERT: Improving pre-training by representing and predicting spans / M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy // Transactions of the Association for Computational Linguistics. 2020. Vol. 8. P. 64–77. doi: 10.1162/tacl_a_00300.

12. Kuratov Yu., Arkhipov M. Adaptation of deep bidirectional multilingual transformers for russian language // Computational Linguistics and Intellectual Technol.: Proc. of the Intern. Conf. «Dialogue 2019». M.: MEPhI, 2019. T. 18. P. 333–339.

13. QLoRA: Efficient finetuning of quantized LLMs / T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer // Advances in Neural Inform. Proces. Syst. (NeurIPS). 2023. Vol. 36. P. 10088–10115.

Information about the authors

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

Ivan I. Kholod – Dr. Sci. (Eng.), Associate Professor, Department of Computer Science, Saint Petersburg Electrotechnical University.
E-mail: iiholod@etu.ru
https://orcid.org/0000-0002-7255-5035

Статья поступила в редакцию 30.09.2025; принята к публикации после рецензирования 06.10.2025; опубликована онлайн 28.11.2025.

Submitted 30.09.2025; accepted 06.10.2025; published online 28.11.2025.