DOI: 10.15514/ISPRAS-2025-37(6)-9

Распознавание заголовков таблиц на основе больших языковых моделей

И.И. Охотин, ORCID: 0009-0003-9600-913X <ilia.ohotin@yandex.ru>
Н.О. Дородных, ORCID: 0000-0001-7794-4462 <nikidorny@icc.ru>
Институт динамики систем и теории управления имени В.М. Матросова СО РАН, Россия, 664033, г. Иркутск, ул. Лермонтова, д. 134

Аннотация. Автоматическое распознавание заголовков таблиц остается сложной задачей из-за разнообразия макетов, включая многоуровневые заголовки, объединенные ячейки и нестандартное форматирование. В данной статье впервые предложена методология оценки эффективности больших языковых моделей в решении этой задачи с использованием текстовых подсказок (промпт-инжиниринга). Исследование охватывает восемь различных моделей и шесть стратегий текстовых подсказок, от минималистичных (zero-shot) до сложных с примерами (few-shot), на выборке из 237 таблиц. Результаты демонстрируют, что размер модели критически влияет на точность: крупные модели (405 млрд. параметров) достигают F-меры ≈ 0.80–0.85, тогда как малые (7 млрд. параметров) показывают F1 ≈ 0.06–0.30. Усложнение текстовых подсказок за счет пошаговых инструкций, критериев поиска и примеров улучшает результаты только для крупных моделей, тогда как для малых приводит к деградации из-за перегруженности контекста. Наибольшие ошибки возникают при обработке таблиц с иерархическими заголовками и объединенными ячейками, где даже средние и крупные модели теряют точность распознавания. Практическая значимость работы заключается в выявлении оптимальных конфигураций текстовых подсказок для разных типов моделей. Например, для крупных моделей эффективны краткие инструкции, а для средних – пошаговые инструкции с критериями поиска. Данное исследование открывает новые возможности по созданию универсальных инструментов для автоматического анализа заголовков таблиц.

Ключевые слова: таблица; заголовки таблицы; распознавание структуры таблиц; распознавание заголовков; большая языковая модель; текстовые подсказки.

Для цитирования: Охотин И.И., Дородных Н.О. Распознавание заголовков таблиц на основе больших языковых моделей. Труды ИСП РАН, том 37, вып. 6, часть 1, 2025 г., стр. 149–166. DOI: 10.15514/ISPRAS–2025–37(6)–9.

Благодарности: Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

Using Large Language Models for Table Header Recognition

I.I. Okhotin, ORCID: 0009-0003-9600-913X <ilia.ohotin@yandex.ru>
N.O. Dorodnykh, ORCID: 0000-0001-7794-4462 <nikidorny@icc.ru>

Matrosov Institute for System Dynamics and Control Theory
of the Siberian Branch of Russian Academy of Sciences (ISDCT SB RAS),
134, Lermontov st., Irkutsk, 664033, Russia.

Abstract. Automatic table header recognition remains a challenging task due to the diversity of table layouts, including multi-level headers, merged cells, and non-standard formatting. This paper is the first to propose a methodology to evaluate the performance of large language models on this task using prompt engineering. The study covers eight different models and six prompt strategies with zero-shot and few-shot settings, on a dataset of 237 tables. The results demonstrate that model size critically affects the accuracy: large models (405 billion parameters) achieve F1 ≈ 0.80–0.85, while small ones (7 billion parameters) show F1 ≈ 0.06–0.30. Complicating prompts with step-by-step instructions, search criteria, and examples improves the results only for large models, while for small ones it leads to degradation due to context overload. The largest errors occur when processing tables with hierarchical headers and merged cells, where even large models lose up to accuracy of recognition. The practical significance of this paper lies in identifying optimal configurations of prompts for different types of models. For example, short instructions are effective for large models, and step-by-step instructions with search criteria are effective for medium ones. This study opens up new possibilities for creating universal tools for automatic analysis of table headers.

Keywords: table; table headers; table structure recognition; header recognition; large language model; prompt engineering.

For citation: Okhotin I.I., Dorodnykh N.O. Using large language models for table header recognition. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 6, part 1, 2025, pp. 149-166 (in Russian). DOI: 10.15514/ISPRAS-2025-37(6)-9.

Acknowledgements. This work was supported by the state assignment of Ministry of Science and Higher Education of the Russian Federation (theme No. 1023110300006-9).

1. Введение

Таблицы служат универсальным инструментом для структурированного представления данных в научных публикациях, финансовых отчетах, веб-документах и других областях. Их способность компактно отображать сложные взаимосвязи между элементами делает их незаменимыми для анализа и принятия решений. Однако автоматическая обработка таблиц остается сложной задачей из-за разнообразия их макетов: объединенные ячейки, многоуровневые заголовки, наличие пустых областей, нестандартное форматирование и отсутствие единых правил оформления. Эти особенности затрудняют как физическое распознавание структуры таблиц (например, границ строк и столбцов), так и семантическую интерпретацию заголовков, которые часто определяют контекст данных. При этом автоматическое распознавание заголовков таблиц актуально для таких приложений, как извлечение данных, построение схем баз данных и предварительная обработка информации для машинного обучения.

Современные подходы к распознаванию структуры таблиц, включая методы на основе свёрточных и графовых нейронных сетей, а также предварительно обученные языковые модели, основанные на архитектуре трансформер, демонстрируют прогресс в этой области [1-2], но сталкиваются с ограничениями при работе с иерархическими заголовками и шумными данными. Появление больших языковых моделей открыло новые возможности для обработки табличных данных за счёт своей способности более глубоко анализировать контекст и семантику данных [3]. Однако их эффективность зависит от стратегий взаимодействия с использованием текстовых подсказок (так называемого, промпт-

инжиниринга – *prompt engineering*), который до сих пор недостаточно изучен в контексте распознавания заголовков таблиц.

В данной статье предлагается новая методология автоматического распознавания заголовков таблиц, обладающих разной структурной компоновкой, на основе использования различных больших языковых моделей. В частности, основной вклад данной работы состоит:

- В оценке влияния различных видов текстовых подсказок, размеров больших языковых моделей и структурной сложности таблиц на точность распознавания заголовков. Так в исследовании протестированы 8 моделей (от 7 млрд. до 405 млрд. параметров) с шестью типами текстовых подсказок, включая *zero-shot* и *few-shot* подходы.
- Впервые получены экспериментальные оценки производительности больших языковых моделей в контексте решения задачи распознавания заголовков таблиц. Эксперименты проведены на выборке из 237 таблиц. Результаты показывают, что даже средние и крупные модели требуют адаптивных стратегий формирования текстовых подсказок для обработки нестандартных макетов таблиц, а их эффективность напрямую зависит от минимализма и ясности формулировок текстовых подсказок.

Статья организована следующим образом: Раздел 2 представляет современное состояние исследований в области автоматического распознавания структуры таблиц. В разделе 3 описывается методология исследования, включая формальную постановку задачи, подготовку данных и другие детали. Раздел 4 включает описание экспериментов, их результаты и обсуждение. В заключении делается краткий обзор полученных результатов, а также приводятся планы будущей работы.

2. Современное состояние исследований

Таблицы являются распространённым и достаточно эффективным средством для представления и хранения структурированных данных во многих предметных областях. Однако они в первую очередь предназначены для интерпретации человеком, что делает их автоматическую обработку проблематичной. Область, занимающаяся автоматическим пониманием табличной информации (*table understanding*) [4-5], рассматривает следующие основные проблемы обработки таблиц:

1) Обнаружение (поиск и идентификация) таблиц в исходном информационном источнике, например, электронном документе или изображении (например, скан документа).
2) Распознавание физической структуры таблиц посредствам идентификации строк, столбцов и ячеек (восстановление матричной сетки) в обнаруженной таблице, которая обычно рассматривается как входное изображение.
3) Восстановление логической структуры таблиц с целью идентификации метаданных (заголовков), которые описывают некоторое подмножество значений ячеек данных в таблице.
4) Семантическое аннотирование содержимого таблиц понятиями из внешнего словаря (модели предметной области, онтологии, или графа знаний) для восстановления отсутствующей семантики данных.

Проблемы распознавания физической и логической структуры таблиц (*table structure recognition*) в известной научной литературе обычно рассматриваются вместе. Существует два основных направления к разработке методов, решающих эти проблемы:

1) Методы, основанные на правилах, например, [6-8], опираются на правила анализа и интерпретации таблиц. Такие решения обычно не охватывают все многообразие

макетов таблиц, форматирования и содержимого. Они, как правило, ограничены обычными макетами, атомарными ячейками и плоскими заголовками, игнорируя случаи, когда эти предположения не выполняются, как обсуждалось в [5, 9].

2) Методы, управляемые данными. Такие решения могут использовать как традиционные модели на основе машинного обучения, например, метод опорных векторов (*Support Vector Machine, SVM*) и случайны лес (*Random Forest*) [10], или кластерный анализ [11], так и модели на основе глубокого обучения, такие как [12-25].

Ранние методы распознавания, основанные на правилах и анализе макета с опорой на техники OCR (*Optical Character Recognition*), показывали невысокую степень адаптивности к разнообразию стилей таблиц и шуму распознавания текста. С приходом глубокого обучения появились решения на основе архитектуры свёрточных нейронных сетей (*Convolutional Neural Network, CNN*), например, модель TableNet [12], которая в режиме «*end-to-end*» одновременно детектирует область таблицы и сегментирует строки и столбцы, достигая передовых результатов на наборах данных ICDAR. Однако такие решения остаются чувствительными к объединённым ячейкам и сложным заголовочным структурам. Графовые модели (*Graph Neural Network, GNN*), такие как TGRNet [13] и TSRNet [14], переосмыслили задачу как восстановление графа ячеек, позволяя захватывать пространственные и логические связи между ними, однако они требуют дорогостоящей разметки ячеек для обучения. Решения, основанные на архитектуре трансформер (*Transformer*), в частности, TSRFormer [15], TableFormer [16], VAST [17], TATR [18] и TableVLM [19] интегрируют механизмы внимания для поэтапного декодирования координат ячеек и логической структуры, улучшая точность на сложных макетах таблиц, но по-прежнему испытывают сложности с многоуровневыми заголовками. Гибридные методы, представленные в TaPaS [20] и TaBERT [21], используют предварительное обучение на парах «*текст-таблица*» для семантического понимания содержимого столбцов и заголовков, однако их качество распознавания ухудшается на предметных областях без схожих метаданных. Системы по типу Global Table Extractor (GTE) [22] объединяют детекторы объектов с иерархическими классификаторами ячеек для совместной оптимизации определения и распознавания структуры, демонстрируя рост точности сегментации, но остаются уязвимыми к размытым границам таблиц. При этом каскадная природа этих конвейеров (*pipelines*) приводит к накоплению ошибок – промахи на этапе обнаружения таблиц усугубляются на этапе распознавания структуры. Попытки интеграции визуальных, текстовых и макетных (*layout*) признаков способствуют снижению влияния различных ошибок, но пока не существует единой архитектуры, полноценно объединяющей все модальности. Высокая вычислительная стоимость решений на основе трансформеров ограничивает их масштабирование на большие корпуса разнообразных документов с таблицами, чему противостоят исследования в направлении лёгких и адаптивных моделей для эффективного извлечения структуры таблиц, включая заголовков. Наконец, подходы вроде TSR-DSAW [23] показали, что моделирование пространственных ассоциаций слов между ячейками может улучшить обнаружение заголовков, но сильно зависит от качества идентификации отдельных слов в изображениях таблиц низкого разрешения.

Отдельно стоит упомянуть подходы, основанные на больших языковых моделях (*large language models*), которые все чаще применяются для решения различных табличных задач с использованием контекстного обучения. Представление текстовой подсказки (*prompt*) для таблицы может играть важную роль в способности моделей обрабатывать табличные данные. В частности, в работах [24-25] исследуется возможность больших языковых моделей понимать структуру таблиц, обладающих разной компоновкой, применяя технику формирования текстовых подсказок (промпт-инжиниринга – *prompt engineering*). Кроме того дается оценка производительности по отдельным форматам представления таблиц и влияния шума в данных. Однако такие походы направлены только на такие структурные задачи,

которые относятся к идентификации строк, столбцов и ячеек по заданным индексам, опуская обработку заголовков.

Таким образом, в современном ландшафте исследований по распознаванию структуры таблиц наблюдается переход от классических правил и эвристик к сложным гибридным и нейросетевым решениям, однако ни один из подходов пока не решает проблему извлечения и семантического понимания заголовков во всём их многообразии. Ключевые направления включают методы для учёта логической структуры на основе CNN-моделей, GNN-моделей и моделей на основе архитектуры трансформер, каждый из которых демонстрирует улучшения на стандартных тестовых наборах данных (бенчмарках), но испытывает трудности с обобщением на реальных таблицах со сложными иерархическими заголовками. Поэтому разработка нового методологического и программного обеспечения, решающего указанные проблемы, остаётся актуальной.

3. Методология

3.1 Формальная постановка задачи

Таблицы представляют собой структурированную форму организации данных, где информация располагается в виде сетки из строк и столбцов. Формально, исходную таблицу *T* можно представить в виде:

$$T = \langle C, A, M \rangle, \tag{1}$$

где $C = \{c_{1,1}, c_{1,2}, \dots, c_{i,j}, \dots, c_{n,m}\}$ – матрица ячеек размерности в *n*-строк и *m*-столбцов; $c_{i,j}$ – содержимое ячейки в *i*-строке *j*-столбца; $A = \{a_{i,j}\}, a_{i,j} \in \{header, data, merged\}$ – матрица структурных атрибутов (типов ячейки), где «*header*» – ячейка заголовка, «*data*» – ячейка данных «*merged*» – объединённая ячейка, при этом $rowspan(c_{i,j})$ и $colspan(c_{i,j})$ – параметры объединённых ячеек; *M* – остальные метаданные, кроме заголовков (например, подпись таблицы, текст окружения в документе, сноски). Следует отметить, что в данной работе в качестве значений ячеек рассматриваются только текстовые данные разных форматов, исключая изображения и другие подтаблицы. Метаданные *M* также не рассматриваются.

Заголовки – это специальные ячейки, которые обозначают категории или описания для данных в соответствующих строках или столбцах *T* (например, «*Год*», «*Население (млн.)*», «*Всего*»). Заголовки обычно находятся в верхней части столбцов или в начале строк, но в сложных таблицах могут располагаться в других позициях, включая многоуровневые или иерархические структуры. Формально, заголовки таблицы *H* можно представить в виде:

$$H = \{h_1, h_2, \dots, h_k\}, \tag{2}$$

где $H \subset C$ и каждый $h_l \in H, l = \overline{1,k}$ – соответствует некоторому логическому уровню иерархии (основной заголовок, подзаголовок столбцов/строк), учитывая структурные особенности *T*.

Задача автоматического распознавания заголовков *H* заключается в определении ячеек, выполняющих функцию заголовков, на основе их содержимого, форматирования и расположения в таблице. Формально это можно представить в виде отображения:

$$F(P_i): T \to H, i = \overline{1,6}, \tag{3}$$

где *F* – модель, предсказывающая заголовки (в данном исследовании используются большие языковые модели для решения этой задачи); *P* – текстовая подсказка, которая на входе получает модель (в данном исследовании предлагается шесть видов текстовых подсказок), при этом: $P_i = \{T', I\}$, где T' – текстовое представление таблицы *T*; *I* – инструкции по распознаванию заголовков *H*.

Таким образом, модель должна определить координаты или идентификаторы ячеек, являющихся заголовками, анализируя семантику, структуру и контекст данных. Ожидается, что большие языковые модели смогут обрабатывать таблицы различной сложности, включая стандартные таблицы с заголовками в первой строке, а также таблицы с многоуровневыми заголовками, объединёнными ячейками или нестандартным расположением заголовков. На рис. 1 представлена обобщенная схема дизайна предлагаемого исследования. Далее подробнее рассмотрим его основные аспекты.

Рис. 1. Общая схема дизайна предлагаемого исследования.
Fig. 1. The general design scheme of the proposed study.

3.2 Вопросы исследования

Исследование направлено на изучение факторов, влияющих на производительность больших языковых моделей в задаче распознавания заголовков таблиц. В частности, сформулированы следующие аспекты:

1) *Влияние текстовых подсказок:* Как сложность и детализация текстовой подсказки влияет на способность модели точно определять заголовки? В исследовании используются различные стратегии формирования текстовых подсказок: от простых (*zero-shot*) до сложных текстовых подсказок с пошаговыми инструкциями и примерами (*few-shot*). Улучшают ли более подробные инструкции и примеры производительность моделей?
2) *Влияние размера модели:* Как количество параметров модели влияет на точность распознавания заголовков таблиц? Ожидается ли, что более крупные модели, содержащие сотни миллиардов параметров, будут стабильно превосходить меньшие модели, или существует точка, после которой увеличение размера не приводит к значительному улучшению?
3) *Влияние структуры таблицы:* Как структурная сложность таблиц (например, наличие объединённых ячеек, многоуровневых заголовков или нестандартного расположения заголовков) влияет на точность распознавания? Способны ли модели эффективно обрабатывать сложные таблицы, или их производительность снижается при увеличении сложности структуры?

Данные вопросы охватывают ключевые переменные исследования, такие как: *вид текстовой подсказки*, *размер модели* и *структура таблиц*. Результаты позволят определить оптимальные конфигурации моделей и текстовых подсказок для различных сценариев, а также выявить ограничения больших языковых моделей в обработке сложных табличных данных.

3.3 Используемые модели

Для изучения способности больших языковых моделей распознавать заголовки таблиц выбрано восемь моделей, различающихся по размеру (количеству параметров). Модели разделены на три категории: *малые* (модели > 70 млрд. параметров), *средние* (модели < 70 и > 100 млрд. параметров) и *крупные* (модели < 100 млрд. параметров), что позволяет оценить влияние размера модели на производительность. Все модели являются инструктивно-обученными (*instruction-tuned*), что делает их подходящими для выполнения задач, требующих следования сложным инструкциям. В табл. 1 приведен список моделей с их краткими характеристиками.

Табл. 1. Большие языковые модели, используемые для исследования.
Table 1. Large language models used for research.

Название	Количество параметров	Краткое описание
Малые модели		
Mistral-7B-Instruct-v0.3 [26]	7 млрд.	Модель от Mistral AI, оптимизированная для выполнения инструкций.
Llama-3.1-8B-Instruct [27]	8 млрд.	Модель из семейства Llama 3 от Meta, предназначенная для диалогов и инструкций.
Mistral-Small-24B-Instruct-2501 [28]	24 млрд.	Более крупная модель от Mistral AI, обладающая компактным размером для запуска на одном GPU RTX 4090.
Gemma-2-27B-it [29]	27 млрд.	Модель от Google, разработанная для обработки инструктивных запросов.
Средние модели		
Llama-3.3-70B-Instruct-Turbo [30]	70 млрд.	Более крупная модель Llama 3 от Meta с улучшенными возможностями.
Qwen2-72B-Instruct [31]	72 млрд.	Модель от Qwen, оптимизированная для сложных инструктивных задач.
DeepSeek-R1-Distill-Llama-70B [32]	70 млрд.	Более эффективная и производительная дистиллированная версия Llama от DeepSeek AI.
Крупные модели		
Llama-3.1-405B-Instruct [33]	405 млрд.	Самая крупная модель Llama 3.1 от Meta с высокой производительностью.

3.4 Подготовка набора данных

Для проведения исследования выбран англоязычный набор данных PubTables-1M [34], который содержит около миллиона таблиц, извлечённых из научных статей, доступных в архиве PubMed Open Access. Этот набор данных предоставляет подробные аннотации для задач обнаружения таблиц, распознавания их структуры и функционального анализа, включая информацию о расположении ячеек, их содержимом и ролях (например, заголовки, данные). PubTables-1M был выбран благодаря его разнообразию таблиц, включающих как простые структуры с заголовками в первой строке, так и сложные таблицы с нестандартным расположением заголовков, которые могут быть многоуровневыми и иерархическими, а

также содержать объединённые ячейки. Все это делает данный набор идеальным для тестирования способности больших языковых моделей распознавать заголовки.

Этапы подготовки набора данных:

1) *Выбор таблиц.* Из исходного набора данных PubTables-1M случайным образом отобрано 237 таблиц, чтобы обеспечить репрезентативность тестового набора. Выборка охватывает таблицы различной сложности, включая стандартные таблицы с заголовками в первой строке, таблицы с иерархическими заголовками и таблицы с нестандартным расположением заголовков. Это позволяет оценить производительность моделей в разнообразных сценариях.
2) *Извлечение и очистка таблиц.* Для каждой выбранной таблицы были извлечены данные и аннотации из оригинальных JSON-файлов, предоставленных в PubTables-1M. Полные аннотации PubTables-1M включают множество деталей, таких как координаты слов, визуальные характеристики (например, размер шрифта) и ограничивающие рамки (*bounding boxes*) в формате PDF и изображений. Эти избыточные данные были удалены из выбранных таблиц. Таким образом, итоговые JSON-аннотации содержат информацию только о структуре таблицы, включая координаты ячеек, их содержимое и роли (например, является ли ячейка заголовком).
3) *Преобразование таблиц в формат DataFrame*. Согласно работе [24], большие языковые модели при решении различных структурных табличных задач лучше всего понимают формат DataFrame (JSON-подобный формат, используемый в библиотеке pandas), который показал самую высокую эффективность в экспериментах. Таким образом, принято решение преобразовать отобранные таблицы и JSON-аннотации в формат pandas DataFrame, что также упрощает последующую обработку и передачу данных в модели.
4) *Разметка таблиц.* Для каждой таблицы был сгенерирован соответствующий файл с разметкой заголовков (*ground truth*), взятых из JSON-аннотации PubTables-1M. Полученные файлы с разметкой содержат строку, указывающую на ячейки заголовков в формате: *«(row-1, column-1); (row-2, column-2); ...»*. Координаты представляют собой пары (строка, столбец), что позволяет точно идентифицировать заголовки для оценки производительности моделей.

Таким образом, подготовленный набор данных состоит из 237 таблиц в формате pandas DataFrame с соответствующими аннотациями заголовков. Данный набор предоставляет прочную основу для тестирования и сравнения различных больших языковых моделей в задаче распознавания заголовков таблиц, а также позволяет проводить анализ влияния структурной сложности таблиц на итоговый результат.

3.5 Формирование текстовых подсказок для моделей

Для взаимодействия с большими языковыми моделями разработаны шесть различных стратегий формирования текстовых подсказок, варьирующихся по уровню детализации и количеству предоставляемых примеров. Целью является оценка того, как сложность текстовой подсказки и наличие примеров влияют на способность моделей точно определять заголовки таблиц. Текстовые подсказки созданы с учётом рекомендаций, предложенных в [35], подчеркивающих важность структурированных инструкций, критериев и примеров (*few-shot*) для задач анализа таблиц.

Шесть основных стратегий формирования текстовых подсказок:

1) Простая текстовая подсказка (*zero-shot*) – представляет собой минимальный запрос, проверяющий базовую способность модели распознавать заголовки без дополнительной информации или примеров. Он подходит для оценки начальных

возможностей моделей, но может быть менее эффективен для анализа сложных таблиц. Пример текстовой подсказки:

- Пользовательский запрос: «*Ты эксперт по анализу таблиц. Определи заголовки в таблице:* {*таблица*}»

2) Текстовая подсказка с пошаговой инструкцией (*step-by-step*) – предоставляет структурированный подход, направляя модель к системному анализу таблицы, что может улучшить точность распознавания. Пример текстовой подсказки:
   - Пользовательский запрос: «*Ты эксперт по анализу таблиц. Определи заголовки в таблице:* {*таблица*}»
   - Системные инструкции: «*Подробный алгоритм, включающий следующие шаги:*
     - *Визуальный осмотр таблицы для изучения её структуры.*
     - *Идентификация потенциальных заголовков на основе обобщающего текста и форматирования.*
     - *Анализ позиционирования заголовков (сверху, слева, справа).*
     - *Формирование JSON-ответа с координатами заголовков».*
3) Текстовая подсказка с пошаговой инструкцией и критериями поиска – углубляет инструкции, предоставляя модели чёткие критерии для идентификации заголовков, что особенно полезно для сложных таблиц. Пример текстовой подсказки:
   - Пользовательский запрос: «*Ты специалист по анализу сложных таблиц. Твоя задача — определить все заголовки в представленной таблице:* {*таблица*}»
   - Системные инструкции: «*Критерии поиска заголовков:*
     - *Семантический анализ для поиска ключевых слов, указывающих на категории.*
     - *Позиционный анализ для определения расположения заголовков.*
     - *Анализ структурных паттернов (например, повторяющиеся заголовки).*
     - *Контекстуальный анализ связей между ячейками.*
     - *Анализ метаданных (размер шрифта, выравнивание).*
     - *Иерархический анализ для выделения многоуровневых заголовков».*
4) Текстовая подсказка с пошаговой инструкцией и критериями, включающая один пример (*1-shot*) – данная текстовая подсказка аналогична предыдущей, но включает также один пример таблицы с её правильными аннотациями заголовков. Пример содержит входные данные (таблицу в формате pandas DataFrame) и ожидаемый выход (JSON с координатами и характеристиками заголовков, такими как текст, позиция и уровень иерархии). Пример помогает модели понять ожидаемый формат ответа и тип заголовков, что может улучшить производительность.
5) Текстовая подсказка с пошаговой инструкцией и критериями, включающая два примера (*2-shot*) – включает два примера таблиц с аннотациями заголовков, охватывающих разные структуры (например, заголовки сверху и сбоку). Это увеличивает контекст и помогает модели лучше обобщать информацию.
6) Текстовая подсказка с пошаговой инструкцией и критериями, включающая три примера (*3-shot*) – наиболее подробная текстовая подсказка, содержащая три примера таблиц с аннотациями. Примеры включают таблицы с различными структурами, такими как многоуровневые заголовки и нестандартное расположение, предоставляя модели максимальный контекст для обучения на образцах.

Каждая текстовая подсказка была разработана для оценки влияния уровня детализации и количества примеров на точность распознавания заголовков. Так простая текстовая

подсказка (*zero-shot*) тестирует базовые способности модели, тогда как текстовые подсказки с примерами (*few-shot*) проверяют, насколько модель может использовать дополнительный контекст для улучшения результатов. Это позволяет определить оптимальную стратегию формирования текстовых подсказок для различных моделей и типов таблиц. Однако увеличение сложности текстовой подсказки может также увеличить вычислительные затраты и требования к контекстному окну модели.

Табл. 2 суммирует характеристики текстовых подсказок и их предполагаемую эффективность, основанную на уровне детализации и наличии примеров.

Табл. 2. Основные характеристики стратегий формирования текстовых подсказок. «int» – индикатор пошаговых инструкций; «crit» – индикатор критериев поиска.
Table 2. The main characteristics of prompting strategies: "int" is an indicator of step-by-step instructions; "crit" is an indicator of search criteria.

№	Вид текстовой подсказки	Пошаговый алгоритм	Наличие критериев поиска	Наличие примеров	Ожидаемая эффективность
1	*zero-shot*	–	–	–	*низкая*
2	*zero-shot+int*	+	–	–	*средняя*
3	*zero-shot+int+crit*	+	+	–	*выше среднего*
4	*1-shot+int+crit*	+	+	+	*высокая*
5	*2-shot+int+crit*	+	+	+	*высокая*
6	*3-shot+int+crit*	+	+	+	*очень высокая*

4. Эксперименты

4.1 Экспериментальные настройки

Для автоматизации проведения экспериментов по распознаванию координат ячеек с заголовками в табличных данных была разработана специальная программная среда на языке Python 3.10. При этом использовалась библиотека LangChain [36], в частности, плагин langchain_together, обеспечивающего удобный интерфейс взаимодействия с API платформы Together.ai [37]. Все запросы к языковым моделям производились через данный API с температурой генерации (*temperature*) равной 0.0 для обеспечения стабильности и воспроизводимости результатов (ответов).

Отобранные и подготовленные таблицы для экспериментов хранятся в каталоге «*Tables_for_models*», каждая из которых сопровождалась отдельным файлом с координатами ячеек, размеченных как заголовки в формате *row_index* и *col_index*.

С использованием разработанной среды было протестировано 8 языковых моделей (модели типа LLaMA, Mistral, DeepSeek, Gemma и Qwen) в сочетании с 6 различными стратегиями текстовых подсказок. В итоге общее количество запросов составило: 237 таблиц × 6 текстовых подсказок × 8 моделей = 11 376 запросов.

4.2 Метрики оценки

В качестве основных метрик оценки качества распознавания заголовков использовались стандартные метрики: *точности* (*Precision*), *полноты* (*Recall*) и *F-меры* (*F1*).

$$Precision = \frac{|TP|}{|TP|+|FP|}, \quad Recall = \frac{|TP|}{|TP|+|FN|}, \quad F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4}$$

где *TP (True Positives)* – корректно предсказанные координаты заголовков моделью; *FP (False Positives)* – координаты, предсказанные моделью, но не являющиеся заголовками; *FN (False Negatives)* – координаты истинных заголовков, не найденные моделью.

4.3 Результаты и обсуждение

Последовательная обработка 11 376 запросов заняла ≈ 49 344 секунд (≈ 13,7 часов). Переход на параллельную обработку в пять потоков снизил это время до ≈ 9 869 секунд (≈ 2,7 часов), что ускорило выполнение обработки запросов почти в пять раз.

Результаты экспериментальной оценки представлены на рис. 2-4. Они демонстрируют тепловую карту значений полученных оценок точности, полноты и F-меры для каждой комбинации «*модель* × *текстовая подсказка*», где на оси «*Y*» представлены названия моделей, а оси «*X*» – номера текстовых подсказок (от самой простой «*zero-shot*» 1 до самой сложной «*3-shot+int+crit*» 6). При этом тёмно-синие цвета соответствуют высоким оценкам, например, F1 (≈ 0.80–0.85) и сконцентрированы в строках крупных и средних моделей, таких как Meta-Llama-3.1-405B и Llama-3.3-70B. У моделей Gemma-2-27b-it и DeepSeek-R1-Distill-Llama-70B преобладают средние значения (оттенки синего с примесью зелёного), что указывает на оценки F1 около 0.70–0.80. Более светлые, «приглушённые» участки в нижней части карты – это результаты малых моделей, таких как Mistral-7B и Mistral-Small-24B, где оценки F1 падает до 0.06–0.30 в зависимости от текстовых подсказок.

Первые две подсказки (номера 1 и 2) дают максимально выраженный контраст: крупные и средние модели уверенно работают в любом режиме, тогда как мелкие сильно реагируют на усложнённую текстовую подсказку под номером 1. Столбцы с текстовыми подсказками 3–6 менее вариативны для больших моделей, что свидетельствует об их устойчивости к формулировке задания, в то время как у малых моделей цвета ряда ещё светлее, показывая деградацию качества.

На рис. 5 приведена радиальная диаграмма, которая представляет собой визуализацию значений оценок F-меры для различных моделей в зависимости от текстовых подсказок. Каждая ось диаграммы соответствует конкретной текстовой подсказки (от 1 до 6), а линии или области обозначают модели разного масштаба.

В целом данная диаграмма соотносится с тепловой картой, представленной на рис. 4. Так, на диаграмме крупные и средние модели, такие как Meta-Llama-3.1-405B и Llama-3.3-70B, образуют обширные многоугольники с вершинами, значительно удалёнными от центра. Это указывает на высокие значения оценки F1 (примерно 0.80–0.85) по всем текстовым подсказкам. Их области занимают большую часть диаграммы, подчёркивая превосходство этих моделей в решении задач независимо от их сложности. Модели среднего уровня, такие как Gemma-2-27b-it и DeepSeek-R1-Distill-Llama-70B, формируют многоугольники с умеренным радиусом. Их вершины расположены на среднем расстоянии от центра, что соответствует оценки F1 в диапазоне 0.70–0.80. Это отражает стабильную, но не выдающуюся производительность. Малые модели, такие как Mistral-7B и Mistral-Small-24B, представлены многоугольниками, стянутыми к центру диаграммы. Их вершины находятся близко к началу координат, что свидетельствует о низких значениях F1-меры (0.06–0.30), особенно на некоторых текстовых подсказках.

При рассмотрении отдельных осей видно, что на текстовых подсказках 1 и 2 крупные и средние модели демонстрируют максимальные радиусы, тогда как малые модели едва выходят за пределы центральной области. Это подчёркивает слабую адаптацию малых моделей к усложнённым заданиям, особенно на текстовой подсказке 2. На текстовых подсказках 3–6 крупные и средние модели сохраняют стабильно высокие радиусы, что указывает на их устойчивость к вариациям в формулировке задач. Малые модели, напротив, продолжают показывать низкие значения, их многоугольники остаются сжатыми, что говорит о снижении качества при обработке более сложных текстовых подсказок.

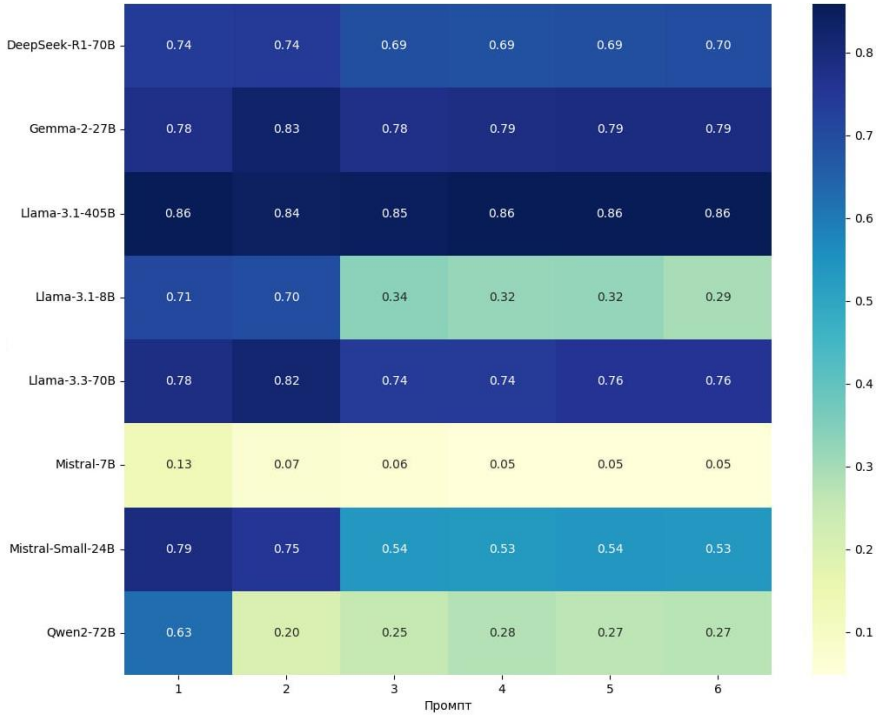

Рис. 2. Тепловая карта полученных значений оценки точности.
Fig. 2. Heat map of the obtained precision values.

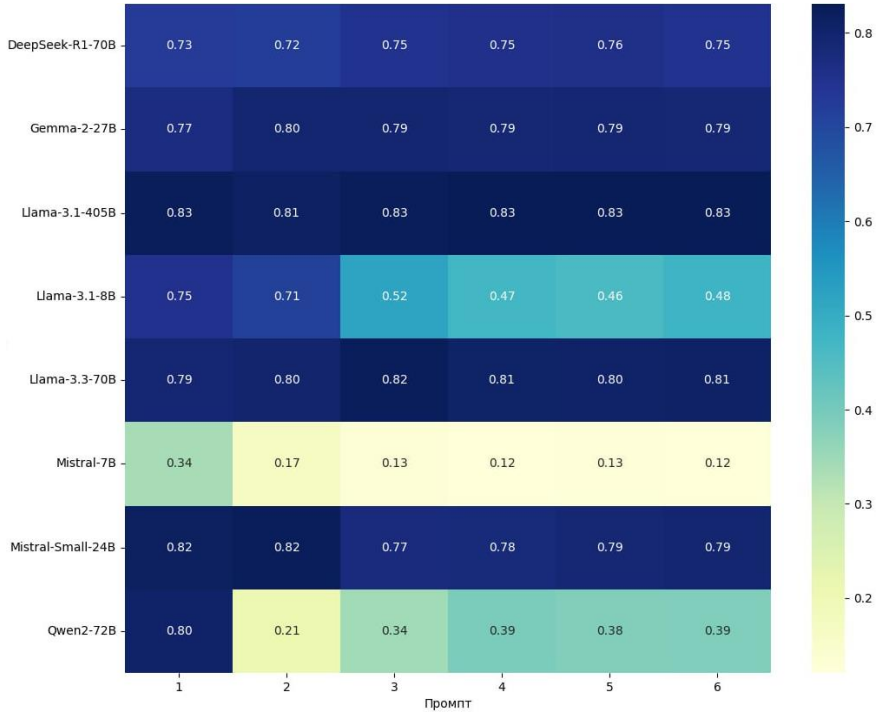

Рис. 3. Тепловая карта полученных значений оценки полноты.
Fig. 3. Heat map of the obtained recall values.

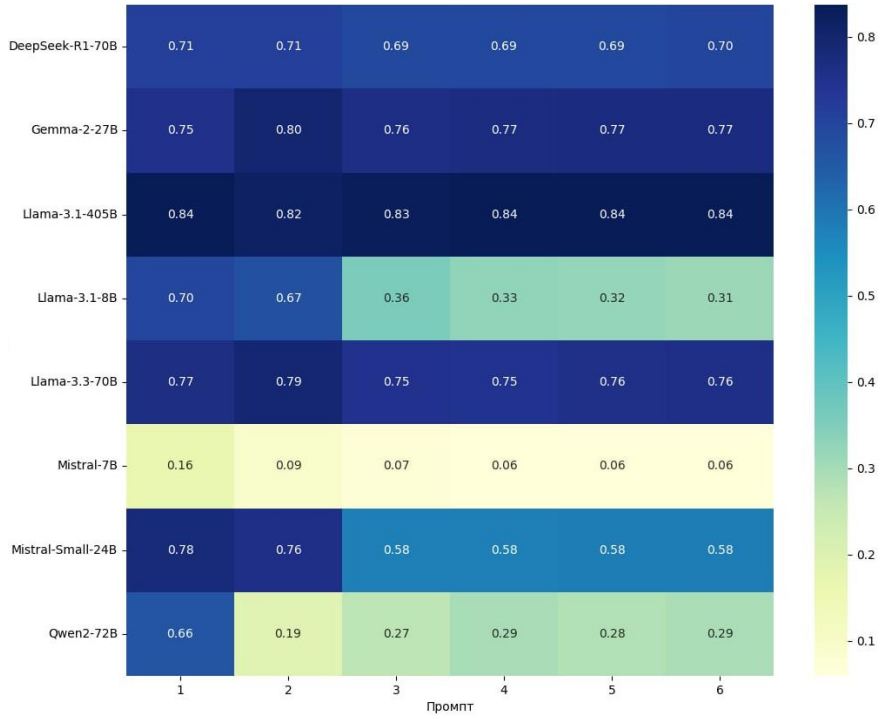

Рис. 4. Тепловая карта полученных значений оценки F-меры.
Fig. 4. Heat map of the obtained F1 score values.

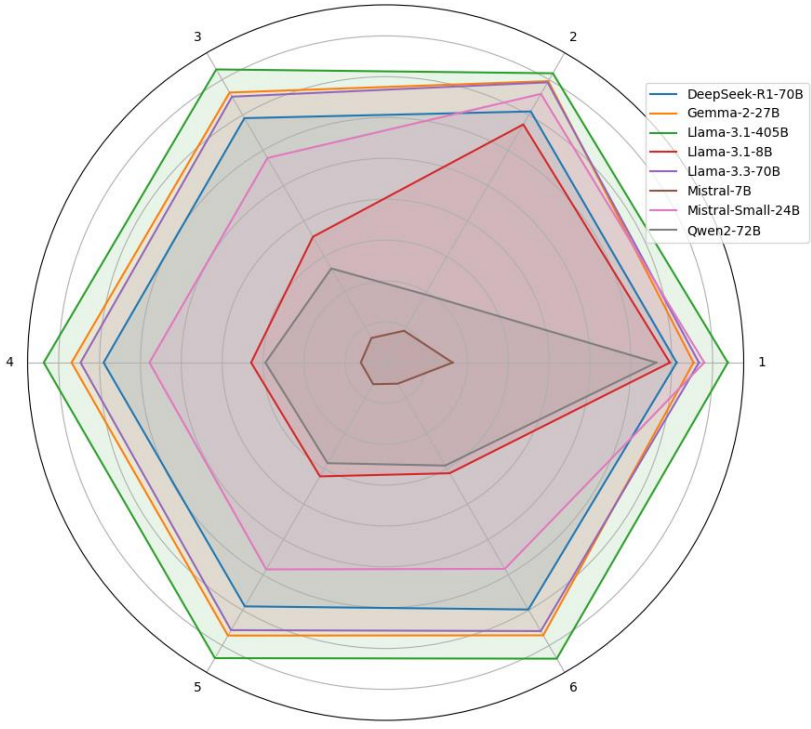

Рис. 5. Радиальная диаграмма полученных значений оценки F-меры.
Fig. 5. Radial diagram of the obtained F1 score values.

Проведенные эксперименты показали, что модели Meta-Llama-3.1-8B, Qwen2-72B и Mistral-7B демонстрируют наибольшую нестабильность при применении усложнённых текстовых подсказок. Вместо ожидаемого улучшения качества посредствам пошаговых инструкций, критериев поиска или несколькими примерами (*few-shot*), у этих моделей наблюдалось:

- **Негативное влияние многосоставных текстовых подсказок:** сложные инструкции иногда вводили модель в заблуждение, при этом общая F-мера снижалась.
- **Нестандартный формат ответов:** несмотря на явные указания в текстовых подсказках о выдаче координат заголовков, модели часто возвращали не числа (координаты), а содержимое ячеек. Вероятно, это связано с архитектурной спецификой генеративных моделей на основе трансформеров, которые оптимизированы на предсказание следующих токенов, а не числовых координат, поэтому они извлекают и возвращают текстовую часть (заголовок), а не позицию. Другими словами модель не получает достоверного сигнала о геометрической позиции и вместо этого использует наиболее вероятный паттерн в виде текста предполагаемого заголовка. Также может быть вариант, что большие языковые модели не специализируются на этой задаче и при их обучении использовались базовые таблицы с заголовком в первой строке, в связи с чем, они склонны использовать этот паттерн.
- **Необходимость адаптивного парсинга ответов:** чтобы компенсировать неконсистентные ответы, был реализован дополнительный шаг – при получении текстовых значений осуществляется их поиск в исходной таблице и сопоставление с координатами. Эта правка позволила исправить одну из самых распространённых ошибок и отразилась на итоговых оценках.

Основным выводом данного эксперимента является то, что общая ожидаемая тенденция «*чем сложнее текстовая подсказка, тем лучше результат*» не оправдала ожиданий. Скорее, наоборот: у мелких моделей избыточная детализация подсказки вызывала ухудшение распознавания, они «путались» в инструкциях. У средних и крупных моделей зависимость от сложности подсказки была менее выражена – их F-мера оставалась относительно стабильной. Таким образом, простые, чёткие текстовые подсказки оказались эффективнее для большинства моделей.

Было отмечено, что модели сильно теряли производительность, если заголовки находились не на первой строке:

- Для средних и сложных таблиц (многоуровневых иерархических заголовков с объединёнными ячейками) практически все модели показывали низкую оценку полноты, пропуская истинные заголовки, расположенные вне первой строки.
- Крупные и средние модели справлялись с такими случаями лучше, но всё равно демонстрировали заметный спад F-меры по сравнению с простыми таблицами.

На данный момент анализ основан на 237 таблицах, из которых лишь небольшая часть (20 таблиц) относится к сложным структурам. Поэтому полученные оценки в большей степени отражают поведение моделей на таблицах с более простой структурой. Для более точной оценки способности моделей работать с нетипичными макетами таблиц требуется расширить выборку за счёт увеличения числа сложных таблиц.

В целом, полученные результаты показывают, что:

- *Размер параметров модели* по-прежнему остаётся главным фактором качества.
- *Чёткость и минимализм текстовых подсказок* критичны даже для средних и крупных моделей.
- *Адаптивный парсинг ответов* обязательный шаг при использовании различных

больших языковых моделей.

- Полностью оценить *устойчивость к сложным таблицам* можно лишь после расширения выборки.

5. Заключение

В данной работе исследуются возможности больших языковых моделей распознавать заголовки таблиц, обладающих разной структурной компоновкой. Эксперименты проведены на выборке из 237 таблиц, отобранных на основе крупномасштабного корпуса PubTables-1M. Проведенное исследование демонстрирует, что модели способны решать данную задачу, но их производительность существенно варьируется в зависимости от размера модели и стратегии формирования текстовых подсказок. Крупные модели (например, Meta-Llama-3.1-405B) показывают стабильно высокие результаты (F1 ≈ 0.80–0.85), тогда как малые модели (например, Mistral-7B) значительно отстают (F1 ≈ 0.06–0.30). Усложнение текстовых подсказок за счет пошаговых инструкций, критериев поиска и примеров (*few-shot*) не всегда улучшает качество, в частности, для малых моделей это приводит к деградации результатов. Наибольшие сложности возникают при обработке таблиц с многоуровневыми заголовками и объединёнными ячейками, где даже у крупных и средних моделей снижается точность распознавания.

В будущей работе планируется расширить собранный набор данных, включив примеры таблиц, обладающих более сложной структурой, а также примеры с наличием различных шумов (ошибок, опечаток и т.п.), присутствующих изначально в выбранном корпусе PubTables-1M. Однако следует отметить, что набор данных составлен на английском языке, так как оригинальные таблицы и аннотации PubTables-1M основаны на англоязычных научных статьях. Поэтому планируется создать аналогичный набор данных на русском языке путём перевода основного набора и сбора новых таблиц, например, из русскоязычной части Википедии или GitHub. Это позволит оценить влияние языка на производительность моделей и их способность обрабатывать таблицы на разных языках. Кроме того, предлагается исследовать гибридные подходы, комбинирующие большие языковые модели с традиционными методами компьютерного зрения, для улучшения обработки визуальных особенностей таблиц. Эти шаги позволят создать более надежные системы автоматического анализа заголовков таблиц, применимые в реальных сценариях, таких как интеграция с базами данных и поддержка научных исследований.

Список литературы

[1]. Dong H., Cheng Z., He X., Zhou M., Zhou A., Zhou F., Liu A., Han S., Zhang D. Table Pre-training: A Survey on Model Architectures, Pre-training Objectives, and Downstream Tasks. Proc. the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria, 2022, pp. 5426-5435. DOI: 10.24963/ijcai.2022/761.

[2]. Badaro G., Saeed M., Papotti P. Transformers for Tabular Data Representation: A Survey of Models and Applications. Transactions of the Association for Computational Linguistics, vol. 11, 2023, pp. 227-249. DOI: 10.1162/tacl_a_00544.

[3]. Dong H., Wang Z. Large Language Models for Tabular Data: Progresses and Future Directions. Proc. the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'24), Washington, USA, 2024, pp. 2997-3000. DOI: 10.1145/3626772.3661384.

[4]. Bonfitto S., Casiraghi E., Mesiti M. Table understanding approaches for extracting knowledge from heterogeneous tables. Wiley Interdisciplinary Reviews, Data Mining and Knowledge Discovery, vol. 11, 2021, e1407. DOI: 10.1002/widm.1407.

[5]. Shigarov A. Table understanding: Problem overview. Wiley Interdisciplinary Reviews, Data Mining and Knowledge Discovery, vol. 13, 2022, e1482. DOI: 10.1002/widm.1482.

[6]. Embley D. W., Krishnamoorthy M. S., Nagy G., Seth S. Converting heterogeneous statistical tables on the web to searchable databases. International Journal on Document Analysis and Recognition, vol. 19, no. 2, 2016, pp. 119-138. DOI: 10.1007/s10032-016-0259-1.

[7]. Rastan R., Paik H.-Y., Shepherd J. TEXUS: a unified framework for extracting and understanding tables in PDF documents. Information Processing and Management: an International Journal, vol. 56, no. 3, 2019, pp. 895-918. DOI: 10.1016/j.ipm.2019.01.008.
[8]. Wu X., Chen H., Bu C., Ji S., Zhang Z., Sheng V. S. HUSS: A heuristic method for understanding the semantic structure of spreadsheets. Data Intelligence, vol. 5, no. 3, 2023, pp. 537-559. DOI: 10.1162/dint_a_00201.
[9]. Roldán J. C., Jiménez P., Corchuelo R. On extracting data from tables that are encoded using html. Knowledge-Based Systems, vol. 190, 2020, 105157. DOI: 10.1016/j.knosys.2019.105157.
[10]. Fang J., Mitra P., Tang Z., Giles C. L. Table header detection and classification. Proc. the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12), Toronto, Ontario, Canada, 2012, pp. 599-605. DOI: 10.5555/2900728.2900814.
[11]. Roldán J. C., Jiménez P., Szekely P., Corchuelo R. TOMATE: A heuristic-based approach to extract data from HTML tables. Information Sciences, vol. 577, 2021, pp. 49-68. DOI: 10.1016/j.ins.2021.04.087.
[12]. Fetahu B., Anand A., Koutraki M. TableNet: An Approach for Determining Fine-grained Relations for Wikipedia Tables. Proc. the World Wide Web Conference (WWW'19), San Francisco, CA, USA, 2019, pp. 2736-2742. DOI: 10.1145/3308558.3313629.
[13]. Xue W., Yu B., Wang W., Tao D., Li Q. TGRNet: A Table Graph Reconstruction Network for Table Structure Recognition. Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 1295-1304. DOI: 10.1109/ICCV48922.2021.00133.
[14]. Li X. H., Yin F., Dai H. S., Cheng-Lin Liu C. L. Table Structure Recognition and Form Parsing by End-to-End Object Detection and Relation Parsing. Pattern Recognition, vol. 132, no. C, 2022, DOI: 10.1016/j.patcog.2022.108946.
[15]. Lin W., Sun Z., Ma C., Li M., Wang J., Sun L., Huo Q. TSRFormer: Table Structure Recognition with Transformers. Proc. the 30th ACM International Conference on Multimedia (MM'22), New York, USA, 2022, pp. 6473-6482. DOI: 10.1145/3503161.3548038.
[16]. Yang J., Gupta A., Upadhyay S., He L., Goel R., Paul S. TableFormer: Robust Transformer Modeling for Table-Text Encoding. Proc. the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22), Dublin, Ireland, 2022, pp. 528-537. DOI: 10.18653/v1/2022.acl-long.40.
[17]. Huang Y., Lu N., Chen D., Li Y., Xie Z., Zhu S., Gao L., Peng W. Improving Table Structure Recognition with Visual-Alignment Sequential Coordinate Modeling. Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 11134-11143. DOI: 10.1109/CVPR52729.2023.01071.
[18]. Smock B., Pesala R., Abraham R. Aligning Benchmark Datasets for Table Structure Recognition. Proc. 17th International Conference of Document Analysis and Recognition (ICDAR'2023), San Jose, CA, USA, 2023, pp 371-386. DOI: 10.1007/978-3-031-41734-4_23.
[19]. Chen L., Huang C., Zheng X., Lin J., Huang X. TableVLM: Multi-modal Pre-training for Table Structure Recognition. Proc. the 61st Annual Meeting of the Association for Computational Linguistics (ACL'23), Toronto, Canada, 2023, pp. 2437-2449. DOI: 10.18653/v1/2023.acl-long.137.
[20]. Herzig J., Nowak P. K., Muller T., Piccinno F., Eisenschlos J. M. TaPas: Weakly Supervised Table Parsing via Pre-training. Proc. 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020, pp. 4320-4333. DOI: 10.18653/v1/2020.acl-main.398.
[21]. Yin P., Neubig G., Yih W. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. Proc. the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8413-8426. DOI: 10.18653/v1/2020.acl-main.745.
[22]. Zheng X., Burdick D., Popa L., Zhong X., Wang N. X. R. Global Table Extractor (GTE): A Framework for Joint Table Identification and Cell Structure Recognition Using Visual Context. Proc. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2021, pp. 697-706. DOI: 10.1109/WACV48630.2021.00074.
[23]. Jain A., Paliwal S., Sharma M., Vig L. TSR-DSAW: Table Structure Recognition via Deep Spatial Association of Words. Proc. the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2021), Online, 2021, pp. 257-262. DOI: 10.14428/esann/2021.es2021-109.
[24]. Singha A., Cambronero J., Gulwani S., Le V., Parnin C. Tabular Representation, Noisy Operators, and Impacts on Table Structure Understanding Tasks in LLMs. Proc. Table Representation Learning Workshop at NeurIPS 2023, Online, 2023, pp. 1-14. DOI: arxiv.org/abs/2310.10358.
[25]. Sui Y., Zhou M., Zhou M., Han S., Zhang D. Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study. Proc. the 17th ACM International Conference

on Web Search and Data Mining (WSDM'24), Merida, Mexico, 2024, pp. 645-654. DOI: 10.1145/3616855.3635752.
[26]. Mistral-7B-Instruct-v0.3, Available at: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3, accessed 09.05.2025.
[27]. Llama-3.1-8B-Instruct, Available at: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, accessed 09.05.2025.
[28]. Mistral-Small-24B-Instruct-2501, Available at: https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501, accessed 09.05.2025.
[29]. Gemma-2-27b-it, Available at: https://huggingface.co/google/gemma-2-27b-it, accessed 09.05.2025.
[30]. Llama-3.3-70B-Instruct, Available at: https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct, accessed 09.05.2025.
[31]. Qwen2-72B-Instruct, Available at: https://huggingface.co/Qwen/Qwen2-72B-Instruct, accessed 09.05.2025.
[32]. DeepSeek-R1-Distill-Llama-70B, Available at: https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B, accessed 09.05.2025.
[33]. Llama-3.1-405B-Instruct, Available at: https://huggingface.co/meta-llama/Llama-3.1-405B-Instruct, accessed 09.05.2025.
[34]. Smock B., Pesala R., Abraham R. PubTables-1M: Towards comprehensive table extraction from unstructured documents. Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 4634-4642. DOI: 10.1109/CVPR52688.2022.00459.
[35]. A developer's guide to prompt engineering and LLMs, Available at: https://github.blog/ai-and-ml/generative-ai/prompt-engineering-guide-generative-ai-llms/, accessed 09.05.2025.
[36]. LangChain framework, Available at: https://www.langchain.com/, accessed 09.05.2025.
[37]. Together AI, Available at: https://www.together.ai/, accessed 09.05.2025.

Информация об авторах / Information about authors

Илья Игоревич ОХОТИН – магистрант Института математики и информационных технологий Иркутского государственного университета (ИМИТ ИГУ) с 2024 года. Сфера научных интересов: большие языковые модели, работа с табличными данными, распознавание структуры таблиц; обработка заголовков таблиц.

Ilia Igorevich OKHOTIN is a master student at the Institute of Mathematics and Information Technology of Irkutsk State University (IMIT ISU) since 2024. Research interests: large language models, tabular data processing, table structure recognition; table header processing.

Никита Олегович ДОРОДНЫХ – кандидат технических наук, старший научный сотрудник Института динамики систем и теории управления им. В.М. Матросова Сибирского отделения РАН (ИДСТУ СО РАН) с 2021 года. Сфера научных интересов: автоматизация создания интеллектуальных систем и баз знаний, получение знаний на основе преобразования концептуальных моделей и электронных таблиц.

Nikita Olegovych DORODNYKH – Cand. Sci. (Tech.), senior associate researcher at the Matrosov Institute of System Dynamics and Control Theory named SB RAS (ISDCT SB RAS) since 2021. Research interests: computer-aided development of intelligent systems and knowledge bases, knowledge acquisition based on the transformation of conceptual models and tables.