

Научная статья
УДК 004.853
doi:10.37614/2949-1215.2025.16.3.006

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ ИЗВЛЕЧЕНИЯ ДАННЫХ ИЗ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Роман Александрович Горбунов¹, Александр Владимирович Вицентий²

^{1, 2}*Институт информатики и математического моделирования имени В. А. Путилова
Кольского научного центра Российской академии наук, Апатиты, Россия*

²*Филиал Мурманского арктического университета в г. Апатиты, Апатиты, Россия*

¹*gorbunov-roma@inbox.ru, <https://orcid.org/0009-0004-4627-504X>*

²*alx_2003@mail.ru, <https://orcid.org/0000-0003-1331-4749>*

Аннотация

Данная работа посвящена исследованию возможностей больших языковых моделей (LLM) для решения задач извлечения структурированных данных в формате RDF-троек из неструктурированных разнородных текстов на естественном языке. Рассматривается проблема эффективности извлечения данных, которая актуальна для автоматического построения семантических сетей, служащих основой для представления геопространственных знаний. Представлена сравнительная оценка различных типов промптинга, являющихся ключевым инструментом взаимодействия с LLM.

Ключевые слова:

большая языковая модель (LLM), граф знаний, DeepSeek, RDF-тройки, промптинг, извлечение структурированных данных, семантическая сеть

Благодарности:

исследование выполнено в рамках государственного задания Института информатики и математического моделирования имени В. А. Путилова Кольского научного центра Российской академии наук от Министерства науки и высшего образования Российской Федерации, тема научно-исследовательской работы «Методы и технологии создания интеллектуальных информационных систем для поддержки развития сложных динамических систем с региональной спецификой в условиях неопределенности и риска» (регистрационный номер 1023032300374-0-2.2.1).

Для цитирования:

Горбунов Р. А., Вицентий А. В. Исследование возможностей больших языковых моделей для извлечения данных из текстов на естественном языке // Труды Кольского научного центра РАН. Серия: Технические науки. 2025. Т. 16, № 3. С. 80–105. doi:10.37614/2949-1215.2025.16.3.006.

Original article

RESEARCH OF THE CAPABILITIES OF LARGE LANGUAGE MODELS FOR EXTRACTING DATA FROM NATURAL LANGUAGE TEXTS

Roman A. Gorbunov¹, Alexander V. Vicentiy²

^{1, 2}*Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre
of the Russian Academy of Sciences, Apatity, Russia*

²*Apatity branch of Murmansk Arctic State University, Apatity, Russia*

¹*gorbunov-roma@inbox.ru, <https://orcid.org/0009-0004-4627-504X>*

²*alx_2003@mail.ru, <https://orcid.org/0000-0003-1331-4749>*

Abstract

This paper is devoted to the study of the capabilities of large language models (LLM) for solving the problems of extracting structured data in the RDF-triples format from unstructured heterogeneous texts in natural language. The problem of data extraction efficiency is considered, which is relevant for the automatic construction of semantic networks that serve as the basis for the representation of geospatial knowledge. A comparative assessment of various types of prompting, which are a key tool for interacting with LLM, is presented.

Keywords:

large language model (LLM), knowledge graph, DeepSeek, RDF-triples, prompting, structured data extraction, semantic network

Acknowledgments:

The study was carried out within the framework of the Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences state assignment of the Ministry of Science and Higher Education of the Russian Federation, research topic “Methods and technologies for creating intelligent information systems to support the development of complex dynamic systems with regional specifics in conditions of uncertainty and risk” (registration number of the research topic 1023032300374-0-2.2.1).

For citation:

Gorbunov R. A., Vicentiy A. V. Research of the capabilities of large language models for extracting data from natural language texts. *Trudy Kol'skogo nauchnogo centra RAN. Seriya: Tekhnicheskie nauki* [Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences], 2025, Vol. 16, No. 3, pp. 80–105. doi:10.37614/2949-1215.2025.16.3.006.

Введение

Извлечение данных из неструктурированных текстов на естественном языке является одной из ключевых задач в области их обработки и одним из важных этапов для решения прикладных проблем, связанных с управлением информацией [1]. В контексте разработки технологии синтеза адаптивных геосемантических изображений на основе геопространственных знаний эта задача приобретает особую значимость [2]. Извлечение сущностей и отношений между ними актуально для автоматического построения и обогащения онтологий — спецификации репрезентативного словаря для общей области дискурса, включающей определения классов, отношений, функций и других объектов [3], которые служат формальной основой для представления геопространственных знаний. Особый интерес представляют нехудожественные тексты (новостные сводки, официальные отчеты, сообщения из обсуждений в социальных сетях). Подобные тексты, как правило, содержат в качестве основы описание некоторого события или ситуации через множество связанных между собой фактов. Например, рассматриваемые в рамках данного исследования тексты о чрезвычайных ситуациях техногенного характера включают в себя информацию о географических объектах и связях между ними в контексте произошедших событий. Получаемые при извлечении структурированные данные, такие как RDF-тройки <субъект, отношение, объект> (Resource Description Framework, RDF) [4], представляют собой элементарную семантическую единицу, формализованное высказывание для построения графовых моделей данных, где субъект и объект определены в качестве узлов графа, а отношение — в качестве связующего элемента. Подобные структуры данных позволяют разрабатываемым программным решениям «понимать» семантические связи, имеющие место между геопространственными объектами в рамках конкретной предметной области, что является фундаментом для последующего семантического моделирования и синтеза геоизображений.

Автоматическое извлечение отношений из текстов на естественном языке также рассматривается в качестве одной из возможных задач в области обработки естественного языка (Natural Language Processing, NLP). Практическая значимость подобной задачи обусловлена ее ролью в решении множества прикладных проблем. Так, в медицине извлечение временных связей из клинических текстов является инструментом доступа к обширному источнику медицинской информации, который описывает статус пациента, включая клинические заключения, процедуры и информацию о проводимом лечении [5]. В финансовом секторе извлечение связей из финансовых отчетов используется для предоставления инвесторам информации о компании, на основе которой они могут принимать соответствующие решения об инвестировании [6]. В сфере безопасности извлечение связей из текстовых записей о чрезвычайных ситуациях в работе городского железнодорожного транспорта обеспечивает эффективную справочную информацию для принятия решений [7]. В качестве основных типовых задач извлечения данных можно обозначить несколько примеров. Распознавание именованных сущностей (Named Entity Recognition, NER) — идентификация и классификация определенных категорий слов в тексте, например имен людей, названий организаций, даты и времени, числовых и денежных выражений, где одной из специализированных вариаций является извлечение топонимов (наименований географических объектов). Извлечение отношений (Relation Extraction, RE) — определение и извлечение семантических связей между выделенными сущностями. Определение семантических ролей (Semantic Role Labeling, SRL) — процесс определения роли и отношения каждого слова в контексте предложения. Исторически

для решения подобных задач применяется широкий спектр методов, эволюция которых демонстрирует переход от жестких правил к нейросетевому обучению. К таким методам можно отнести, например, методы на основе словарей и правил, которые, несмотря на высокую точность, требуют большого объема ручного труда экспертов и плохо масштабируются; статистические методы, где используются размеченные корпуса текстов для автоматического сбора данных, но которые часто создают проблемы с точностью из-за зашумленности данных; нейросетевые модели, в том числе предобученные, которые стали современным стандартом, демонстрируют впечатляющие результаты по извлечению данных за счет способности учитывать контекст. Такой современный подход часто используется для решения задач извлечения сущностей и отношений [8; 9], поэтому он может отлично подойти для решения задачи извлечения множества RDF-троек из текстов на естественном языке, представленной в данном исследовании.

Высокоперспективным направлением в решении задач извлечения данных из текстов на естественном языке является применение больших языковых моделей (Large Language Model, LLM). Они представляют собой предобученные на значительных объемах информации нейросетевые инструменты для обработки запросов с возможностью выявления зависимостей между словами в последовательности. Предварительное обучение позволяет им не только генерировать связанный текст, но и решать описанные ранее сложные семантические задачи. Эволюция LLM — от ранних концептов до современных высокоразвитых систем — привела к появлению множества мощных моделей, доступных через API (Application Programming Interface, API) или чат-интерфейсы. Среди популярных примеров можно выделить ChatGPT (OpenAI) [10], Claude (Anthropic) [11], Gemini (Google) [12]. Активно развиваются и отечественные разработки, включая GigaChat (Сбер) [13] и YandexGPT (Яндекс) [14]. В подобных моделях ключевым элементом взаимодействия является промпт (англ. prompt — подсказка), а под промптингом (prompting) понимается методология составления текстовых инструкций (промптов), направляющих модель при решении конкретных задач.

Поскольку LLM по своей сути являются универсальными предсказателями слов в последовательности, то именно промпт фокусирует их вычислительные возможности на требуемой операции, такой как извлечение RDF-троек. При этом эффективность извлечения данных напрямую зависит от выбранной стратегии промптинга, которая помогает экспертам ответить на вопрос «Какой способ подачи информации модели является наиболее эффективным для решения поставленной задачи?», и типа промптинга для ответа на вопрос «В какой форме представить модели инструкцию для эффективного решения поставленной задачи?». Таким образом, стратегия определяет методику подачи информации, а тип — структуру инструкции.

В работе рассматривается проблема оценки влияния выбора стратегии промптинга на эффективность извлечения структурированных данных из текстов на естественном языке с помощью применения больших языковых моделей. При этом ключевой гипотезой исследования является предположение о значительном влиянии выбранных стратегий и типов промптинга на эффективность извлечения структурированных данных из нехудожественных разнородных текстов. В настоящее время LLM демонстрируют революционные возможности для решения задач обработки данных, выступая мощным инструментом для их извлечения. Их применение для обработки разнородных текстовых данных является не только практическим инструментом, но и объектом исследования, поскольку эффективность работы LLM сильно зависит от методики взаимодействия пользователя с конкретной реализацией такой модели при использовании структурированных промптов. Существует множество различных классификаций и подходов к конструированию промптов, предложенных различными научными группами [15–17], однако, несмотря на это, нет единого подхода к стандартизации типов промптов, а также общей точки зрения на их эффективность в задачах извлечения данных из текстов на естественном языке.

В научной сфере представлены различные примеры того, где LLM используются для извлечения данных, включая извлечение данных для получения информации о вредителях в сельском хозяйстве [18], извлечение структурированных табличных данных из текстовых медицинских отчетов [19], а также извлечение структурированных данных для проведения химических исследований [20]. Существующие решения на основе LLM показывают, что, несмотря на их высокий потенциал, результаты извлечения данных могут значительно варьироваться в зависимости

от формулировки промпта, используемой модели, формата и вида текста, а также специфики предметной области. На основе представленных сведений можно сделать вывод о том, что существующие типы промптинга требуют систематизированного анализа для последующего эффективного и точного извлечения RDF-троек для решения задач обработки разнородных текстов на естественном языке, что является основным стимулом для проведения настоящего исследования. Основной целью данной работы является исследование и систематизация имеющихся стратегий и типов промптинга для понимания их эффективности при извлечении RDF-троек из разнородных текстовых данных в рамках глобальной темы по формированию геопространственных знаний.

Материалы и методы

В рамках данной работы проводится исследование типов промптов, относящихся к различным стратегиям (техникам) промптинга для проверки их эффективности при извлечении RDF-троек из разнородных текстов на естественном языке. Методика включает в себя следующие этапы: отбор исходного текстового экземпляра, который является основой для выполняемого языковой моделью задания; проектирование структуры задания по преобразованию текста в набор структурированных RDF-троек; систематизация техник и типов промптинга и последующее составление универсальных шаблонов для взаимодействия с языковой моделью; сборка и отправка заданий языковой модели по преобразованию текста в набор RDF-троек по подготовленным шаблонам; выборка наиболее эффективных стратегий промптинга.

Для обеспечения репрезентативности результатов выбран текст о чрезвычайной ситуации техногенного характера, повествующий о железнодорожной аварии на станции Княжая, содержащий описание некоторого множества географических объектов. Текст обладает характерными для подобного вида повествования признаками, такими как хронологическая последовательность изложения, наличие конкретных географических ориентиров, упоминание участников события и описание динамических взаимодействий между объектами. Хронология событий четко выражена от первоначальной остановки грузового состава на уклоне до последующего самопроизвольного движения и столкновения с пассажирским поездом. Географические ориентиры представлены многоуровневой системой локализации от станции Княжая до Мурманской области, что демонстрирует иерархическую организованность географического пространства. Участники события включают как физических лиц (машинист и помощник), так и технические объекты (грузовой состав, пассажирский поезд), между которыми описываются различные виды взаимодействий через систему глагольных конструкций: «укатился с перегона», «врезался в хвост», «повредив третий вагон». Текст также содержит разнообразные типы отношений между различными категориями сущностей, таких как пространственные («на станции Княжая», «недалеко от станции»), временные («18 декабря 2024 года», «спустя полтора часа»), причинно-следственные («вместо 33 башмаков — лишь пять» — «воздух ушёл» — «состав начал двигаться» — «столкновение»).

Текст, описывающий событие железнодорожного происшествия, который был использован в качестве экземпляра для анализа эффективности извлечения RDF-троек различными типами промптов, представлен ниже.

Использованный текст: *«Столкновение поездов на станции Княжая — железнодорожная авария, произошедшая 18 декабря 2024 года на станции Княжая в посёлке Зеленоборский Кандалакшского района Мурманской области России. Грузовой состав № 2013 задним ходом самопроизвольно укатился с перегона, в результате чего произошло боковое столкновение с хвостом стоящего на станции пассажирского поезда № 11 Мурманск — Санкт-Петербург. Погибли 2 человека, пострадал 31. По предварительным данным, грузовой поезд недалеко от станции Княжая остановился на 17-тысячном уклоне, но вместо положенных 33 тормозных башмаков его зафиксировали лишь пятью. В результате спустя полтора часа воздух из тормозной системы ушел, и состав начал самопроизвольно двигаться вниз, набирая скорость. Попытки остановить его оказались безуспешными, и грузовой поезд врезался в хвост пассажирского состава пассажирского поезда № 11 «Санкт Петербург — Мурманск», повредив третий с конца вагон. В результате столкновения погибли два человека, более тридцати пострадали (из них 5 детей; тяжёлые ранения получили четверо). Следствие задержало двух сотрудников — 45-летнего машиниста*

Алексея Зычкова и его 23-летнего помощника Рамиля Садыгова. Во время допроса они признались, что закрепили состав минимальным количеством багмаков и ожидали помощи, не учитывая риск, связанный с уклоном пути. По факту трагедии возбуждено уголовное дело по статье 263 УК РФ — «Нарушение правил безопасности движения и эксплуатации железнодорожного транспорта, повлекшее гибель людей». Пока единственным подозреваемым остаётся машинист грузового состава. До устранения последствий аварии ожидалась задержки в движении трёх пассажирских поездов.» [21].

Одним из важных шагов исследования является проектирование структуры унифицированного задания по преобразованию неструктурированного текста в набор RDF-троек. На этом этапе разработана четкая, краткая формулировка задания без детализированных правил извлечения, обеспечивающая стандартизацию взаимодействия с языковой моделью. Разработанная формулировка задания выглядит следующим образом: *«Проведите разбиение заданного текста на триплеты. После разбиения проведите подсчёт количества полученных триплетов. Ответ на данное задание предоставьте в следующем формате: «Триплеты: [субъект, отношение, объект], ...; Количество триплетов: ».*

Такой подход выбран по нескольким причинам. Во-первых, это позволило оценить способность языковой модели к самостоятельному выявлению семантических связей без внешних подсказок о том, какие именно типы отношений считаются релевантными. Во-вторых, такой подход способствовал существенному уменьшению количества используемых токенов при отправке запросов и получении ответа от языковой модели, что повысило эффективность использования ресурсов для взаимодействия с ней. Необходимо отметить тот факт, что краткость формулировки позволила более детально исследовать те виды промптов, где подразумевается личное рассуждение самой языковой модели, поскольку некоторые инструкции не навязывали готовых шаблонов анализа и оставляли пространство для проявления ее собственных аналитических возможностей. Данный подход выявил готовность языковой модели к работе в условиях неполной спецификации задачи и подтвердил ее способность восполнять недостающую информацию за счет собственных языковых знаний. Стандарт построения RDF-троек <субъект, отношение, объект> является семантически прозрачным, но при этом достаточно абстрактным, что создает необходимые условия для креативности языковой модели в интерпретации текста при сохранении базовой структурной согласованности. Требование подсчета количества RDF-троек послужило дополнительным критерием контроля выполнения заданий и позволило количественно оценить эффективность различных стратегий промптинга.

Для проведения экспериментальной части исследования была выбрана большая языковая модель DeepSeek-V3.2 [22]. Выбор был обусловлен комплексом практических и методологических соображений. Ключевыми факторами стали доступность и экономическая целесообразность использования. Модель DeepSeek предоставляет доступ к чату с достаточно широкими лимитами использования, что делает ее доступной для научных исследований без необходимости привлечения дополнительного финансирования. Это особенно важно при тестировании типов промптов, для которых требуется выполнение нескольких последовательных запросов к модели, требующих повышенного количества токенов на ответ. С технической точки зрения модель основана на современной трансформерной архитектуре и демонстрирует конкурентные показатели в различных бенчмарках [23–25], что обеспечивает релевантность получаемых результатов при решении задач по извлечению структурированных данных.

Предварительные тесты подтвердили, что данная большая языковая модель может эффективно справляться с задачами по структурированию данных и извлечению RDF-троек из текстов на русском языке, проявляя адекватную производительность при обработке нехудожественных разнородных текстов с включением геопространственной семантики. Важно подчеркнуть, что в современных условиях существует множество альтернативных языковых моделей, а также отечественных разработок, о которых упоминалось ранее. Такие модели потенциально также могли бы быть использованы в данном исследовании, однако принципиальный выбор конкретной реализации модели не является критически важным для достижения основной цели работы, поскольку фокус исследования направлен в сторону методологии промптинга, а не сравнительного анализа архитектурных особенностей или возможностей различных моделей. В контексте данного исследования LLM рассматривается прежде всего как инструмент-посредник для реализации различных стратегий и типов промптинга. Такой подход позволяет абстрагироваться от специфических особенностей конкретных моделей и сфокусироваться на универсальных принципах конструирования промптов.

Процесс взаимодействия с языковой моделью DeepSeek был организован через веб-интерфейс чата, доступного на официальном сайте разработчика. Для каждого экспериментального прогона использовался новый отдельный сеанс чата, что исключало влияние предыдущего контекста на результаты работы модели. Важно отметить, что дополнительные опции платформы, такие как «глубокое мышление», которое позволяет нейросети решать задачи поэтапно с подробным описанием мыслительного процесса, и «поиск в интернете», который позволяет нейросети исследовать актуальные интернет-ресурсы, не применялись при проведении экспериментальных прогонов для обеспечения чистоты исследования и исключения факторов, которые могли бы исказить оценку эффективности типов промптинга. Каждый экспериментальный прогон выполнялся в идентичных условиях, что обеспечило сопоставимость данных, полученных при тестировании различных типов промптинга.

Структура отправляемого запроса языковой модели (рис. 1) представляет собой целостный объект, состоящий из нескольких блоков. Поставленная задача включает в себя текст задания и анализируемый моделью экземпляр текста о происшествии на станции Княжая. Запрос, кроме задачи, включает в себя также модификатор, образованный от типа промптинга. Сформированный экземпляр запроса вводится в чат модели единым сообщением, где инструкции модификатора могут располагаться в начале или в конце запроса или (в редких случаях) могут быть разделены на две части, где одна часть располагается в начале, а другая в конце запроса.

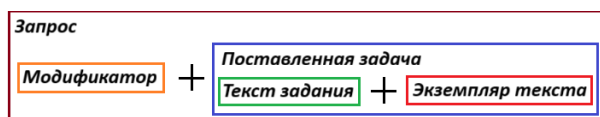


Рис. 1. Общая структура отправляемого запроса языковой модели

Далее представлен пример типичного процесса взаимодействия с большой языковой моделью, наглядно демонстрирующий полный цикл обработки запроса. На рис. 2 показан пример пользовательского промпта, введенного в интерфейсе чата DeepSeek, где был применен модификатор типа Zero-Shot Chain-of-Thought, о чем явно свидетельствует инструкция модели решить задачу шаг за шагом.

На рис. 3 зафиксирован процесс генерации ответа, где интерфейс чата отображает последовательную цепочку логических рассуждений модели, выполняемую в соответствии с полученной инструкцией. На рис. 4 представлен финальный этап, где DeepSeek дает окончательный ответ, который был сгенерирован в строгом соответствии с заданным форматом. В дальнейшем сгенерированный ответ подвергался экспертному анализу.

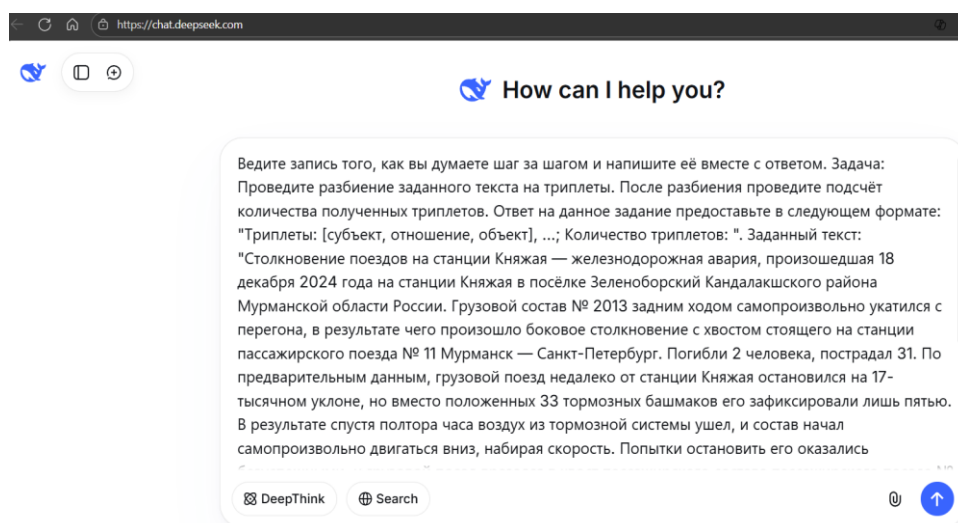


Рис. 2. Пример промпта в интерфейсе чата DeepSeek с запросом Zero-Shot Chain-of-Thought

Шаг 3: Форматирование в триплеты	
Хорошо, давайте разберём задачу шаг за шагом.	[Столкновение поездов на станции Княжая, является, железнодорожная авария]
Шаг 1: Понимание задачи	[Столкновение поездов на станции Княжая, произошло, 18 декабря 2024 года]
Нужно извлечь из текста триплеты в формате [субъект, отношение, объект], а затем посчитать их количество.	[Столкновение поездов на станции Княжая, место, станция Княжая]
Триплет — это факт, где есть субъект (кто/что), отношение (что делает/как связано) и объект (с кем/с чем/куда).	[Станция Княжая, находится в, посёлок Зеленоборский]
Шаг 2: Выделение основных фактов из текста	[Посёлок Зеленоборский, находится в, Кандалакшский район]
	[Кандалакшский район, находится в, Мурманская область]
	[Мурманская область, находится в, Россия]
	[Грузовой состав № 2013, укатился задним ходом, с перегона]
	[Грузовой состав № 2013, столкнулся с, пассажирский поезд № 11]
	[Пассажирский поезд № 11, маршрут, Мурманск — Санкт-Петербург]
	[Авария, погибшие, 2 человека]
	[Авария, пострадавшие, 31 человек]
	[Грузовой поезд, остановился на, 17-тысячный уклон]
	[Грузовой поезд, должен был быть зафиксирован, 33 тормозных башмака]
	[Грузовой поезд, зафиксирован, 5 тормозных башмаков]
	[Через 1,5 часа, воздух из тормозной системы, ушёл]
	[Грузовой состав, начал двигаться, вниз]
	[Грузовой состав, набирал, скорость]
	[Попытки остановить грузовой состав, результат, безуспешны]
	[Грузовой поезд, врезался в, хвост пассажирского поезда]
	[Грузовой поезд, повредил, третий с конца вагон]
	[Следствие, задержало, машинист Алексей Зычков]
	[Следствие, задержало, помощник Рамиль Садыгов]
	[Алексей Зычков, возраст, 45 лет]
	[Рамиль Садыгов, возраст, 23 года]
	[Они, признались, закрепили состав минимальным количеством башмаков]
	[Они, не учитывали, риск уклона пути]
	[По факту трагедии, возбуждено, уголовное дело]
	[Уголовное дело, статья, 263 УК РФ]
	[Статья 263 УК РФ, название, Нарушение правил безопасности движения и эксплуатации железнодорожного транспорта, повлекшее гибель людей]
	[Подозреваемый, машинист грузового состава]
	[До устранения последствий, задержки в движении, трёх пассажирских поездов]
	Шаг 4: Подсчёт триплетов
	Выше получилось 32 триплета.

Рис. 3. Интерфейс чата DeepSeek с рассуждением модели во время ответа на запрос

Шаг 5: Оформление ответа

Триплеты: [Столкновение поездов на станции Княжая, является, железнодорожная авария], [Столкновение поездов на станции Княжая, произошло, 18 декабря 2024 года], [Столкновение поездов на станции Княжая, место, станция Княжая], [Станция Княжая, находится в, посёлок Зеленоборский], [Посёлок Зеленоборский, находится в, Кандалакшский район], [Кандалакшский район, находится в, Мурманская область], [Мурманская область, находится в, Россия], [Грузовой состав № 2013, укатился задним ходом, с перегона], [Грузовой состав № 2013, столкнулся с, пассажирский поезд № 11], [Пассажирский поезд № 11, маршрут, Мурманск — Санкт-Петербург], [Авария, погибшие, 2 человека], [Авария, пострадавшие, 31 человек], [Грузовой поезд, остановился на, 17-тысячный уклон], [Грузовой поезд, должен был быть зафиксирован, 33 тормозных башмака], [Грузовой поезд, зафиксирован, 5 тормозных башмаков], [Через 1,5 часа, воздух из тормозной системы, ушёл], [Грузовой состав, начал двигаться, вниз], [Грузовой состав, набирал, скорость], [Попытки остановить грузовой состав, результат, безуспешны], [Грузовой поезд, врезался в, хвост пассажирского поезда], [Грузовой поезд, повредил, третий с конца вагон], [Следствие, задержало, машинист Алексей Зычков], [Следствие, задержало, помощник Рамиль Садыгов], [Алексей Зычков, возраст, 45 лет], [Рамиль Садыгов, возраст, 23 года], [Они, признались, закрепили состав минимальным количеством башмаков], [Они, признались, ожидали помощи], [Они, не учитывали, риск уклона пути], [По факту трагедии, возбуждено, уголовное дело], [Уголовное дело, статья, 263 УК РФ], [Статья 263 УК РФ, название, Нарушение правил безопасности движения и эксплуатации железнодорожного транспорта, повлекшее гибель людей], [Подозреваемый, машинист грузового состава], [До устранения последствий, задержки в движении, трёх пассажирских поездов];
Количество триплетов: 32

Рис. 4. Интерфейс чата DeepSeek с итоговым ответом модели на запрос

Для комплексной оценки результатов извлечения структурированных данных из текста на естественном языке применялись количественные и качественные показатели эффективности извлечения RDF-троек. Оценка указанных показателей проводилась с помощью привлечения экспертов, что обеспечивает надежность результатов. Количественная оценка включала подсчет общего количества извлеченных из текста RDF-троек. Данная информация была напрямую получена от модели и уже была включена в ее ответ. Особое внимание уделялось подсчету количества встречающихся в ответе модели уникальных RDF-троек, которое позволяет исключить дублирование семантических связей, что повысило точность количественной оценки. Качественная оценка включала определение информационной ценности извлеченных RDF-троек. Основными показателями являлись общее соответствие извлеченных связей фактическому содержанию исходного текста, а также извлечение наиболее значимых, а не поверхностных или второстепенных связей.

Представленная методика обеспечивает надежную основу для получения достоверных результатов исследовательской части работы и может быть адаптирована для решения схожих задач в области обработки и извлечения данных из текстовых источников.

Результаты

В рамках данного исследования проведена оценка эффективности 59 типов промптинга, относящихся к 6 основным стратегиям (техникам) промптинга, для решения задач извлечения RDF-троек из разнородных текстов на естественном языке. Все исследованные типы промптинга классифицированы в соответствии с представленным набором стратегий: 1) Zero-Shot Prompting (типы 1–6); 2) Few-Shot Prompting (типы 7–13); 3) Thought Generation Prompting (типы 14–30); 4) Ensembling Prompting (типы 31–40); 5) Self-Criticism Prompting (типы 41–47); 6) Decomposition Prompting (типы 48–59).

Визуальное представление разработанной классификации представлено на рис. 5.

Для каждого типа промптинга составлен и апробирован специальный модификатор — дополнительный текстовый блок, интегрируемый в экземпляр запроса. Результаты применения каждого модификатора, выраженные в количестве извлеченных RDF-троек, а также пометки

о применимости модификатора систематизированы в сводной таблице, в которой аббревиатура НП, означает, что данный тип неприменим к поставленной задаче, ТПР указывает на то, что для использования этого модификатора необходима программная реализация алгоритма или формальной логики, составляющей основу метода для получения результатов извлечения RDF-троек в данном типе промптинга, а ТЭП — на требование предоставления экспертных примеров.

Типы промптинга, обозначенные в таблице как неприменимые к поставленной в исследовании задачи извлечения RDF-троек, были исключены из данного исследования. Например, типы, предназначенные для машинного перевода, анализа диалогов или работы с визуальными данными, принципиально не могут быть адаптированы для решения поставленной задачи без полного искажения их исходной концепции.

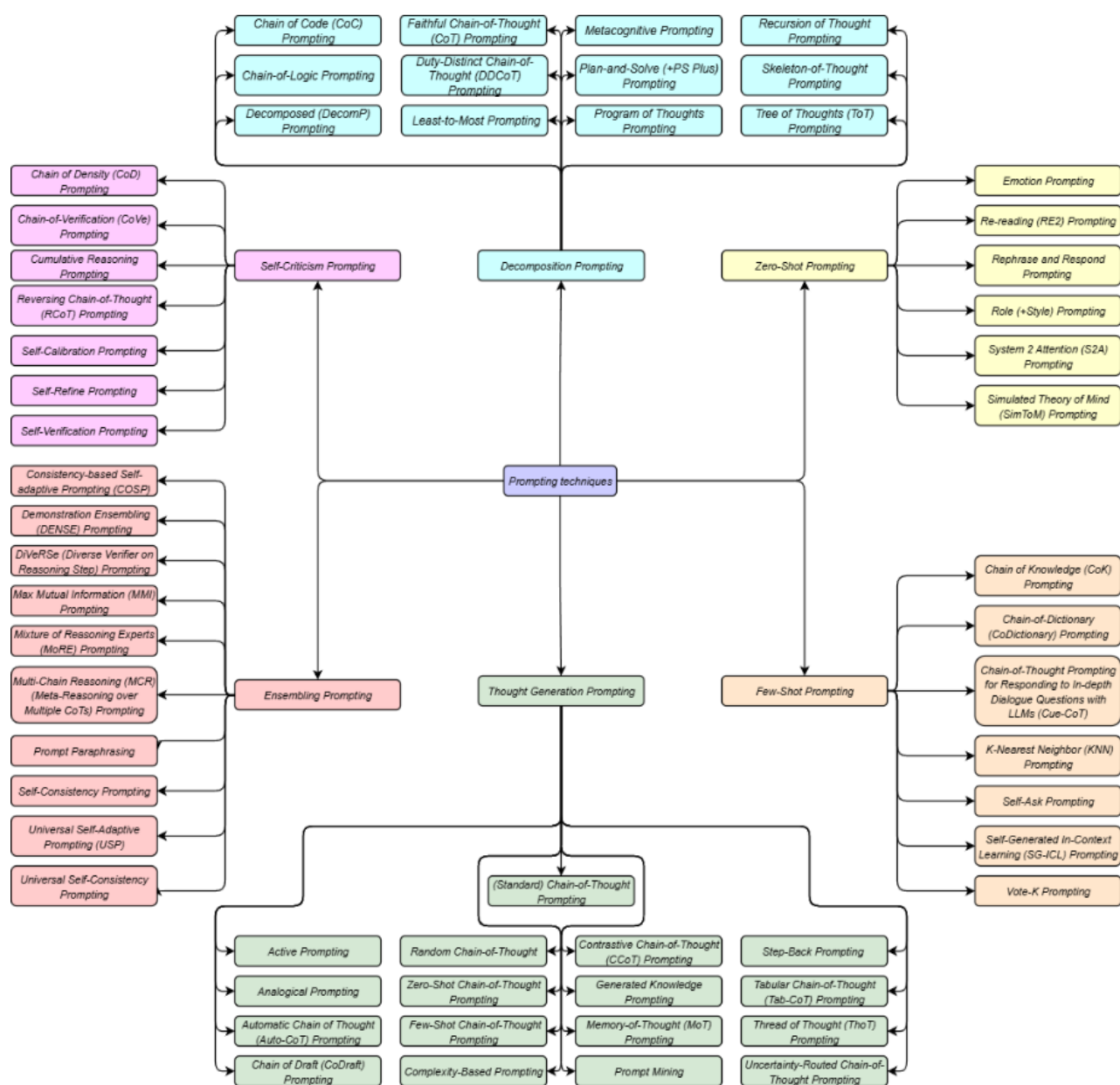


Рис. 5. Классификация стратегий и типов промптинга

Сводная таблица стратегий и типов промптинга

Техника промптинга	№	Тип промптинга	Модификатор промптинга	Количество извлеченных RDF-троек
1	2	3	4	5
Zero-Shot Prompting	1	<i>Emotion Prompting</i>	"Прошу помогите мне! От успешности выполнения данного задания зависит положение в моей карьере! Задача: [Поставленная задача]"	45
	2	<i>Re-reading (RE2) Prompting</i>	"Задача: [Поставленная задача]; Прочитайте задачу ещё раз: [Поставленная задача]"	46
	3	<i>Rephrase and Respond (RaR) Prompting</i>	"Задача: [Поставленная задача]; Перефразируйте и разверните задачу, а затем напишите ответ."	44
	4	<i>Role (+Style) Prompting</i>	"Вы являетесь [Непрямое указание роли]; Задача: [Поставленная задача]"; {Пример роли: "Вы решили стать одним из самых успешных учёных России в области ИТ, и это возможно, если вы, в меру своих научных познаний, зная, что вы являетесь старшим научным сотрудником, справитесь со следующей задачей"}"	37
	5	<i>Simulated Theory of Mind (SimToM) Prompting</i>	"Ниже приведена последовательность событий: [Последовательность событий поставленной задачи]; О каких событиях знает [имя/название основного персонажа/элемента]?", "Составьте рассказ с точки зрения [имя/название основного персонажа/элемента]; Ответьте на следующий вопрос: [Главный вопрос поставленной задачи]"	НП
	6	<i>System 2 Attention (S2A) Prompting</i>	"Получив текст от пользователя, извлеките ту часть, которая является непредвзятой, а не его мнение, чтобы использование только этого текста было хорошим контекстом для предоставления непредвзятого ответа на вопросительную часть текста. Укажите фактический вопрос или запрос пользователя. Разделите его на две категории, помеченные как "Непредвзятый текстовый контекст (включает весь контент, кроме предвзятости пользователя):" и "Вопрос/Запрос (не включает предвзятость/предпочтения пользователя):". После, дайте ответ на полученный Вопрос/Запрос. Текст пользователя: [Поставленная задача]"	48

Продолжение таблицы

1	2	3	4	5
Few-Shot Prompting	7	<i>Chain of Knowledge (CoK) Prompting</i>	"Задача: [Пример задачи]; Доказательство: 1. [субъект, отношение, объект], 2. [субъект, отношение, объект], 3. [субъект, отношение, объект] ...; Объяснение: [Обоснование логических связей между доказательствами]; Ответ: [Заключение]; Задача: [Поставленная задача]"	НП
	8	<i>Chain-of-Dictionary (CoDictionary) Prompting</i>	"Переведите следующий текст с [исходный язык] на [целевой язык]: [исходное предложение]; Подсказка для перевода: Переведите следующий текст с [исходный язык] на [целевой язык]: [исходное предложение]; [слово X на языке оригинала] означает [слово X на языке перевода] означает [слово X на вспомогательном языке 1] означает [слово X на вспомогательном языке 2]"	НП
	9	<i>Chain-of-Thought Prompting for Responding to In-depth Dialogue Questions with LLMs (Cue-CoT)</i>	{O-Cue CoT (One-step Cue Chain-of-Thought):} "Здесь происходит диалог между пользователем и системой. [Контекст диалога]; Сначала выведите одну строку, содержащую статус пользователя, такой как черты характера пользователя, его психологические и эмоциональные состояния, проявленные в разговоре. В следующей строке, пожалуйста, сыграйте роль системы и сгенерируйте ответ на основе статуса пользователя и контекста диалога; Задача: [Поставленная задача]"; {M-Cue CoT (Multi-Step Cue Chain-of-Thought):} "[Копия O-Cue CoT без поставленной задачи]; Здесь происходит диалог между пользователем и системой. [Контекст диалога]; Вот статус пользователя. [Извлечённый моделью статус пользователя]; Пожалуйста, играйте роль системы и генерируйте ответ на основе статуса пользователя и контекста диалога. Задача: [Поставленная задача]"	НП
	10	<i>K-Nearest Neighbor (KNN) Prompting</i>	{Банк примеров: Запрос (Q) - Ответ (A)}, {Значение K метода KNN}; "[Пример N1(k): Q - A, N2(k): Q - A, ...]; Задача: [Поставленная задача]"	ТПР
	11	<i>Self-Ask Prompting</i>	"Задача: [Пример задачи]; Нужны ли здесь дополнительные вопросы: Да; Продолжение: [Подзадача 1] Промежуточный ответ: [Верный ответ на подзадачу 1]; Последующие действия: [Подзадача 2]; Промежуточный ответ: [Верный ответ на подзадачу 2]; Таким образом, окончательный ответ: [Верный ответ на пример задачи]; Задача: [Поставленная задача]; Здесь необходимы дополнительные вопросы: "	ТЭП

Продолжение таблицы

1	2	3	4	5
Few-Shot Prompting	12	<i>Self-Generated In-Context Learning (SG-ICL) Prompting</i>	"Задача: [Пример задачи 1]; Ответ: [Верный ответ на пример задачи 1]; Задача: [Пример задачи схожего с 1]: ", "Задача: [Пример задачи 2]; Ответ: [Верный ответ на пример задачи 2]; Задача: [Пример задачи схожего с 2]: ", "Задача: [Пример задачи схожего с 1]; Ответ: [Ответ модели на пример задачи схожего с 1]; Задача: [Пример задачи схожего с 2]; Ответ: [Ответ модели на пример задачи схожего с 2]; Задача: [Поставленная задача]"	ТЭП
	13	<i>Vote-K Prompting</i>	{Банк примеров: Запрос (Q) - Ответ (A)}, {Значение K метода Vote-K}; "[Пример N1(k): Q - A, N2(k): Q - A, ...] Задача: [Поставленная задача]"	ТПР
Thought Generation Prompting	14	<i>Active Prompting</i>	{Банк примеров задач (Q)} {Uncertainty Estimation: (Uncertainty = H/K, H - количество уникальных ответов)} "[Провести K раз Standard CoT или Zero-Shot CoT или Few-Shot CoT над каждым примером задачи из банка задач]", {Selection: ($U \uparrow \Rightarrow 1$ (Сложная задача - берём на доработку), $U \downarrow \Rightarrow 0$ (Лёгкая задача - отбрасываем))}, {Annotation: (Создаём вручную рассуждение над сложной задачей через Chain-of-Thought)}, {Inference:} "[Пример(ы) сложной(ых) задач(и) с рассуждением]; 3 адача: [Поставленная задача]"	ТПР
	15	<i>Analogical Prompting</i>	"Задача: [Поставленная задача]; Инструкция: 1) Релевантные задачи: Вспомните примеры задач, которые имеют отношение к исходной задаче. Ваши задачи должны отличаться друг от друга и от исходной задачи (например, с разными числами и именами). 2) После написания "Q: ", опишите задачу, после "A: ", поясните решение. Решите первоначальную задачу: "	33
	16	<i>Automatic Chain of Thought (Auto-CoT) Prompting</i>	{Банк примеров задач (Q)} {Clustering Questions: (Реализация кластеризации при помощи Sentence-BERT)}, {Generating Reasoning Chains:} "[Провести N раз Zero-Shot CoT над каждым примером репрезентативной задачи из каждого кластера банка задач]; [Пример(ы) сложной(ых) задач(и) с рассуждением]; Задача: [Поставленная задача]"	ТПР
	17	<i>Chain of Draft (CoDraft) Prompting</i>	"Ведите запись того, как вы думаете шаг за шагом и напишите её вместе с ответом. Составляйте только минимальные записи для каждого шага мышления отдельно от ответа, максимум 5 слов; Задача: [Поставленная задача]"	17

Продолжение таблицы

1	2	3	4	5
Thought Generation Prompting	18	<i>(Standard) Chain-of-Thought Prompting</i>	"Задача: [Пример задачи]; Рассуждение: 1. ..., 2. ..., 3. ...; Окончательный ответ: [Верный ответ на пример задачи]; Задача: [Поставленная задача]"	ТЭП
	19	<i>Zero-Shot Chain-of-Thought Prompting</i>	"Ведите запись того, как вы думаете шаг за шагом и напишите её вместе с ответом. Задача: [Поставленная задача]"	42
	20	<i>Few-Shot Chain-of-Thought Prompting</i>	"Задача: [Пример задачи 1]; Рассуждение: 1. ..., 2. ..., 3. ...; Окончательный ответ: [Верный ответ на пример задачи 1]; Задача: [Пример задачи 2]; Рассуждение: 1. ..., 2. ..., 3. ...; Окончательный ответ: [Верный ответ на пример задачи 2]; Задача: [Поставленная задача]"	ТЭП
	21	<i>Complexity-Based Prompting</i>	{Банк примеров: Запрос (Q) - Рассуждение (R) - Ответ (A)} {Select Complex Prompts: (Выборка примеров с большим количеством шагов рассуждения)}, {Generate Multiple Outputs: (Выборка рассуждений на поставленную задачу с большим количеством шагов)} "[Провести N раз Standard CoT над поставленной задачей в модели обученной на выбранных сложных примерах]", {Vote Among Complex Chains: (Выборка окончательного ответа из отобранных с большим количеством шагов по более частым ответам)}	ТПР
	22	<i>Contrastive Chain-of-Thought (CCoT) Prompting</i>	"Задача: [Пример задачи]; Верное рассуждение: 1. ..., 2. ..., 3. ...; Верный окончательный ответ: [Верный ответ на пример задачи]; Неверное рассуждение: 1. ..., 2. ..., 3. ...; Неверный окончательный ответ: [Неверный ответ на пример задачи]; Задача: [Поставленная задача]"	ТЭП
	23	<i>Generated Knowledge Prompting</i>	{Single prompt approach:} "Соберите и выпишите факты по теме поставленной задачи, решите её, воспользовавшись ими; Задача: [Поставленная задача]" или {Dual prompt approach:} "Собери факты по теме поставленной задачи; Задача: [Поставленная задача с полученными от модели фактами]"	36
	24	<i>Memory-of-Thought (MoT) Prompting</i>	{Банк примеров: Запрос (Q) - Ответ (A)}, {Pre-thinking: (Модель занимается построением нескольких цепочек мыслей к каждому примеру и отбирает самые точные)}, "Решите поставленную задачу, опираясь на прошлые размышления по отобранному набору примеров. Задача: [Поставленная задача]", {Recalling: (Модель извлекает релевантные мысли из памяти, основываясь на сходстве между текущим вопросом и сохраненными)}	ТПР

Продолжение таблицы

Thought Generation Prompting	25	<i>Prompt Mining</i>	{Prompt Generation: (Создание набора шаблонов [субъект, отношение, объект] по изначальной задаче из внешнего источника данных)}, {Prompt Selection: (Отбор лучших шаблонов по определённым метрикам)}, "Задача: [Оптимизированная по отобранным шаблонам поставленная задача]"	НП
	26	<i>Random Chain-of-Thought</i>	"Задача: [Пример задачи из внешнего источника данных]; Рассуждение: [Цепочки рассуждений к примеру задачи из внешнего источника данных]; Окончательный ответ: [Верный ответ на пример задачи из внешнего источника данных]; Задача: [Поставленная задача]"	ТЭП
	27	<i>Step-Back Prompting</i>	{Step-back question - производный вопрос от исходного на более высоком уровне абстракции} "[Задаём модели step-back question по поставленной задаче]; Задача: [Поставленная задача, дополненная ответом на step-back question]" {Пример step-back question: "Как представить содержание текста в виде структурированных семантических троек?"}	53
	28	<i>Tabular Chain-of-Thought (Tab-CoT) Prompting</i>	"Представьте процесс решения поставленной задачи в виде таблицы: " шаг подзадача процедура результат "; После, предоставьте ответ, извлекая результат из заполненной таблицы; Задача: [Поставленная задача]"	32
	29	<i>Thread of Thought (ThoT) Prompting</i>	"[«Хаотичный» контекст]; Задача: [Поставленная задача]; Проведите меня через этот контекст по управляемым частям шаг за шагом, подводя итоги и анализируя по ходу дела.", "[«Хаотичный» контекст]; Задача: [Поставленная задача]; Проведите меня через этот контекст по управляемым частям шаг за шагом, подводя итоги и анализируя по ходу дела; [Ответ модели на предыдущий запрос]; Таким образом, ответ таков: "	НП
	30	<i>Uncertainty-Routed Chain-of-Thought Prompting</i>	"Ведите N независимых записей того, как вы думаете шаг за шагом, напишите их вместе с ответами и оцените свою уверенность от 1 до 10 для каждого шагов рассуждений и окончательных ответов. Сравните полученные ответы и оценки уверенности. Если все цепочки пришли к одному ответу и средняя оценка уверенности превышает порог в 7 баллов, то этот ответ считается надежным, иначе это указывает на высокую неопределенность. На основе мажоритарного голосования и анализа уверенности дайте окончательный ответ. Задача: [Поставленная задача]"	13

Продолжение таблицы

1	2	3	4	5
Ensembling Prompting	31	<i>Consistency-based Self-adaptive Prompting (COSP)</i>	{Банк примеров: Запрос (Q) - Ответ (A)}, {Generating Responses: (Модель занимается построением нескольких цепочек мыслей Zero-Shot CoT к каждому примеру, после все пути оцениваются по надежности и согласованности)}, {Selecting Demonstrations: (Лучшие ответы отбираются в качестве демонстраций на основе таких критериев, как последовательность (повторяется ли один и тот же ответ), минимальное количество повторений и разнообразие путей рассуждения)}; "Решите поставленную задачу несколькими путями, опираясь на отобранные демонстрации, оцените каждый ответ от 1 до 10 на предмет их надежности, после мажоритарного голосования дайте окончательный ответ. Задача: [Поставленная задача]"	ТПР
	32	<i>Demonstration Ensembling (DENSE) Prompting</i>	{Повторить N раз:} "Проанализируйте следующие примеры - [Пример задачи 1], [Пример задачи 2], [Пример задачи 3] и решите следующую задачу: [Поставленная задача]", "Сравните все полученные ответы обоснованных и надежных вариантов в итоговый ответ"	ТЭП
	33	<i>DiVeRSe (Diverse Verifier on Reasoning Step) Prompting</i>	"Ведите N независимых записей того, как вы думаете шаг за шагом, напишите их вместе с ответами. Задача: [Поставленная задача]", {Score Reasoning Paths: (Оценка вероятности правильности для каждого шага рассуждений в процентах при помощи верификатора голосования)}, {Step-Aware Verification: (Применение пошаговой верификации для проверки правильности отдельных шагов рассуждения)}, "Используйте взвешенное голосование, чтобы прийти к окончательному ответу, выбрав наиболее вероятный правильный ответ на основе проверенных путей рассуждения"	ТПР
	34	<i>Max Mutual Information (MMI) Prompting</i>	"Сгенерируйте набор шаблонов промптов для поставленной задачи. Задача: [Поставленная задача]", {Calculate Mutual Information Scores: (Подключите каждый шаблон к алгоритму взаимной информации, чтобы получить оценку для каждого шаблона)}, "Выберите и примените шаблон, получивший наивысшую оценку взаимной информации, на поставленной задаче, и решите её. Задача: [Поставленная задача]"	НП
	35	<i>Mixture of Reasoning Experts (MoRE) Prompting</i>	"Задача: [Поставленная задача]", {Expert Predictions: (Каждая специализированная модель генерирует ответ на основе своего опыта в рассуждениях)}, {Answer Selection: (Селектор ответов выбирает наиболее надежный ответ, основанный на согласии между экспертами и достоверности прогноза, если такого ответа нет, система воздержится)}	ТПР

Продолжение таблицы

1	2	3	4	5
Ensembling Prompting	36	<i>Multi-Chain Reasoning (MCR) (Meta-Reasoning over Multiple CoTs) Prompting</i>	"Задача: [Пример задачи]; Нужны ли здесь дополнительные вопросы: Да; Продолжение: [Подзадача 1] Промежуточный ответ: [Верный ответ на подзадачу 1]; Последующие действия: [Подзадача 2]; Промежуточный ответ: [Верный ответ на подзадачу 2]; Таким образом, окончательный ответ: [Верный ответ на пример задачи]; Задача: [Поставленная задача]; Ведите запись того, как вы думаете шаг за шагом при решении промежуточных подзадач и напишите её вместе с окончательным ответом. Здесь необходимы дополнительные вопросы: ", "На основе ваших цепочек рассуждений, представленных при решении промежуточных подзадач, сгенерируйте окончательный ответ"	ТЭП
	37	<i>Prompt Paraphrasing</i>	"Переведите условие исходного запроса на другой язык, не трогая предоставленные данные, а затем обратно, не решая его. Задача: [Поставленная задача]", "Задача: [Перефразированная переводом поставленная задача]"	НП
	38	<i>Self-Consistency Prompting</i>	{Повторить N раз, вводя после первого запроса фразу "Попробуйте решить задачу ещё раз":} "Задача: [Поставленная задача]", "На основе ваших предыдущих ответов предоставьте окончательный"	45
	39	<i>Universal Self-Adaptive Prompting (USP)</i>	{Банк примеров: Запрос (Q) - Ответ (A)}, {Task-Type Categorization: (Определение задач к одному из трех типов: CLS, SFG или LFG)}, {Pseudo-Demonstrations: (Генерация демонстрационных примеров для этой категории задач. Модель создает несколько ответов для каждого запроса)}, {Scoring and Selection: (Для отбора наилучших псевдодемонстраций используется целевая функция уверенности. Для CLS - энтропия логов, для SFG - самосогласованность ответов, для LFG - метрики схожести)}, "Используйте отобранные псевдодемонстрации в качестве примеров для решения поставленной задачи. Задача: [Поставленная задача]"	ТПР
	40	<i>Universal Self-Consistency Prompting</i>	"Сгенерируйте N вариантов решения следующей задачи через цепочки рассуждений с ответами. Оцените все ответы. Выберите наиболее последовательный ответ на основе консенсуса большинства. Начните свой ответ со слов «Наиболее последовательным ответом является ответ X» (без кавычек). Задача: [Поставленная задача]"	20

Продолжение таблицы

1	2	3	4	5
Self-Criticism Prompting	41	<i>Chain of Density (CoD) Prompting</i>	"Статья: [Информационная статья]; Вы будете создавать все более краткие, насыщенные сущностями резюме вышеупомянутой статьи. Повторите следующие 2 шага 5 раз. Шаг 1. Определите 1-3 информативных сущности («;» с разделителями) из статьи, которые отсутствуют в ранее сгенерированном резюме. Шаг 2. Напишите новое, более плотное резюме той же длины, которое охватывает все сущности и детали из предыдущего резюме плюс отсутствующие сущности. Отсутствующая сущность – это: 1) Актуально: к основному сюжету; 2) Конкретный: описательный, но лаконичный (5 слов или меньше); 3) Роман: нет в предыдущем кратком изложении; 4) Верные: присутствуют в Статье; 5) В любом месте: находится в любом месте статьи. Руководящие принципы: 1) Первое резюме должно быть длинным (4-5 предложений, ~80 слов), но крайне неконкретным, содержащим мало информации, за исключением объектов, помеченных как отсутствующие. Используйте слишком многословный язык и заполнители (например, «в этой статье обсуждается»), чтобы достичь ~80 слов; 2) Сделайте каждое слово важным: перепишите предыдущее резюме, чтобы улучшить поток и освободить место для дополнительных сущностей; 3) Освободите пространство с помощью слияния, сжатия и удаления неинформативных фраз типа «обсуждается в статье»; 4) Резюме должно быть очень плотным и кратким, но в то же время самодостаточным, т.е. легко воспринимаемым без статьи. 5) Отсутствующие сущности могут отображаться в любом месте новой сводки; 6) Никогда не удаляйте сущности из предыдущей сводки. Если пространство освободить не удастся, добавьте меньше новых объектов; Помните, что для каждого резюме используется одно и то же количество слов. Ответ в формате JSON. JSON должен представлять собой список (длина 5) словарей, ключами к которым являются "Missing_Entities" и "Denser_Summary"	НП
	42	<i>Chain-of-Verification (CoVe) Prompting</i>	"Задача: [Поставленная задача]", "[Задаём модели возвращающий к поставленной задаче вопрос]", "Воспользуйтесь своим уточнением, чтобы дать окончательный вариант ответа" {Пример возвращающего вопроса: "Верно ли извлечены все триплеты из заданного текста, все ли триплеты указаны?"}	25

Продолжение таблицы

1	2	3	4	5
Self-Criticism Prompting	43	<i>Cumulative Reasoning Prompting</i>	"Для решения поставленной задачи вы будете последовательно исполнять три роли: Proposer, Verifier и Reporter. Proposer: Иницирует рассуждение, предлагая возможные следующие логические шаги или гипотезы на основе текущего контекста и предыдущих установленных фактов. Verifier: Критически оценивает предложения предлагающего. Проверяет их на логическую обоснованность, соответствие предыдущим шагам и отсутствие противоречий. Отвергает неверные предложения. Reporter: Фиксирует верифицированные шаги, обновляя общую цепочку рассуждений. Решает, когда процесс рассуждения может быть завершен для формирования окончательного ответа, или дает команду предлагающему продолжить, исходя из накопленного контекста. Задача: [Поставленная задача]"	16
	44	<i>Reversing Chain-of-Thought (RCoT) Prompting</i>	"Задача: [Поставленная задача]", "Укажите конкретную задачу, которая может привести к такому ответу. Задача должна содержать всю основную и необходимую информацию и соответствовать ответу. В задаче может быть задан только один результат. Ответ: [Ответ модели на поставленную задачу].", "Перечислите условия исходной задачи и реконструированной задачи. Условий может быть несколько. Исходная задача: [Поставленная задача]; Реконструированная задача: [Реконструированная моделью задача]", "Найдите галлюцинированные и пропущенные условия из списка, если такие имеются: [Список условий исходной и реконструированной задачи]", "Как вы считаете, эти задачи, в конечном счете, задают один и тот же вопрос? Распишите свою причину и ответьте "да" или "нет". Задача: [Поставленная задача]; Реконструированная задача: [Реконструированная моделью задача]", "Соберите все фактические несоответствия из ваших рассуждений и пересмотрите своё решение, если это необходимо. Предоставьте окончательный ответ на поставленную задачу. Задача: [Поставленная задача]"	48
	45	<i>Self-Calibration Prompting</i>	"Задача: [Поставленная задача]", "Предложенный вами ответ является истинным (верным) или ложным (неверным)?", {Ответ ложный (неверный) или частично истинный (верный):} "Проведите повторное решение поставленной задачи"	35
	46	<i>Self-Refine Prompting</i>	"Задача: [Поставленная задача]", "Что вы думаете о данном ответе на поставленную задачу?", "Воспользуйтесь своим уточнением, чтобы дать окончательный вариант ответа"	37

Продолжение таблицы

1	2	3	4	5
Self-Criticism Prompting	47	<i>Self-Verification Prompting</i>	{Повторить N раз:} "Задача: [Пример задачи]; Рассуждение: 1. ..., 2. ..., 3.; Окончательный ответ: [Верный ответ на пример задачи]; Задача: [Поставленная задача]", "Переформулируйте ответ в утверждение, включающее все условия задачи", {Отсекаем ответ от полученного утверждения:} "[Утверждение без ответа]; Назовите ответ, подразумеваемый в данном утверждении.", "Выберите лучший ответ, учитывая логическую согласованность и отсутствие противоречий"	ТЭП
Decomposition Prompting	48	<i>Chain of Code (CoC) Prompting</i>	{Code Generation:} "Задача: [Поставленная задача]; # Код на Python, возвращающий ответ", {Code Execution with an LMulator: LMulator использует рассуждения на основе языка для моделирования того, каким должен быть вывод на основе контекста семантических частей кода и задачи}	ТПР
	49	<i>Chain-of-Logic Prompting</i>	"Правило: [Пример правила]; Факты: [Факты, относящиеся к примеру правила]; Задача: [Пример задачи, которую требуется решить]; Я проанализирую эти данные, используя логическую цепочку: 1. Структура входных данных: Правило: [Переформулированное правило]; Факты: [Факты, относящиеся к переформулированному правилу]; Проблема: [Переформулированная задача]; 2. Элементы правила: А: [Первый ключевой элемент правила]; В: [Второй ключевой элемент]; С: [Третий ключевой элемент, если применимо]; 3. Логическое выражение: [Показать, как элементы сочетаются с И/ИЛИ]; 4. Анализ элементов: {Повторить для каждого элемента:} Для элемента А: Q: [Перефразируйте элемент как вопрос] - А: [Анализ, основанный на фактах + Истина/Ложь]; 5. Логический синтез: [Показать выражение с истинными/ложными значениями]; 6. Окончательное решение: [Решение логического выражения]; Теперь, проанализируйте этот случай: Правило: [Правило по поставленной задаче]; Факты: [Факты, относящиеся к исходному правилу]; Задача: [Поставленная задача]"	НП
	50	<i>Decomposed (DecomP) Prompting</i>	{QC: [Поставленная задача]}, {Каждая подзадача задаётся разным моделям или в разных чатах} "Q1: [Первая подзадача]", "Q2: [Вторая подзадача]", "Q3: [Третья подзадача]", "Задача: [Поставленная задача]; Выведи окончательный ответ на основе уже полученных ответов на подзадачи; A1: [Ответ на первую подзадачу]; A2:[Ответ на вторую подзадачу]; A3:[Ответ на третью подзадачу]"	40

Продолжение таблицы

1	2	3	4	5
Decomposition Prompting	51	<i>Duty-Distinct Chain-of-Thought (DDCoT) Prompting</i>	"Система: Вы полезный, высокоинтеллектуальный управляемый помощник. Вы делаете все возможное, чтобы помочь людям выбрать правильный ответ на вопрос. Обратите внимание, что недостаточная информация для ответа на вопросы является обычным явлением, потому что у вас может не быть информации о картине. Окончательный ответ должен быть одним из вариантов; Пользователь: Учитывая контекст, вопросы и варианты, продумайте шаг за шагом предварительные знания для ответа на вопрос, разберите вопрос как можно полнее, вплоть до необходимых подвопросов на основе контекста, вопросов и вариантов. Затем, чтобы помочь людям ответить на исходный вопрос, попытайтесь ответить на подвопросы. Ожидаемая форма ответа выглядит следующим образом: Подвопросы: [Подвопрос 1]; [Подвопрос 2] ...; Дополнительные ответы: [Подответ 1] или "Неопределенно"; [Подответ 2] или "Неопределенно" ...; Ответ: [Один из вариантов] или "Неопределенно"; Что касается вопроса, если у вас нет никакой информации о картине, то всё равно попытайтесь ответить на подвопросы, расставьте приоритеты, могут ли ваши общие знания ответить на него, а затем подумайте, может ли помочь контекст. Если на подвопросы можно ответить, то отвечайте как можно короче предложением. Если подвопросы не могут быть определены без информации в изображениях, пожалуйста, сформулируйте соответствующий подответ как "Неопределенно". Используйте "Неопределенно" в качестве ответа только в том случае, если оно присутствует во вложенных ответах. Все ответы ожидаются максимально лаконичными. Условие: Контекст: [Текстовое описание или любой дополнительный контекст для задачи]; Изображение: [Логический индикатор (например, "True" или "False"), указывающий, есть ли изображение]; Вопрос: [Поставленная задача]; Опции: [Список вариантов ответа]", {Модель визуальных ответов на вопросы (VQA) обрабатывает подвопросы, помеченные как неопределенные}, {Модель сочетает свои лингвистические рассуждения с результатами визуального распознавания для создания обоснований:} "Система: Вы полезный, очень умный учитель. Вы не только сделаете все возможное, чтобы привести людей к правильному ответу, но и предоставите обоснования в качестве ориентира.	ТПР

Продолжение таблицы

1	2	3	4	5
Decomposition Prompting	51	<i>Duty-Distinct Chain-of-Thought (DDCoT) Prompting</i>	Пользователь: Учитывая контекст, вопросы, варианты, дополнительную информацию, думайте шаг за шагом и отвечайте на вопросы. Обратите внимание, что нужен не только ответ, но и, обоснование получения ответа. Ожидаемая форма ответа выглядит следующим образом: Обоснование: [Обоснование]; Ответы: [Один из вариантов ответа]; Обратите внимание, что предоставленная дополнительная информация не всегда может быть действительной. Пожалуйста, выберите достоверную информацию для обоснования и выберите относительно правильный вариант в качестве ответа. Условие: Контекст: [Текстовое описание или любой дополнительный контекст для задачи]; Изображение: [Логический индикатор (например, "True" или "False"), указывающий, есть ли изображение]; Вопрос: [Поставленная задача]; Опции: [Список вариантов ответа]; Дополнительная информация: [Дополнительная информация]"	ТПР
	52	<i>Faithful Chain-of-Thought (CoT) Prompting</i>	"Задача: [Пример задачи]; Ведя запись того, как вы думаете шаг за шагом, разбейте исходную задачу на более простые подзадачи, на которые легко ответить: Q1: [Первая подзадача], Q2: [Вторая подзадача], Q3: [Третья подзадача]; # Предоставьте код на Python, возвращающий ответ на задачу: [Код на Python, возвращающий ответ на пример задачи]; Задача: [Поставленная задача]; Ведя запись того, как вы думаете шаг за шагом, разбейте исходную задачу на более простые подзадачи, на которые легко ответить; # Предоставьте код на Python, возвращающий ответ на задачу"	ТПР
	53	<i>Least-to-Most Prompting</i>	"Задача: [Пример задачи]; Ведя запись того, как вы думаете шаг за шагом, разбейте исходную задачу на более простые подзадачи, на которые легко ответить: Q1-A1: [Первая подзадача][Верный ответ на первую подзадачу], Q2-A2: [Вторая подзадача][Верный ответ на вторую подзадачу], Q3-A3: [Третья подзадача][Верный ответ на третью подзадачу]; Выведи окончательный ответ на основе уже полученных ответов на подзадачи: [Верный ответ на пример задачи]; Задача: [Поставленная задача]; Ведя запись того, как вы думаете шаг за шагом, разбейте исходную задачу на более простые подзадачи, на которые легко ответить; Выведи окончательный ответ на основе полученных ответов на подзадачи"	ТЭП

Продолжение таблицы

1	2	3	4	5
Decomposition Prompting	54	<i>Metacognitive Prompting</i>	"Задача: [Поставленная задача]; Ведя запись того, как вы думаете шаг за шагом при решении поставленной задачи, выполните все следующие действия: 1. Уточните своё понимание исходного вопроса; 2. Выполните предварительное суждение своего рассуждения; 3. Критически оцените свой предварительный анализ. Если вы не уверены, попробуйте пересмотреть решение; 4. Подтвердите свое окончательное решение. 5. Оцените свою уверенность (от 0 до 100%) в своем анализе и объясните этот уровень уверенности"	20
	55	<i>Plan-and-Solve (+PS Plus) Prompting</i>	"Задача: [Поставленная задача]; Сначала разберитесь в проблеме и разработайте план ее решения. Затем, ведя запись того, как вы думаете шаг за шагом, выполните разработанный план", {Запись выполнения плана шаг за шагом передаётся в другую модель или чат:} "Задача: [Поставленная задача]; Таким образом, окончательным ответом на поставленную задачу будет: "	41
	56	<i>Program of Thoughts Prompting</i>	"Задача: [Поставленная задача]; # Предоставьте код на Python, возвращающий ответ на задачу"	ТПР
	57	<i>Recursion of Thought Prompting</i>	{Обученная модель по рекурсивному принципу «разделяй и властвуй»: (Каждый раз, когда модель сталкивается со сложной проблемой в середине цепочки рассуждений, он отправляет эту проблему в другой запрос, где после завершения этого процесса ответ отправляется в исходный)} "Задача: [Пример задачи]; Рассуждение: 1. ..., 2. ..., 3. ...; Окончательный ответ: [Верный ответ на пример задачи]; Задача: [Поставленная задача]"	ТПР
	58	<i>Skeleton-of-Thought Prompting</i>	{Skeleton Stage:} "Вы являетесь организатором, ответственным за предоставление только скелета (а не полного содержания) для ответа на задачу. Предоставьте скелет в списке пунктов (пронумерованных 1., 2., 3., и т.д.) для ответа на задачу. Вместо того, чтобы писать полное предложение, каждый скелет должен быть очень коротким, всего 3-5 слов. Как правило, скелет должен иметь 3-10 токенов. Предоставьте скелет для следующей задачи. Задача: [Поставленная задача]", {Point-Expanding Stage:} "Вы несете ответственность за расширение пунктов скелета, а также за написание ответа на поставленную задачу на его основе. Расширяя пункты скелета, расписывайте их очень кратко по 1-2 предложению на каждый. Сначала верните расширенный скелет, затем ответ. Задача: [Поставленная задача]; Скелет: [Полученный от модели скелет]"	19

Окончание таблицы

1	2	3	4	5
Decomposition Prompting	59	<i>Tree of Thoughts (ToT) Prompting</i>	"Задача: [Поставленная задача]; Создайте несколько первоначальных подходов к решению исходной задачи. Организуйте мысли в древовидную структуру, где узлы - промежуточные шаги решения, а ветви - альтернативные пути развития мыслей. Оцените перспективность каждого пути. При необходимости возвращайтесь к предыдущим узлам для исследования альтернативных путей решения задачи"	23

Типы промптинга, требующие программной реализации, также не могли быть реализованы в рамках стандартного интерфейса чата языковой модели DeepSeek, поскольку они предполагают интеграцию с внешними вычислительными модулями, которые либо необходимо искать в открытых источниках и подключать отдельно, либо разрабатывать самостоятельно.

Необходимо отметить, что типы промптинга, требующие предоставления экспертных примеров, в рамках данного исследования не тестировались эмпирически из-за методологических ограничений, поскольку требовали наличия заранее подготовленного набора размеченных экспертом примеров, создание которых представляет отдельную сложную задачу. Такая задача требует привлечения квалифицированных предметных специалистов.

Далее представлены полученные в ходе практического исследования результаты, включая количественные и качественные оценки, анализ применимости различных типов промптов, а также выявленные закономерности.

Необходимо отметить, что для сбора информации о модификаторах промптов каждого типа использовались данные из различных источников. Одна из сложностей при проведении исследования состояла в том, что собранные данные часто были неточными, неполными и даже противоречивыми. В частности, трактовка одних и тех же типов промптинга могла сильно варьироваться от источника к источнику, одинаковые типы относились к различным стратегиям и, наоборот, один и тот же тип мог быть включен в разные стратегии промптинга в разных источниках. В связи с этим исследование и систематизация имеющихся подходов промптинга для понимания их эффективности при извлечении RDF-троек из текстов на естественном языке представляют собой нетривиальную задачу.

Классификация и формирование модификаторов типов промптинга имеют достаточно гибкую интерпретацию, поскольку каждый тип промптинга может работать в связке друг с другом, использовать параметры друг друга, включать один или несколько экспертных примеров или вообще не иметь их. Поэтому предложенная в работе классификация типов промптинга и список их модификаторов представляют собой один из возможных вариантов систематизации и не рассматривается как полностью исчерпывающее или единственно верное решение.

Количественная оценка результатов показала значительный разброс в эффективности различных типов промптинга. Среднее количество извлеченных RDF-троек для всех исследованных типов составило 34. Наибольшее количество RDF-троек (53) было извлечено с использованием Step-Back Prompting (тип 27) благодаря способности модели выявлять абстрактные принципы перед детальным анализом. Среди наиболее эффективных типов промптинга также можно выделить Zero-Shot Chain-of-Thought (тип 19) с 42 RDF-тройками и Reversing Chain-of-Thought (тип 44) с 48 RDF-тройками, демонстрирующие преимущество типов, сочетающих поэтапное рассуждение с механизмом самокоррекции. В то же время наихудшие результаты с наименьшим количеством найденных RDF-троек показали Uncertainty-Routed Chain-of-Thought Prompting (тип 30) с 13 RDF-тройками и Chain of Draft (тип 17) с 17 RDF-тройками, а также Self-Calibration Prompting (тип 45) с 20 RDF-тройками. Такие показатели подтверждают, что высокая вычислительная сложность не является гарантией значительных результатов.

Качественная оценка результатов выявила важную закономерность: даже при высоких количественных показателях некоторые типы промптинга демонстрировали склонность к генерации искаженных данных. В частности, типы, основанные на многошаговых рассуждениях, то есть те, что подразумевают длительную обработку информации, хотя и показывали в среднем высокие количественные результаты, часто порождали семантические галлюцинации — добавление информации, отсутствующей в исходном тексте, или искажение уже имеющейся. Напротив, подходы с элементами самокритики, такие как Chain-of-Verification (тип 42) и Reversing Chain-of-Thought (тип 44) демонстрировали значительно более высокую точность при сохранении хороших количественных показателей.

На основе приведенных в ходе исследования оценок эффективности типов промптинга были сформулированы практические рекомендации по их использованию. Для достижения максимальной полноты извлечения данных следует применять Step-Back Prompting (тип 27), Zero-Shot Chain-of-Thought (тип 19). Для баланса между полнотой и точностью оптимальны Rephrase and Respond (тип 3), Generated Knowledge Prompting (тип 23) и Universal Self-Consistency Prompting (тип 40). Следует избегать по возможности типов промптинга с высокими вычислительными затратами и низкой отдачей, таких как Uncertainty-Routed Chain-of-Thought (тип 30), а также типов, требующих программной реализации, при отсутствии значительных финансовых средств.

Заключение

Проведенное исследование подтвердило основную гипотезу, состоящую в том, что выбор стратегий и типов промптинга оказывает значительное влияние на эффективность извлечения структурированных данных из нехудожественных разнородных текстов. На основе полученных в процессе исследования данных была предложена классификация типов промптинга, выделены 6 базовых стратегий для 59 типов промптинга. Экспериментальная оценка 59 типов промптинга позволила не только выявить закономерности в их эффективности, но и сформировать практические рекомендации по выбору оптимальных решений для задач преобразования неструктурированного текста в целостный набор RDF-троек. Особую значимость приобретают полученные результаты в контексте разработки технологии синтеза адаптивных геосемантических изображений на основе геопространственных знаний, которая является предметом наших дальнейших исследований.

Практическая значимость работы состоит в том, что установленные показатели эффективности в дальнейшем послужат ориентиром в планировании оптимального подхода для автоматического построения графов знаний при помощи больших языковых моделей и для создания эффективных конвейеров обработки разнородных текстов, что позволит существенно повысить релевантность и информативность визуализаций текстовых описаний с географическими свойствами. Выявленные наиболее эффективные типы промптинга, например, такие как Step-Back Prompting и Zero-Shot Chain-of-Thought, представляют ценность для обеспечения полноты извлечения пространственных отношений, что важно для формирования многоуровневых геосемантических моделей. Повышенный интерес представляет адаптация выявленных закономерностей для обработки не только текстов о чрезвычайных ситуациях, но и разнородных географических описаний, что позволит расширить и без того комплексные семантические модели пространственно-временных ситуаций.

Список источников

1. Vicentiy A. V., Dikovitsky V. V., Shishaev M. G. The semantic models of arctic zone legal acts visualization for express content analysis // Computer Science On-line Conference. Springer, Cham. 2018. Vol. 763. P. 216–228.
2. Vicentiy A. Definition and formalization of the user mental model for creating adaptive geointerfaces of decision support systems // Lecture Notes in Networks and Systems. Springer, Cham. 2024. Vol. 733. P. 1095–1105.
3. Gruber T. R. A translation approach to portable ontology specifications // Knowledge Acquisition. Academic Press. 1993. Vol. 5, № 2. P. 199–220.
4. RDF 1.2 Concepts and Abstract Data Model [Электронный ресурс]. URL: <https://www.w3.org/TR/rdf12-concepts> (дата обращения: 31.10.2025).
5. Gumiel Y. B. et al. Temporal relation extraction in clinical texts: a systematic review // Association for Computing Machinery Computing Surveys. 2022. Vol. 54, № 7. P. 1–36.

6. Sun T. Relation extraction from financial reports: Doctoral dissertation. University of York, 2022.
7. Zhu G. et al. Relationship extraction method for urban rail transit operation emergencies records // *Institute of Electrical and Electronics Engineers Transactions on Intelligent Vehicles*. 2023. Vol. 8, № 1. P. 520–530.
8. Sharma T., Emmert-Streib F. Deep mining the textual gold in relation extraction // *Artificial Intelligence Review*. 2024. Vol. 58, № 1. P. 983–1021.
9. Wang H. et al. Deep neural network-based relation extraction: an overview // *Neural Computing and Applications*. 2022. Vol. 34, № 6. P. 4781–4801.
10. ChatGPT [Электронный ресурс]. URL: <https://chatgpt.com> (дата обращения: 01.11.2025).
11. Claude [Электронный ресурс]. URL: <https://claude.com> (дата обращения: 01.11.2025).
12. Gemini [Электронный ресурс]. URL: <https://gemini.google.com> (дата обращения: 01.11.2025).
13. GigaChat [Электронный ресурс]. URL: <https://giga.chat> (дата обращения: 01.11.2025).
14. YandexGPT [Электронный ресурс]. URL: <https://ya.ru/ai/gpt> (дата обращения: 01.11.2025).
15. Schulhoff S. et al. The prompt report: a systematic survey of prompt engineering techniques // *arXiv preprint arXiv:2406.06608*. 2024.
16. Liu P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing // *Association for Computing Machinery Computing Surveys*. 2023. Vol. 55, № 9. P. 1–35.
17. Sahoo P. et al. A systematic survey of prompt engineering in large language models: Techniques and applications // *arXiv preprint arXiv:2402.07927*. 2024.
18. Peng R. et al. Embedding-based retrieval with llm for effective agriculture information extracting from unstructured data // *arXiv preprint arXiv:2308.03107*. 2023.
19. Bisercic A. et al. Interpretable medical diagnostics with structured data extraction by large language models // *arXiv preprint arXiv:2306.05052*. 2023.
20. Schilling-Wilhelmi M. et al. From text to insight: large language models for chemical data extraction // *Chemical Society Reviews*. 2025. Vol. 54. P. 1125–1150.
21. Столкновение поездов на станции Княжая // Википедия : сайт [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Столкновение_поездов_на_станции_Княжая (дата обращения: 31.10.2025).
22. DeepSeek [Электронный ресурс]. URL: <https://www.deepseek.com> (дата обращения: 31.10.2025).
23. LiveBench [Электронный ресурс]. URL: <https://livebench.ai> (дата обращения: 01.11.2025).
24. Vellum Leaderboard [Электронный ресурс]. URL: <https://www.vellum.ai/llm-leaderboard> (дата обращения: 01.11.2025).
25. The Big Benchmarks Collection Leaderboard [Электронный ресурс]. URL: <https://huggingface.co/collections/open-llm-leaderboard/the-big-benchmarks-collection> (дата обращения: 01.11.2025).

References

1. Vicentiy A. V., Dikovitsky V. V., Shishaev M. G. The semantic models of arctic zone legal acts visualization for express content analysis. *Computer Science On-line Conference*. Springer, Cham, 2018, vol. 763, pp. 216–228.
2. Vicentiy A. Definition and formalization of the user mental model for creating adaptive geointerfaces of decision support systems. *Lecture Notes in Networks and Systems*. Springer, Cham, 2024, vol. 733, pp. 1095–1105.
3. Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*. Academic Press, 1993, vol. 5, no. 2, pp. 199–220.
4. RDF 1.2 Concepts and Abstract Data Model. Available at: <https://www.w3.org/TR/rdf12-concepts> (accessed 31.10.2025).
5. Gumiel Y. B. et al. Temporal relation extraction in clinical texts: a systematic review. *Association for Computing Machinery Computing Surveys*, 2022, vol. 54, no. 7, pp. 1–36.
6. Sun T. *Relation extraction from financial reports*. Doctoral dissertation. University of York, 2022.
7. Zhu G. et al. Relationship extraction method for urban rail transit operation emergencies records. *Institute of Electrical and Electronics Engineers Transactions on Intelligent Vehicles*, 2023, vol. 8, no. 1, pp. 520–530.
8. Sharma T., Emmert-Streib F. Deep mining the textual gold in relation extraction. *Artificial Intelligence Review*, 2024, vol. 58, no. 1, pp. 983–1021.
9. Wang H. et al. Deep neural network-based relation extraction: an overview. *Neural Computing and Applications*, 2022, vol. 34, no 6, pp. 4781–4801.
10. ChatGPT. Available at: <https://chatgpt.com> (accessed 01.11.2025).
11. Claude. Available at: <https://claude.com> (accessed 01.11.2025).
12. Gemini. Available at: <https://gemini.google.com> (accessed 01.11.2025).
13. GigaChat. Available at: <https://giga.chat> (accessed 01.11.2025).
14. YandexGPT. Available at: <https://ya.ru/ai/gpt> (accessed 01.11.2025).

15. Schulhoff S. et al. The prompt report: a systematic survey of prompt engineering techniques. arXiv preprint arXiv:2406.06608, 2024.
16. Liu P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Association for Computing Machinery Computing Surveys*, 2023, vol. 55, no 9, pp. 1-35.
17. Sahoo P. et al. A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927, 2024.
18. Peng R. et al. Embedding-based retrieval with llm for effective agriculture information extracting from unstructured data. arXiv preprint arXiv:2308.03107, 2023.
19. Bisercic A. et al. Interpretable medical diagnostics with structured data extraction by large language models. arXiv preprint arXiv:2306.05052, 2023.
20. Schilling-Wilhelmi M. et al. From text to insight: large language models for chemical data extraction. *Chemical Society Reviews*, 2025, vol. 54, pp. 1125–1150.
21. Столкновение поездов на стании Князья [Train Collision at Knyazhaya Station]. (In Russ.). Available at: https://ru.wikipedia.org/wiki/Столкновение_поездов_на_станции_Княжая (accessed 31.10.2025).
22. DeepSeek. Available at: <https://www.deepseek.com> (accessed 31.10.2025).
23. LiveBench. Available at: <https://livebench.ai> (accessed 01.11.2025).
24. Vellum Leaderboard. Available at: <https://www.vellum.ai/llm-leaderboard> (accessed 01.11.2025).
25. The Big Benchmarks Collection Leaderboard. Available at: <https://huggingface.co/collections/open-llm-leaderboard/the-big-benchmarks-collection> (accessed 01.11.2025).

Информация об авторах

Р. А. Горбунов — стажер-исследователь;

А. В. Вицентий — кандидат технических наук, старший научный сотрудник.

Information about the authors

R. A. Gorbunov — Research Assistant;

A. V. Vicentiy — Candidate of Science (Tech.), Senior Research Fellow.

Статья поступила в редакцию 21.11.2025; одобрена после рецензирования 24.11.2025; принята к публикации 25.11.2025.
The article was submitted 21.11.2025; approved after reviewing 24.11.2025; accepted for publication 25.11.2025.