УДК 001.92 *Научная статья* DOI: 10.18287/2223-9537-2025-15-1-114-129

# Извлечение информации из текстов на основе онтологии и больших языковых моделей

**© 2025, Е.А. Сидорова[1,2]✉, А.И. Иванов[2], К.А. Овчинникова[2]**

[1] *Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, Россия*

[2] *Новосибирский государственный университет (НГУ), Новосибирск, Россия*

**Аннотация**

Рассматривается извлечение информации из текстов на основе онтологии предметной области и нейросетевых методов анализа текста с привлечением больших языковых моделей. Обсуждается роль эксперта при разработке и сопровождении систем на примере задачи извлечения информации из аналитических статей и при построении онтологий по компьютерной лингвистике, описывающих основные понятия, интересующие пользователя/заказчика системы. Создание онтологии сопровождается созданием словаря - терминологического ядра онтологии с дальнейшей разработкой методов извлечения новых терминов данной предметной области. Данная задача рассматривается как задача извлечения именованных сущностей, для решения которой стандартом является обучение нейросетевой модели на представительном наборе данных. Этот подход сравнивается с подходом на основе больших языковых моделей, для реализации которого разработаны лексико-синтаксические шаблоны, шаблоны инструкций для проверки гипотез относительно новых терминов-словосочетаний, инструкции для верификации результатов. Разработанные инструкции для решения задачи извлечения отношений включают вопросы оценки компетенций на естественном языке, генерируемые автоматически для каждого отношения онтологии. Новизна предлагаемого подхода заключается в интеграции онтологических, лингвистических и нейросетевых подходов для извлечения информации из текстов. Показана возможность решать задачи анализа текста и извлечения информации путём выстраивания цепочки больших языковых моделей, инструкции для которых динамически формируются на основе результатов предыдущих этапов анализа. В эксперименте достигнуты следующие оценки F1-меры: для извлечения и классификации терминов F1=0.8, для извлечения отношений F1=0.87.

***Ключевые слова:*** *извлечение информации, онтология предметной области, большие языковые модели, нейросетевые модели, разработка инструкций.*

***Цитирование:*** *Сидорова Е.А., Иванов А.И., Овчинникова К.А.* Извлечение информации из текстов на основе онтологии и больших языковых моделей. *Онтология проектирования*. 2025. Т.15, №1(55). С.114-129. DOI:10.18287/2223-9537-2025-15-1-114-129.

***Конфликт интересов:*** авторы заявляют об отсутствии конфликта интересов.

## Введение

При создании систем, основанных на знаниях, появляются новые направления исследований, связанные с организацией работы со знаниями (например, экосистемы знаний [1]), новые задачи, такие как генерация запросов на основе базы знаний (БЗ), и др. Несмотря на активное внедрение интеллектуальных методов, разработка информационных систем сопряжена с рядом трудностей. В первую очередь, это решение задач, связанных с моделированием предметной области (ПрО), пополнением БЗ, своевременным обнаружением ошибок. При этом роль эксперта при разработке и дальнейшем сопровождении систем недооценивается в связи с устоявшимся мнением о возросших возможностях глубоких нейронных сетей после появления *больших языковых моделей* (БЯМ). Задачи, связанные с глубокой проработкой ПрО, анализом ошибок и разработкой способов их исправления, подготовкой качественных наборов данных не могут пока решаться полностью автоматически.

Для систематизации, хранения и поиска информации в специализированных системах часто используются графы знаний (ГЗ), онтологии, БЗ, создание которых - трудоёмкий процесс. Активно исследуется возможность автоматического создания и пополнение БЗ на основе автоматической обработки текстов (АОТ). Разработка систем АОТ в разное время опиралась на различные подходы: лингвистический, инженерный, онтологический, классическое и нейросетевое машинное обучение. Соответственно изменялись задачи эксперта при сопровождении систем АОТ – от моделирования знаний о языке и ПрО, описания процессов понимания текстов человеком/машиной до подготовки наборов данных и формулирования задач и разработки инструкций на естественном языке (ЕЯ) для генеративных моделей.

Целью работы является разработка методики создания и сопровождения информационных систем на основе онтологий, АОТ и методов извлечения информации из русскоязычных источников, интегрирующей онтологические, лингвистические и нейросетевые подходы.

## 1 Методы автоматизации построения онтологий

*Онтология* – это способ формализации знаний в какой-либо ПрО, реализованный на основе формального описания объектов, фактов и отношений между ними, и ориентированный на многократное использование для различных задач [2]. Под ГЗ понимается семантическая сеть, в которой хранится информация об объектах и взаимосвязях между ними [3]. В данной работе онтология является семантической основой представления данных ПрО, базирующаяся на логике и включающая терминологический словарь и набор утверждений о моделируемых сущностях, а ГЗ содержит экземпляры понятий и отношений заданной онтологии.

Первые методологии для создания онтологий были направлены на разработку онтологии конкретной ПрО для решения определённых задач, таких как извлечение знаний, систематизация, формализация и поиск информации [4-6]. Активно развивается подход к разработке онтологий, основанный на использовании шаблонов (паттернов) онтологического проектирования[1]. Каждый шаблон представляет собой стандартизированное описание ранее созданного фрагмента онтологии, включающее в т.ч. определение и набор вопросов оценки компетентности (ВОК) на ЕЯ. Суть этих исследований заключается в разработке способов повторного использования шаблонов при создании новых онтологий.

Для создания онтологии компьютерной лингвистики (КЛ) можно обратиться к схожим по тематике материалам. Существует несколько ресурсов, посвящённых КЛ либо связанным с ней областям знаний, например, русско-английский тезаурус по КЛ[2] и портал по КЛ[3]. Известны также несколько онтологий машинного обучения, например, [7]. Для построения терминологического ядра онтологии используются методы извлечения терминов и иерархических связей между ними для создания семантических предметных словарей (ПрС) [8] или тезаурусов.

Выявление терминов в тексте относится к классу задач *распознавания именованных сущностей* [9]. В общем случае, для решения данной задачи требуется найти участки текста, содержащие термины заданной ПрО, а также определить сущности, обозначаемые этими терминами, т.е. определить тип терминов. Одни и те же сущности могут иметь разный тип в зависимости от контекста. Например, термин «*классификация*» может в одном тексте означать *Метод_исследования*, а в другом – *Задачу_исследования*, которую необходимо решить.

Для решения задачи распознавания именованных сущностей могут применяться методы с использованием синтаксических правил [10], методы машинного обучения, например, ме-

[1] Портал Ассоциации *ODPA* (*Association for Ontology Design & Patterns*). http://ontologydesignpatterns.org.
[2] Русско-английский тезаурус по компьютерной лингвистике. https://uniserv.iis.nsk.su/thes.
[3] Портал по компьютерной лингвистике. https://uniserv.iis.nsk.su/cl.

тод опорных векторов или метод случайного леса. Эти методы опираются на характеристики слов, такие как принадлежность к синтаксической группе, к семантическому значению, чтобы определить правила классификации или задать расстояние между словами и разделить их на классы [11]. В большинстве случаев используются методы *глубокого обучения* или гибридные методы на их основе [12-14].

Существуют различные подходы на основе БЯМ, которые являются трансформерами, обученными на больших объёмах текстовых данных. Основой для применения БЯМ является *разработка инструкций* [15]. Суть данного подхода заключается в составлении человеком инструкции, описывающей конкретную задачу на ЕЯ. Данная инструкция подаётся на вход модели, в ответ модель генерирует текст, из которого извлекается решение задачи. Преимуществом подхода является отсутствие необходимости в обучении модели. К недостаткам подхода можно отнести сложность создания оптимальной инструкции, поскольку от её содержания и размера зависит качество работы модели (большая инструкция может привести к более низким результатам).

Подходы на основе инструкций включают разнообразные техники для создания инструкций. Например, *инструкции «с нулевым примером»* [16] не предполагают предоставление модели примеров. Данная техника может не подходить для сложных задач, например, когда требуется получить термины определённой области знаний и/или определить их тип. В таких случаях применяется техника с использованием «*нескольких примеров*» [17], позволяющая обеспечить контекстное обучение с предоставлением демонстраций решения в инструкции. Более сложной техникой является *цепочка рассуждений* [16], суть которой заключается в описании последовательности шагов рассуждения, которую необходимо выполнить для решения задачи (см. рисунок 1 слева).

Одной из стратегий для решения сложной задачи является её декомпозиция на более простые и разработка для каждой из подзадач собственного решения. Применительно к БЯМ такой подход называется *цепочка инструкций*, когда для каждого этапа создаётся отдельная инструкция и результаты предыдущих этапов могут использоваться для генерации инструкции следующего этапа. Цепочка инструкций подразумевает использование выходных данных одной модели в качестве входных данных для следующей модели и, по сути, включает несколько подсказок (см. рисунок 1 справа). Метод цепочки рассуждений использует одну, более длинную инструкцию, которая описывает пошаговый процесс рассуждений для получения ответа.

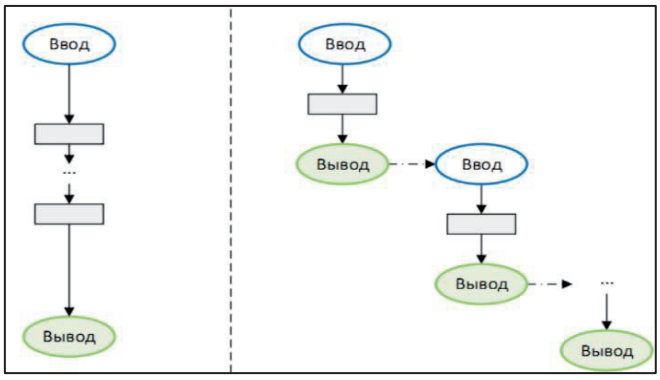



Рисунок 1 – Сравнение схем подходов на основе БЯМ

В первом случае предлагается более простое и быстрое решение с использованием всего одной БЯМ; кроме того, в сравнении со вторым подходом отсутствует накопление ошибки, характерной для применения цепочки решателей. Преимуществом второго подхода является возможность своевременно корректировать результаты при переходе от одного этапа к другому. В работе [18] показано, что для задачи реферирования текста подход на основе цепочки инструкций показывает лучшие результаты по сравнению с одной инструкцией, включающей цепочку задач, и отмечается, что данные результаты можно масштабировать на другие задачи анализа текста. Таким образом, применение методологии построения онтологий научных ПрО [19] должно обогащаться современными методами интеллектуального анализа текста на основе БЯМ.

## 2 Методика исследования

Предлагаемое исследование направлено на решение задачи разработки онтологии КЛ посредством АОТ и извлечения значимой информации. Методика исследования включает декомпозицию задачи извлечения информации на ряд подзадач, проведение экспериментов на текстовом корпусе, сравнительный анализ различных подходов, анализ ошибок, выдвижение гипотез для улучшения результатов и их экспериментальную оценку. Ключевым вопросом проводимых работ является экспертный анализ и оценка его роли в достижении поставленных целей.

С внедрением технологий, позволяющих извлекать знания из неструктурированных источников данных, задачи эксперта изменяются и включают:

- проектирование верхнеуровневых абстракций;
- подготовку данных для применения методов машинного обучения;
- разработку инструкций для генеративных моделей на основе БЯМ;
- валидацию и анализ полученных результатов.

Решение первой задачи необходимо для формализации структуры важной с точки зрения пользователя информации и всех необходимых данных. Для решения этой задачи используется методика разработки онтологии ПрО на основе базовой онтологии, концепты которой специализируются на более точные понятия и отношения ПрО.

Подготовка данных в задачах АОТ включает создание репрезентативного корпуса текстов ПрО, а также анализ существующих ресурсов и оценку возможности их применения для решения задач. Корпус текстов служит основой для верификации результатов на всех этапах решения задачи.

Разработка инструкций опирается на структуру онтологии, знания эксперта о сущностях ПрО и способах их описания на ЕЯ, а также на понимание основных этапов решения задачи.

Валидация и содержательный анализ получаемых результатов на каждом этапе позволяют выдвинуть гипотезы о причинах ошибок и настроить решатели путём подбора параметров моделей, корректировки инструкций, обработки специальных случаев и т.п.

Можно выделить два этапа исследования. Подготовительный этап включает задачи, которые плохо автоматизируются и решаются экспертом. Экспериментальный этап включает разработку алгоритмов решения задачи и анализ результатов.

### 2.1 Подготовительный этап исследования

Разработка интеллектуальной системы начинается с подготовки данных и формализации знаний о ПрО (рисунок 2а). Создаётся онтология, которая служит основой для формирования терминологического ядра онтологии – ПрС. Для её построения можно использовать готовые базовые онтологии или шаблоны онтологического проектирования для описания ключевых классов онтологий [20].

Конкретизация онтологии на определённую ПрО происходит посредством анализа корпуса текстов, который должен содержать примеры использования всех сущностей ПрО. В онтологию добавляются новые классы и редактируются уже имеющиеся.

На основе анализа грамматической структуры многословных терминов могут разрабатываться лексико-синтаксические шаблоны (ЛСШ), которые используются для генерации гипотез о словосочетаниях кандидатов в термины. ЛСШ – это структурный образец языковой конструкции, который отображает её лексические и поверхностные синтаксические свойства.

Использование подхода на основе нейросетевых моделей предполагает подготовку размеченных данных для проведения экспериментов, валидацию данных и нахождение ошибок.

Для облегчения ручной разметки тексты могут быть первично автоматически размечены, а для увеличения набора данных при необходимости могут использоваться методы перевода или перефразирования с помощью специализированных генеративных моделей [21].

Последняя задача подготовительного этапа связана с разработкой шаблонов инструкций для БЯМ, которые используются на всех экспериментальных этапах.

## 2.2 Алгоритмы экспериментального этапа исследования

Процесс извлечения информации включает два этапа: извлечение терминов и извлечение отношений (рисунок 2б).

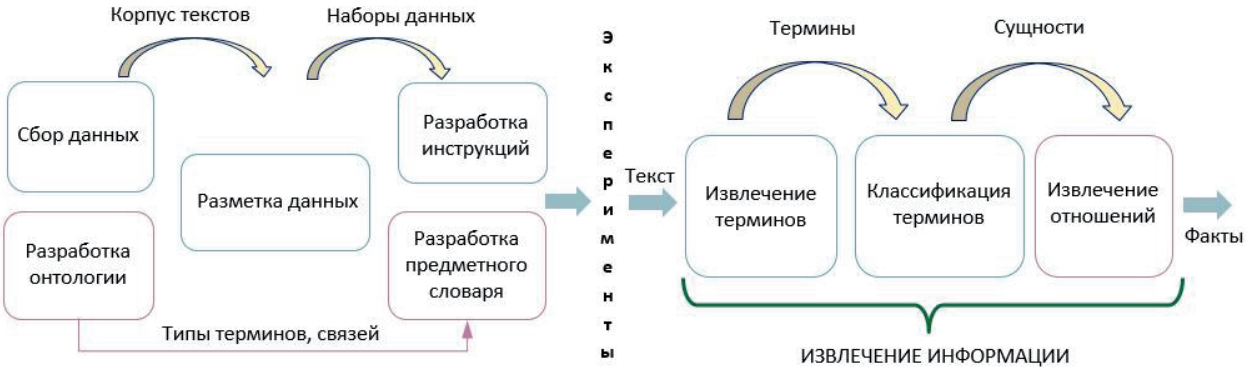



Рисунок 2 – Задачи подготовительного (а) и экспериментального (б) этапов исследования

Для извлечения терминов проводится сравнительное исследование различных подходов. Например, один подход может реализовывать классическую парадигму обучения нейросетевых моделей на наборе размеченных данных (датасетах). Другой подход может реализовывать различные техники на основе БЯМ, где наборы данных не обязательны. В первом подходе выбранная модель обучается отдельно для поиска терминов и их классификации на подготовленных наборах данных. Для поиска терминов используется набор данных с *BIO*[4]-разметкой, а для классификации – набор данных с *BIO*-разметкой, расширенной именами классов терминов [8].

В процессе поиска терминов токенизатор преобразует слово в несколько токенов, затем модель предсказывает одну из трёх меток для каждого токена: начало термина, середина термина или не термин. Итоговая метка для слова определяется как самая частая метка токенов, которые к этому слову относятся. Классификация извлечённых терминов происходит в контексте всего предложения, в котором термин выделяется специальными тегами <term> … </term>. Данный подход позволяет модели передавать на вход контекст, выделяя термин, класс для которого необходимо предсказать.

Подход к извлечению терминов, предлагаемый в данной работе, заключается в комплексном использовании ЛСШ, ПрС и БЯМ (рисунок 3).

На первом этапе проводится поиск и извлечение терминоподобных слов и словосочетаний с использованием ЛСШ. Извлечённые кандидаты в термины анализируются с точки зрения их принадлежности к ПрО. Для этого осуществляется поиск семантически близких терминов в словаре, что позволяет предварительно определить наиболее подходящие классы для термина. Для каждого термина-кандидата генерируется инструкция, в которой найден-

[4] *BIO*-разметка используется для выделения в тексте границ именованных сущностей и их типа. B-метка (*beginning*) проставляется для обозначения начала интересующей сущности; I (*inside*) — для обозначения слова внутри неё; O (*outside*) — это любое слово за её пределам.

ные варианты терминов и их классы используются в качестве примеров, и с помощью БЯМ происходит уточнение класса или вывод о его отсутствии, что позволяет исключить данный термин из рассмотрения. После этого ответ модели подвергается верификации у той же или другой БЯМ.

По сравнению с подходом [8] для комплексного подхода нет необходимости создания больших наборов данных для обучения моделей, что достигается путём привлечения экспертных знаний: лингвистических моделей для генерации гипотез о наличии терминов и ПрС для генерации гипотез о классе терминов.

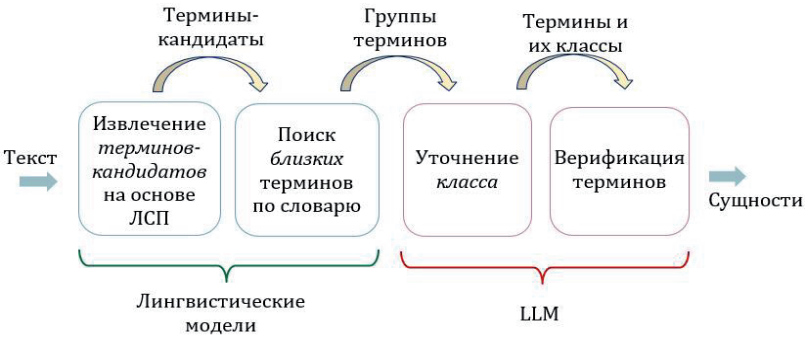



Рисунок 3 – Комплексный подход к извлечению терминов на основе больших языковых моделей (LLM)

*Предложен следующий алгоритм для извлечения и классификации терминов.*

Пусть $T$ – множество всех терминов выбранной ПрО. Для произвольной входной строки $s$ требуется найти множество терминов $T_s \subset T$, которые в ней содержатся, причём для каждого $t \in T_s$ должен быть определён класс онтологии $c \in C$, к которому он относится, где $C$ – множество всех классов терминов в ПрО. Таким образом, требуется построить отображение $F: S \rightarrow T \times C$.

**Алгоритм 1.** Общий алгоритм извлечения и классификации терминов ПрО.
**Input:** Строка s, БЯМ $L$, словарь терминов $D$ и набор ЛСШ $P$
**Output:** Множество пар $(t, c)$, где $t \in s$, $t \in T$, $c \in C$

```
1: phrases ← P.find_entries ( s )
2: for all phrase ∈ phrases do
3:    similar_terms ← D.find_similar_terms ( phrase )
4:    cls_prompt ← create_cls_prompt ( s, phrase, similar_terms )
5:    class ← L.invoke ( cls_prompt )
6:    if class = null do
7:       continue
8:    end if
9:    ver_prompt ← create_ver_prompt ( s, phrase, class )
10:   ver_class ← L.invoke ( verification_prompt )
11:   if ver_class = null do
12:      continue
13:   end if
14:   yield ( phrase, ver_class )
15: end for
```

Извлечение кандидатов в термины из s с помощью набора шаблонов $P$ осуществляется функцией *find_entries* (строка 1). Поиск кандидата *phrase* в словаре терминов $D$ осуществляется функцией *find_similar_terms* (строка 3). Генерация инструкций для классификации и верификации терминов происходит в строках 4 и 9 соответственно. Вызов процедуры *invoke* выполнения запросов к модели $L$ - в строках 5 и 10.

На следующем этапе поиска отношений необходимо извлекать предметные связи, заданные онтологией. Это означает, что отношение между двумя терминами возможно только в том случае, если оно возможно между их классами в онтологии. Поэтому рассматриваются только те пары терминов, которые соответствуют этому условию.

Для решения данной задачи предлагает применять подход на основе БЯМ. Для каждой пары терминов формируется специальная инструкция, в которой размещается фрагмент текста (предложение или абзац), включающий эти термины, описание классов терминов, описание проверяемого типа от-

ношения и несколько примеров желаемых ответов модели для заготовленных на этапе подготовки данных примеров. При составлении инструкций применяется техника, в которой вначале модель должна ответить, есть отношение или нет, и в случае, если оно есть, указать метку класса, к которому оно относится.

*Алгоритм для извлечения отношений следующий.*

Пусть $T_s \subset T$ – множество найденных терминов, полученных на предыдущем этапе. Для каждого термина известен его класс $c \in C$. Пусть множество всех классов отношений ПрО обозначено *Rel*. Необходимо найти все такие пары $t_1, t_2 \in T_s$, где между $t_1$ и $t_2$ есть семантическое отношение $rel \in Rel$.

**Алгоритм 2.** Общий алгоритм извлечения семантических отношений между терминами.
**Input:** $T_s \subset T$, классы терминов $C_s \subset C$, БЯМ $L$, онтология $O$
**Output:** Relations = { $(t_1, rel, t_2) \mid t_1, t_2 \in T_s, rel \in Rel$ }
1: term_pairs = organize_pairs ( $T_s$, $C_s$ )
2: **for all** ($t_1$, $c_1$, $t_1$, $c_2$) ∈ term_pairs do
3: relation_classes ← $O$.get_relations ( $c_1$, $c_2$ )
4: **if** relation_classes = ∅ **do**
5: continue
6: **end if**
7: cls_prompt ← create_cls_prompt ( $s$, $t_1$, $t_1$, relation_classes )
8: $rel$ ← $L$.invoke ( cls_prompt )
9: **if** $rel$ = null **do**
10: continue
11: **end if**
12: yield ($t_1$, $rel$, $t_1$)
13: **end for**

Процедура *organize_pairs* обеспечивает перебор всех возможных пар, найденных в текстовом фрагменте терминов и их классов (строка 1). Получение множества возможных семантических отношений между двумя классами онтологии *O* осуществляется функцией *get_relations* (строка 3). Генерация инструкций для проверки наличия отношения между терминами в заданном контексте *s* и выполнение запросов к модели *L* отражено в строках 7 и 8 соответственно.

## 3 Построение онтологии компьютерной лингвистики

Для проведения исследования собран корпус текстов с русскоязычного веб-сайта Хабр[5], связанных с направлением КЛ, объёмом около 1,5 тысячи текстов. Отбор необходимых статей происходил на основе собранного списка хабов[6]. На основе данного корпуса проводился анализ ПрО. Выбор источника определялся его актуальностью, постоянным обновлением содержания и наличием материалов, отражающих современные тенденции развития данной области. В качестве базовой онтологии использовалась онтология научной ПрО [19], содержащая 11 классов (*Деятельность, Задача, Информационный ресурс, Метод исследования, Объект исследования, Организация, Предмет исследования, Публикация, Раздел науки, Результат или продукт, Событие*) и 92 отношения.

При конкретизации онтологии на область КЛ использовались корпусные методы анализа собранных текстов, а также рассмотренные выше ресурсы той же направленности[2,3]. В результате, во-первых, удалены некоторые классы и отношения, относящиеся к научной деятельности, а не к знаниям; во-вторых, добавлены новые классы, такие как *Модель*, *Набор_данных*, *Метрика*, *Приложение* и *Окружение*; в-третьих, добавлены новые подклассы таким классам, как *Метод_исследования* (рисунок 4).

Созданная онтология КЛ содержит 15 классов (10 из базовой онтологии) и 111 отношений (92 из базовой онтологии). Для экспериментальных исследований классы и отношения,

[5] https://habr.com/ru/feed/.
[6] Раздел сайта, где публикуются материалы на определённую тему.

которые в текстах упоминались редко, исключены из рассмотрения. В итоге для разметки данных и построения наборов данных для экспериментов использовалось 15 классов и 51 отношение.

Создание терминологического ядра онтологии описано в [8]. Система классов ПрС автоматически генерируется на основе системы классов онтологии по набору заданных шаблонов. Можно рассмотреть данную методику на нескольких примерах.

a. Шаблон {*класс*}.*Название*: термины, относящиеся к названию объектов класса Задача, получают лексико-семантический класс *Задача.Название*,
b. Шаблон {*класс*}.{*атрибут*}: термины, относящиеся к значению атрибута Значение класса Метрика, получают лексико-семантический класс *Метрика.Значение*,
c. Шаблон {*класс*}.{*отношение*}.{*класс*}: термины, относящиеся к отношению *обучена_на* между объектами класса *Модель* и объектами класса *Набор_данных*, получат лексико-семантический класс *Модель.обучена_на.Набор_данных*.

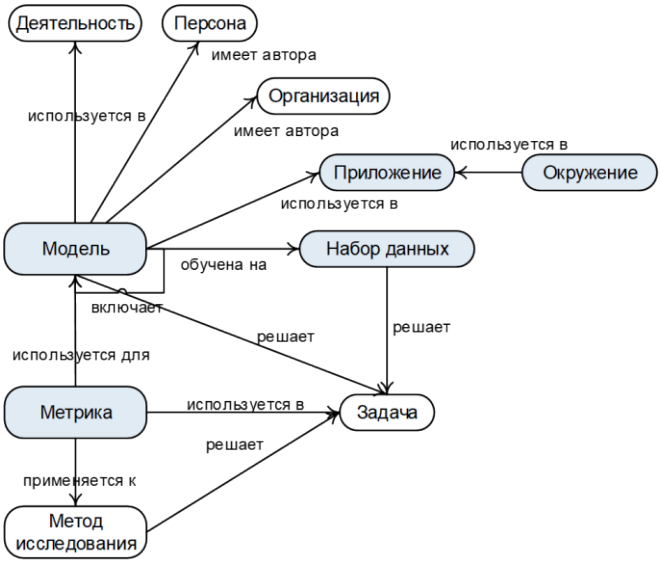



Рисунок 4 – Фрагмент онтологии по КЛ

Начальное наполнение ПрС осуществлено в два этапа.

На первом этапе проведён анализ русско-английского тезауруса КЛ[2]. Из него отобраны подходящие термины, относящиеся к следующим категориям (классам): *Деятельность, Задача, Информационный_ресурс, Метод_исследования, Метрика, Объект_исследования, Предмет_исследования, Раздел_науки* и *Результат*. Всего отобрано 585 терминов.

На втором этапе термины тезауруса автоматически сопоставлены терминам на портале КЛ[3], который содержит онтологию научной деятельности, включающую описанные классы. Для терминов подобраны соответствующие онтологические классы. Для тех терминов, которые не найдены на портале, указаны классы их синонимов или более общих понятий, если такие имелись. Оставшиеся термины размечены вручную. В результате получен ПрС, содержащий 2640 терминов.

## 4 Извлечение терминов

На первом этапе подхода (см. раздел 2.2) – поиска кандидатов в термины – применялась библиотека *Spacy* и набор ЛСШ для русского языка [22]. Данный набор включает 16 типовых шаблонов для сборки словокомплексов на основе грамматических свойств слов.

Для каждого выделенного слова или словокомплекса осуществлён поиск семантически близких к нему терминов в словаре на основе *векторного представления слов*. На данном этапе применена модель, которая вычисляет векторное представление для всех терминов в словаре, а также векторные представления найденных в тексте кандидатов. Семантическая близость между терминами и кандидатами определяется как косинусная мера между двумя векторными представлениями.

Найденные термины и их классы (максимум шесть вариантов) помещаются в шаблон инструкции, предназначенной для предсказания терминов выбранной ПрО. В инструкции указываются подробные описания классов терминов, сформированных на подготовительном

этапе. Например, для класса *Метод_исследования* использовано следующее описание: *способы, приёмы, при помощи которых осуществляется исследование*.

Шаблон инструкции хранится в виде файла, который находится в окружении программного обеспечения, и во время исполнения программы в него подставляются конкретные значения терминов, классов и описаний классов. Пример автоматически сгенерированных запроса и ответа системы.

**Текст:** *Компания Яндекс разработала голосовой помощник под названием Алиса.*
**Словосочетание:** *голосовой помощник*
**Возможные классы:** *Model*, *Application*
**Описания классов:**
- *Model*: модель машинного обучения.
- *Application*: самостоятельная программа, предоставляющая функции и возможности.
Является ли словосочетание «*голосовой помощник*» термином в этом тексте?
**Ответ:** да, *Application*

В качестве БЯМ выбрана модель (*Mixtral-8x7b-Instruct*), которая получает на вход инструкцию и генерирует ответ, из которого извлекается результат: является ли слово или словокомплекс термином и (в случае положительного ответа) к какому классу из представленных оно относится в данном контексте.

В целях повышения точности извлечения и классификации терминов предложено использовать дополнительный запрос для верификации предсказаний системы с применением следующего шаблона инструкции.

**[INST]**
Задача состоит в том, чтобы определить, является ли объект представителем класса «{*class*}» или нет.
Используй формат да/нет.
Описание класса «{*class*}»: {*description*}.
Давай ответы только в рамках указанного ниже текста.
**[/INST]**
**Текст:** «{*text*}»
Является ли «{*term*}» термином класса «{*class*}» в рамках этого текста?
**Ответ:**

В данный шаблон подставляются название и описание предполагаемого класса, термин и текст, в котором он находится. Модель должна подтвердить, что в рамках данного контекста этот термин относится к указанному классу. Используемый формат ответа выбран коротким: БЯМ нужно сгенерировать всего один выходной токен («да» или «нет»), что позволяет системе работать быстрее.

Для оценки качества извлечения терминов использовался набор данных из [11], который включает 1088 предложений, выбранных из корпуса, и содержит *BIO*-разметку 3136 терминов и 1517 отношений. Результаты извлечения терминов, полученные с использованием двух рассмотренных подходов, представлены в таблице 1. Из таблицы видно, что применение комплексного подхода увеличивает полноту.

Таблица 1 – Результаты извлечения терминов

| Метод | Полнота | Точность | F1-мера |
|---|---|---|---|
| *ruRoBERTa* | 0.70 | 0.84 | 0.76 |
| *ЛСШ + Словарь + Sentence-BERT + Mixtral* | **0.75** | 0.85 | **0.80** |

Для анализа классификации терминов выполнено сравнение с подходом для одновременного извлечения и классификации терминов. Полученные результаты представлены в таблице 2. Из таблицы 2 видно, что применение комплексного подхода увеличивает полноту, но уступает традиционному подходу в точности.

Таблица 2 – Результаты извлечения и классификации терминов

| Метод | Полнота | Точность | F1-мера |
|---|---|---|---|
| *ruRoBERTa (извлечение) + ruRoBERTa (классификация)* | 0.71 | **0.83** | **0.73** |
| *ЛСШ + Словарь + Sentence-BERT + Mixtral* | **0.83** | 0.73 | 0.71 |

## 5 Извлечение отношений

Извлечение отношений между терминами предлагается осуществлять на основе БЯМ.

### 5.1 Разработка инструкций для извлечения отношений

Для предсказания отношений между терминами на основе БЯМ используется инструкция, уникальная для каждой пары классов терминов, между которыми возможно отношение. Инструкция делится на несколько логических частей, которые можно рассмотреть на примере пары классов *Модель_обучения* и *Организация*.

Вначале располагается краткое описание задачи и список вспомогательных вопросов, которые раскрывают семантику отношения. Далее указываются несколько примеров с наличием или отсутствием отношения между терминами этих классов.

Наконец, приводятся конкретные входные данные, для которых модель должна дать ответ. Модель получает на вход термины, для которых нужно указать отношение, связывающее их, а также контекст, в котором они располагаются. Инструкция предназначена для того, чтобы получить от модели ответ о наличии или отсутствии отношения между двумя терминами. Так, она должна ответить *Да* и вывести название отношения из списка, если оно есть, и *Нет*, если его нет.

### 5.2 Обогащение инструкций на основе онтологии

Для извлечения отношений и предоставления бóльшего контекста модели предложено использовать в инструкциях ВОК – набор вопросов на ЕЯ, которые сопоставляются отношениям онтологии и позволяют пользователям уточнить смысл, вкладываемый авторами онтологии при их создании, а также провести проверку на соответствие при добавлении новых данных. ВОК позволяют задать начальные синтаксические и/или семантические ограничения на извлекаемые связи.

Онтологические шаблоны содержат в своём описании набор ВОК. Для получения необходимого и достаточного набора ВОК предложен подход для их автоматической генерации с помощью БЯМ. В данной работе генерация ВОК проводилась с помощью модели *GPT*-4. Для создания набора ВОК необходимо выполнить следующие шаги.

1) разработать шаблон инструкции.
2) автоматически сгенерировать инструкции для всех отношений и атрибутов классов онтологии на основе предложенного шаблона.
3) отправить запросы-инструкции и получить ответы от модели.
4) проанализировать ответы и составить набор ВОК.

Для генерации вопросов, содержащих отношения между двумя классами, использовалась техника [17]. Инструкция состоит из двух частей: постановка задачи и пример. Например, для генерации нескольких вопросов, содержащих отношение между классами *Метрика* и *Модель*, использовалась следующая инструкция:

Приведи 5 примеров вопросов, в которых *Метрика* применяется для *Модели*. В вопросах должны быть слова «*Метрика*» и «*Модель*».

**Пример:** *Может ли метод исследования использоваться для решения задачи?*

В данном случае в инструкции представлены названия для классов *Метрика* и *Модель*, которые связывает *отношение* применяется для *Модели* из онтологии. Пример в данном случае демонстрирует лишь структуру вопроса.

Ответ GPT-4 на данную инструкцию включал следующие вопросы:

- *Какую метрику можно использовать для оценки качества модели обучения?*
- *Может ли метрика быть использована для проверки эффективности модели обучения?*

- *Как применяется метрика в модели обучения?*
- *Может ли метрика служить надёжным индексом для модели обучения?*
- *Каково значение метрики и как она используется в модели обучения?*

Итоговый вариант шаблона инструкций для определения отношения между терминами выглядит следующим образом:

Твоя задача состоит в определении отношений между терминами, если таковые есть.
Для ответа на этот вопрос тебе помогут следующие вопросы:
{*questions_list*}
Примеры наличия и отсутствия отношения между терминами:
{*examples_list*}
<...>
Помни, что между терминами класса {*class1*} и {*class2*} возможны ТОЛЬКО следующие отношения: {*relations_list*}
Есть ли подходящее отношение между терминами «<термин-1>» и «<термин-2>» в этом тексте?
{*text*}

Результаты извлечения семантических отношений приведены в таблице 3, где отражены самые представительные классы отношений, т.е. те классы, для которых имелось более 30 примеров в тестовом наборе данных. Среднее значение F1-меры составило 87%. Достигнут порог в 60% точности для всех отношений.

Таблица 3 – Результаты извлечения семантических отношений

| Класс отношений | F1-мера |
|---|---|
| Модель.поддерживает.Язык | 0.98 |
| Задача_исследования.решается_в.Раздел_науки | 0.98 |
| Метрика.имеет.Значение | 0.97 |
| Метрика.используется_для.Модель | 0.94 |
| Метрика.используется_в.Задача_исследования | 0.94 |
| Метод_исследования.применяется_к.Объект_исследования | 0.93 |
| Модель.используется_для_решения.Задача_исследования | 0.93 |
| Метод_исследования.решает.Задача_исследования | 0.92 |
| Приложение.имеет_автора.Организация | 0.91 |
| Набор_данных.составлен_для_решения. Задача_исследования | 0.90 |
| Приложение.используется_для_решения. Задача_исследования | 0.89 |
| Объект_исследования.используется_в_решении. Задача_исследования | 0.89 |
| Приложение.применяется_к.Объект_исследования | 0.88 |
| Модель.имеет_автора.Организация | 0.83 |
| Метод_исследования.использует.Модель | **0.64** |
| Модель.является_примером.Модель | **0.26** |
| Метод_исследования.является_частью.Метод_исследования | **0.12** |

Для генерации ВОК, отражающих связь между экземплярами классов и атрибутами, использовалась генерация синонимичного ряда вопросов:

Напиши синонимичные вопросы к «*Какая дата появления у Метода исследования?*».

Постобработка ВОК представляет собой внесение исправлений в вопросы вручную или автоматически. Автоматическая обработка может строиться следующим образом.

1) Поиск и удаление из корпуса вопросов, не содержащих названия обоих классов.
2) Замена некоторых слов их синонимами в случаях неточного названия класса (например, если в онтологии существует класс *Организация*, а в предложении употребляется термин «*компания*», то его необходимо заменить на термин «*организация*»).
3) Удаление или замена конкретных наименований на более общие (например, «*Яндекс*» может быть заменен на «*организация*»).

Использование приведённых шаблонов инструкций с примерами позволяет уменьшить трудоёмкость обработки вопросов.

## 6 Анализ результатов

Анализ ошибок извлечения терминов позволил установить следующее.

- При отсутствии верификации ответов модели все методы демонстрируют ошибки извлечения терминов, для которых не существует подходящего онтологического класса.

- Некорректное отнесение терминов к другой категории, когда он является частью или пересекается с названием категории (например, термин «*данные*» был отнесен к категории *Набор_данных*).
- Один и тот же термин извлекается по-разному в разных контекстах, что связано с тем, что модель определяет принадлежность термина к определённому классу с разной степенью вероятности в зависимости от контекста.
- Термины, относящиеся к таким категориям, как *Деятельность, Набор_данных* и *Объект_исследования* извлекаются хуже (см. таблицу 4).

Таблица 4 – Результаты извлечения и классификации терминов отдельно по классам

| Класс терминов | Полнота | Точность | F1-мера |
|---|---|---|---|
| Значение | 1.0 | 0.83 | 0.9 |
| Язык | 0.79 | 0.97 | 0.87 |
| Модель | 0.82 | 0.88 | 0.84 |
| Библиотека | 0.8 | 0.8 | 0.8 |
| Дата | 0.66 | 1.0 | 0.79 |
| Раздел науки | 1.0 | 0.65 | 0.78 |
| Организация | 0.65 | 1.0 | 0.78 |
| Приложение | 0.87 | 0.7 | 0.77 |
| Задача исследования | 0.67 | 0.72 | 0.69 |
| Метод исследования | 0.72 | 0.68 | 0.69 |
| Персона | 0.57 | 0.8 | 0.66 |
| Метрика | 0.66 | 0.62 | **0.63** |
| Информационный ресурс | 0.71 | 0.58 | **0.63** |
| Объект исследования | 0.96 | 0.45 | **0.61** |
| Набор данных | 0.93 | 0.42 | **0.57** |
| Деятельность | 0.71 | 0.31 | **0.43** |

Основная сложность заключается в определении всего словосочетания, содержащего термин. Например, к категории *Деятельность* были отнесены такие термины, как «*модель проекта*» вместо «*проект*», «*ход экспериментов*» вместо «*эксперименты*», «*продолжение экспериментов с оптимизациями*» вместо «*эксперименты*».

В качестве примеров ошибочного предсказания класса *Объект исследования* можно привести термины «*dataset*» (класс *Набор_данных*) и «*InfoNCE*» (класс *Метрика*).

- Один и тот же термин извлекается по-разному в разных контекстах, что связано с тем, что модель определяет принадлежность термина к определённому классу с разной степенью вероятности в зависимости от контекста.

В результате анализа ошибок извлечения отношений установлены следующие факторы.

- Оптимальное количество ВОК для генерации моделью *GPT*-4 определено экспериментально и составило 5.
- Некоторые отношения неправильно извлечены из-за неверной классификации терминов.
- Выявлены сложности в определении отношения включения между терминами некоторых классов и в дифференциации отношений между терминами.

В процессе работы над ошибками предприняты попытки модифицировать инструкции (улучшены формулировки названий отношений, изменены примеры, введены специальные инструкции для некоторых отношений). В результате удалось подобрать модификации, которые дали общее улучшение качеств извлечения отношений.

Проведённые исследования показали, что привлечение эксперта повышало качество полученных решений на каждом этапе исследований: от внедрения шаблонов для сборки терминов-словокомплексов, формирования определения для классов терминов и до разработки эффективных инструкций для БЯМ.

## Заключение

Предлагаемая методика извлечения информации из текстов на основе онтологии и БЯМ включает: разработку верхнего уровня предметной онтологии; сбор корпуса текстов и подготовку наборов данных; создание ПрС; генерацию ВОК; разработку шаблонов инструкций к

БЯМ; автоматическое извлечение терминов на основе словаря и БЯМ; автоматическое извлечение отношений на основе БЯМ и ВОК.

Проведённые эксперименты для ПрО КЛ показали, что предложенные методы могут достигать хороших значений полноты и точности для задачи извлечения терминов (F1=0.8) и для задачи извлечения семантических связей между ними (F1=0.87).

Привлечение эксперта повышает качество полученных решений на каждом этапе исследований. Уточнение инструкций, применение в них показательных примеров, добавление отрицательных примеров и корректных ВОК позволяет повысить качество извлечения терминов и отношений.

## Авторский вклад

Концепция и план исследования: Е.С. (Елена Сидорова) и А.И. (Александр Иванов); сбор и разметка данных: К.О. (Кристина Овчинникова); реализация программной части: А.И.; анализ и интерпретация результатов: Е.С., К.О. и А.И.; подготовка текста статьи: Е.С. и К.О.

## Список источников


[1] ***Массель Л.В.*** Экосистема знаний как развитие и специализация цифровой экосистемы // Труды Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии–2023». Таганрог: Издатель Ступин С.А., 2023. С.155-164.

[2] ***Лукашевич Н.В., Добров Б.В.*** Проектирование лингвистических онтологий для информационных систем в широких предметных областях. *Онтология проектирования*. Том 5. №1 (15). 2015. С.47-69.

[3] ***Ehrlinger L., Wöß W.*** Towards a Definition of Knowledge Graphs // Joint Proceedings of the Posters and Demos Track of 12th International Conference on Semantic Systems (SEMANTiCS2016) and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16). Leipzig, 2016. P.13–16.

[4] ***Fernández-López M., Gómez-Pérez A., Pazos A., Pazos J.*** Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems & their applications*, 1999, 4(1). P.37–46.

[5] ***Sure Y., Staab S., Studer R.*** On-To-Knowledge Methodology. *Handbook on Ontologies*. 2003. № 6. P.135–152.

[6] ***Uschold M., King M.*** Towards a Methodology for Building Ontologies. *Proc. of the Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal, Canada. 1995. P.6.1–6.10.

[7] ***Braga J., Dias Joaquim L.R., Regateiro F.*** A Machine Learning Ontology, 2023. DOI: 10.31226/osf.io/rc954.

[8] ***Овчинникова К., Иванов А., Сидорова Е.*** Автоматизация построения терминологического ядра онтологии по компьютерной лингвистике на основе корпуса текстов. *Системная информатика*. 2023. № 23. С.13-32.

[9] ***Kim Sang E., Meulder F.*** Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition // Proc. of the 7th Conference on Natural Language Learning at HLT-NAACL 2003. P.142–147.

[10] ***Гусев В.Д., Саломатина Н.В.*** Метод итерационного построения шаблонов для поиска в текстах по катализу информации о химических процессах и условиях их протекания. *Информационные и математические технологии в науке и управлении*. 2016. № 4-1. С.37–45.

[11] ***Zhu F., Shen B.*** Combined SVM-CRFs for biological named entity recognition with maximal bidirectional squeezing // PLoS ONE 7(6): e39230. 2012. DOI: 10.1371/journal.pone.0039230.

[12] ***Ganaie M.A., Hu Minghu, Malik A.K., Tanveer M., Suganthan P.N***. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*. 2022. P. 28-29

[13] ***Li J., Sun A., Han J., Li C.*** A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge & Data Engineering*. 2022. Vol.34. N.1. P.50-70.

[14] ***Лагутина Н.С., Васильев А.М., Зафиевский Д.Д.*** Задачи в области распознавания именованных сущностей: технологии и инструменты. *Моделирование и анализ информационных систем*. 2023. №30(1). С.64-85.

[15] ***Brown T., Mann B., Ryder N., Subbiah M., Kaplan J.D.*** Language Models are Few-Shot Learners // In: Advances in Neural Information Processing Systems. Vol.33. Curran Associates, Inc., 2020. P.1877-1901.

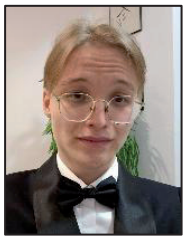
[16] ***Wei Jason, Bosma Maarten, Zhao Vincent Y., Guu Kelvin*** Finetuned language models are zero-shot learners. *Conference paper at ICLR* 2022. 2022. DOI: 10.48550/arXiv.2109.01652.

[17] ***Kaplan J., McCandlish S., Henighan T., Brown T.B., Chess B., Child R., Gray S., Radford A., Wu J., Amodei D.*** Scaling Laws for Neural Language Models. 23 Jan 2020. 19 p. DOI: 10.48550/arXiv.2001.08361.

[18] ***Shichao Sun, Ruifeng Yuan, Ziqiang Cao, Wenjie Li, Pengfei Liu***. Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization // In Findings of the Association for Computational Linguistics ACL 2024. Bangkok, Thailand and virtual meeting. Association for Computational Linguistics, 2024. P.7551–7558.
[19] ***Zagorulko Yu.A., Borovikova O.I.*** Using a System of Heterogeneous Ontology Design Patterns to Develop Ontologies of Scientific Subject Domains // Programming and Computer Software. 2020. 46(4). P.273–280.
[20] ***Gangemi A., Presutti V.*** Ontology Design Patterns // Handbookon Ontologies. Springer, 2009. P.221-243.
[21] ***Dewayne Whitfield*** Using GPT-2 to Create Synthetic Data to Improve the Prediction Performance of NLP Machine Learning Classification Models // Computation and Language. arXiv preprint arXiv: 2104.10658, 2020.
[22] ***Sidorova E., Akhmadeeva I.*** The software environment for multi-aspect study of lexical characteristics of text // In: Alexander Elizarov, Boris Novikov, Sergey Stupnikov (eds.): Data Analytics and Management in Data Intensive Domains. Proc. of the XXI International Conference DAMDID/RCDL'2019. Kazan, 2019. P.380-389.

## Сведения об авторах

***Сидорова Елена Анатольевна***, 1977 г. рождения. Окончила НГУ в 2000 г., к.ф.-м.н. (2006). Старший научный сотрудник лаборатории искусственного интеллекта Института систем информатики им. А.П. Ершова, доцент кафедры программирования и кафедры систем информатики НГУ, член Российской ассоциации искусственного интеллекта. В списке научных трудов более 170 работ в области компьютерной лингвистики, онтологического инжиниринга, анализа аргументации и разработки интеллектуальных систем. Author ID (РИНЦ): 146000; ORCID: 0000-0001-8731-3058; Author ID (Scopus): 41961707000; Researcher ID (WoS): K-2432-2018. *lsidorova@iis.nsk.su.* ✉.



***Иванов Александр Иванович***, 2002 г. рождения. Окончил НГУ в 2024 г. по направлению информатика и вычислительная техника. Программист, эксперт по аналитическим данным. Author ID (РИНЦ): 176166. *a.ivanov15@alumni.nsu.ru.*

***Овчинникова Кристина Алексеевна***, 2000 г. рождения. Окончила НГУ в 2024 г. по направлению фундаментальная и прикладная лингвистика, магистр. Специалист (NLP разработчик). В списке научных трудов 7 работ в области компьютерной лингвистики. ORCID: 0000-0002-7400-0586; Author ID (РИНЦ): 1267926; Author ID (Scopus): 57374190900. *k.ovchinnikova2@alumni.nsu.ru.*

*Поступила в редакцию 02.12.2024, после рецензирования 16.01.2025. Принята к публикации 20.01.2025.*



*Scientific article* DOI: 10.18287/2223-9537-2025-15-1-114-129

# Information extraction from texts based on ontology and large language models

**© 2025, E.A. Sidorova[1,2]✉, A.I. Ivanov[2], K.A. Ovchinnikova[2]**

[1] *A.P. Ershov Institute of Informatics Systems SB RAS, Novosibirsk, Russia*

[2] *Novosibirsk State University (NSU), Novosibirsk, Russia*

**Abstract**

The article examines the extraction of information from texts using the ontology of a subject area combined with neural network-based text analysis methods, including the use of large language models. It discusses the expert's role in developing and maintaining systems, illustrated through the task of extracting information from analytical articles and constructing ontologies in computational linguistics to represent key concepts relevant to the system's user or customer. The process of ontology creation is accompanied by the development of a dictionary that forms the ontology's termino-

logical core, followed by methods for extracting new terms within the specified subject area. This task is considered as a named entity recognition problem, traditionally addressed by training a neural network model on a representative dataset. The study compares this approach with a methodology leveraging large language models. For this, lexical and syntactic patterns, as well as instruction patterns for hypothesis testing regarding new term-phrases and result verification, were developed. The developed instructions for solving the problem of relation extraction also include the automated generation of natural language competency assessment questions for each ontology relation. The novelty of the proposed approach lies in the integration of ontological, linguistic and neural network approaches to extract information from texts. The study demonstrates the possibility of solving tasks of text analysis and information extraction problems through a chain of large language models, with dynamically generated instructions based on the outcomes of prior analysis stages. The following F1-measure scores were achieved in the experiments: F1=0.8 for term extraction and classification and F1=0.87 for relation extraction.

***Keywords:*** *information extraction, domain ontology, large language model, neural network models, prompt engineering.*

***For citation:*** *Sidorova EA, Ivanov AI, Ovchinnikova KA*. Information extraction from texts based on ontology and large language models [In Russian]. *Ontology of designing*. 2025; 15(1): 114-129. DOI:10.18287/2223-9537-2025-15-1-114-129.

***Conflict of interest:*** The authors declare no conflict of interest.

## List of figures and tables



## References

[1] ***Massel LV.*** Knowledge ecosystem as development and specialization of digital ecosystem [In Russian]. Proc. of the International Scientific and Technical Congress “Intellectual Systems and Information Technologies-2023”. Taganrog: Publisher Stupin S.A., 2023: 155-164.

[2] ***Loukachevitch NV, Dobrov BV.*** Developing linguistic ontologies in broad domains [In Russian]. *Ontology of Designing*. 2015; 5(1): 47-69.

[3] ***Ehrlinger L, Wöß W.*** Towards a Definition of Knowledge Graphs. Joint Proc. of the Posters and Demos Track of 12th Int. Conf. on Semantic Systems (SEMANTiCS2016) and 1st Int. Workshop on Semantic Change & Evolving Semantics (SuCCESS16). Leipzig, 2016: 13–16.

[4] ***Fernández-López M, Gómez-Pérez A, Pazos A, Pazos J.*** Building a Chemical Ontology Using Methontology and the Ontology Design Environment. IEEE Intelligent Systems & their applications. 1999; 4(1): 37–46.

[5] ***Sure Y, Staab S, Studer R.*** On-To-Knowledge Methodology. Handbook on Ontologies. 2003; 6: 135–152.

[6] ***Uschold M, King M.*** Towards a Methodology for Building Ontologies. Proc. of the Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, Canada. 1995: 6.1–6.10.

[7] ***Braga J, Dias JLR, Regateiro F.*** A Machine Learning Ontology, 2023. DOI: org/10.31226/osf.io/rc954.

[8] ***Ovchinnikova K, Ivanov A, Sidorova E.*** Automation of the construction of the terminological core of ontology in computer linguistics based on a corpus of texts [In Russian]. System Informatics. 2023; 23: 13-32.

[9] ***Kim SE, Meulder F.*** Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition // Proc. of the 7th Conference on Natural Language Learning at HLT-NAACL 2003: 142–147.

[10] ***Gusev VD, Salomatina NV.*** Method of iterative construction of templates for searching in texts on catalysis information about chemical processes and conditions of their occurrence [In Russian]. *Information and mathematical technologies in science and management*. 2016; (1): 37–45.

[11] ***Zhu F, Shen B.*** Combined SVM-CRFs for biological named entity recognition with maximal bidirectional squeezing. PLoS ONE 7(6): e39230. 2012. DOI: 10.1371/journal.pone.0039230.

[12] ***Ganaie MA, Hu Minghu, Malik AK, Tanveer M, Suganthan PN***. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*. 2022: 28-29.
[13] ***Li J, Sun A, Han J, Li C.*** A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge & Data Engineering*. 2022; 34(1): 50-70.
[14] ***Lagutina NS, Vasiliev AM, Zafievsky DD.*** Tasks in the field of named entities recognition: technologies and tools [In Russian]. *Modeling and analysis of information systems*. 2023; 30(1): 64-85.
[15] ***Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD.*** Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems, vol.33. Curran Associates, Inc.; 2020: 1877-1901.
[16] ***Wei J, Bosma M, Zhao VY, Guu K.*** Finetuned language models are zero-shot learners. Conference paper at ICLR 2022. DOI:10.48550/arXiv.2109.01652, 2022.
[17] ***Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, Gray S, Radford A, Wu J, Amodei D.*** Scaling Laws for Neural Language Models. 23 Jan 2020. 19 p. DOI: 10.48550/arXiv.2001.08361.
[18] ***Sun S, Yuan R, Cao Z, Li W, Liu P.*** Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization. In: Findings of the Association for Computational Linguistics ACL 2024. Bangkok, Thailand and virtual meeting. Association for Computational Linguistics; 2024: 7551–7558.
[19] ***Zagorulko YA, Borovikova OI.*** Using a System of Heterogeneous Ontology Design Patterns to Develop Ontologies of Scientific Subject Domains. Programming and Computer Software. 2020; 46(4): 273–280.
[20] ***Gangemi A, Presutti V.*** Ontology Design Patterns. Handbookon Ontologies. Springer, 2009: 221-243.
[21] ***Whitfield D.*** Using GPT-2 to Create Synthetic Data to Improve the Prediction Performance of NLP Machine Learning Classification Models. Computation and Language. arXiv preprint arXiv: 2104.10658, 2020.
[22] ***Sidorova E, Akhmadeeva I.*** The software environment for multi-aspect study of lexical characteristics of text. In: Alexander Elizarov, Boris Novikov, Sergey Stupnikov (eds.): Data Analytics and Management in Data Intensive Domains. Proc. of the XXI International Conference DAMDID/RCDL'2019. Kazan, 2019: 380-389.

## About the authors

***Elena Anatolievna Sidorova*** (b. 1977) graduated from the NSU in 2000, PhD (2006). She is a Senior Researcher of the Laboratory of Artificial Intelligence at the A.P. Ershov Institute of Informatics Systems, and an Associate Professor at Novosibirsk State University. She is a member of Russian and European Associations for Artificial Intelligence. There are more than 170 peer-reviewed publications in the field of Computational Linguistics, Argument Mining, Intelligent System Development, Knowledge and Ontology Engineering. Author ID (RSCI): 146000; ORCID: 0000-0001-8731-3058; Author ID (Scopus): 41961707000; Researcher ID (WoS): K-2432-2018. *lsidorova@iis.nsk.su* ✉.

***Alexander Ivanovich Ivanov*** (b. 2002) graduated from the NSU in 2024 with a degree in Computer Science and Engineering. Programmer, Data scientist. Author ID (RSCI): 176166. *a.ivanov15@alumni.nsu.ru.*

***Kristina Alekseevna Ovchinnikova*** (b. 2000) graduated from the NSU in 2024 with a degree in Fundamental and Applied Linguistics. Specialist (NLP Developer). The list of scientific works includes 7 publications in the field of computational linguistics. ORCID: 0000-0002-7400-0586; Author ID (RSCI): 1267926; Author ID (Scopus): 57374190900. *k.ovchinnikova2@alumni.nsu.ru*.

*Received December 2, 2024. Revised January 16, 2025. Accepted January 20, 2025.*