

# ПРИМЕНЕНИЕ ГЕНЕРАТИВНЫХ ПОДХОДОВ В СЕМАНТИЧЕСКОМ АНАЛИЗЕ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

## APPLICATION OF GENERATIVE APPROACHES IN SEMANTIC ANALYSIS OF NATURAL LANGUAGE TEXTS

A. Kuzmenko  
V. Kireev

*Summary.* It is estimated that about 80 % of corporate data is unstructured, which makes it difficult to process and analyze it using traditional methods. Extracting relationships between entities from this data in the form of relational triples allows you to transform unstructured information into a structured form, facilitating access, analysis, and use of knowledge. The paper examines modern approaches to extracting relational triples from natural language texts based on sequence transformation technology. The authors have identified and structured the existing solutions into a set of groups of methods: classical sequence transformation methods, sequence transformation methods into sets, and prompt methods. A comparative analysis of the advantages and disadvantages of these methods is carried out. A methodology for building an auxiliary system for extracting relational triples is proposed.

*Keywords:* relational triple, neural network, natural language processing, transformers, seq-to-seq, set-to-seq, large language models.

### Актуальность

Реляционная тройка представляет собой каркас предложения или некоторого текстового фрагмента в форме субъекта, объекта и отношения между ними. Сущности, представленные субъектом и объектом — это люди, организации, местоположения и другие. Отношения — человек-принадлежность, организация-местоположение, например, конкретное лицо может быть аффилировано с определенной организацией. Как правило, набор отношений имеет фиксируемый размер, однако, существуют методы извлечения реляционных троек, способные убирать это ограничение.

Целью извлечения реляционных троек является идентификация всех возможных троек в данном предложении или тексте. Более формально, для предложения или фрагмента текста  $s = \{s_1, \dots, s_n\}$  требуется смоделировать условную вероятность целевых тройных множеств [1-2]  $Y = \{(s, r_1, o_1), \dots, (s, r_k, o_k)\}$ .

**Кузьменко Андрей Владимирович**  
аспирант, Национальный исследовательский  
ядерный университет «МИФИ»  
andrey\_kuzmenko2907@mail.ru

**Киреев Василий Сергеевич**  
кандидат технических наук, доцент, Национальный  
исследовательский ядерный университет «МИФИ»  
vskireev@mephi.ru

*Аннотация.* По оценкам, около 80 % корпоративных данных являются неструктурированными, что затрудняет их обработку и анализ традиционными методами. Извлечение отношений между сущностями из этих данных в форме реляционных троек позволяет преобразовать неструктурированную информацию в структурированную форму, облегчая доступ, анализ и использование знаний. В работе рассматриваются современные подходы извлечения реляционных троек из текстов на естественном языке, основанные на технологии преобразования последовательностей. Авторами выделены и структурированы существующие решения в набор групп методов: классические методы преобразования последовательностей, методы преобразования последовательностей в множество и prompt-методы. Проводится сравнительный анализ преимуществ и недостатков этих методов. Предлагается методология построения вспомогательной системы извлечения реляционных троек.

*Ключевые слова:* реляционная тройка, нейронная сеть, обработка естественного языка, seq-to-seq, set-to-seq, большие языковые модели.

$$p(Y | s, \theta) = p_L(n | s) \prod_{i=1}^n p(Y_i | s, Y_{j \neq i}, \theta) \quad (1)$$

где  $p_L(n|s)$  — моделирует размер набора реляционных троек;  $p_i(Y_i | s, Y_{j \neq i}, \theta)$  — моделирует тройку, при условии, что она связана не только с предложением, но и с другими тройками  $Y_{j \neq i}$ ; — параметры модели.

Современные технологии, такие как семантический веб, онтологии и базы знаний, требуют точного и структурированного представления информации [3]. Извлечение реляционных троек является фундаментальным шагом в построении таких структур, обеспечивая семантическое обогащение данных и улучшая качество информационного поиска. Кроме того, это способствует развитию интеллектуальных агентов и чат-ботов, способных понимать и генерировать осмысленные ответы на основе извлеченных знаний. Продолжение исследований в этой области способствует созданию более совершенных алгоритмов и методов, способных удов-

летворить растущие потребности различных отраслей в качественной и оперативной обработке информации.

**Цель работы**

Целью данной работы является выявление и сравнительный анализ существующих подходов к извлечению реляционных троек. В дальнейшем эти результаты предполагается использовать для совершенствования методов суммаризации и семантического анализа текстов на естественном языке.

**Состояние проблемы**

Существующие подходы к извлечению реляционных троек разделяют на четыре парадигмы [2–3]: (1) конвейерные методы [1, 2, 4–6], (2) табличные методы [7–10], (3) методы использующие дополнительную маркировку текстовых последовательностей; (4) методы на основе преобразования последовательностей [11–13]. В работе предлагается модификация данной классификации. А именно замена методов на основе преобразования последовательностей на группу генеративных методов, которая охватывает большее количество решений и отражает фундаментальные принципы, заложенные в построении такого рода систем. Группу генеративных методов предлагается разделить на классические методы преобразования последовательностей (seq-to-seq), методы преобразования последовательностей в множество (seq-to-set), prompt-методы. В работе продемонстрирована каждая из подгрупп, выделены их преимущества и недостатки.

**Классические методы преобразования последовательностей**

Развитие классических методов преобразования последовательностей пошло с задачи извлечения перекрывающихся отношений [14]. Реализуются такого рода системы, как правило, в виде кодировщик-декодер архитектур, например архитектура CopyRL [15].

Кодировщик CopyRL преобразует предложение  $s = [w_1, \dots, w_n]$ , где  $w_i$  —  $i$ -е слово, а  $n$  — длина предложения, в векторные представления при помощи двунаправленной рекуррентной сети Bi-LSTM. Для каждого  $i$ -го слова формируется выход  $o_i^f = [o_i^f : o_{n-t+1}^f]$ . Также есть работа [15], где вместо конкатенации используют усреднение.

Декодер модели выполняет генерацию векторов, из которых предсказывается каждый из элементов реляционной тройки. Он состоит из однонаправленной рекуррентной сети LSTM, механизма внимания и механизма копирования. Выражение для LSTM выглядит в сокращенной форме следующим образом:

$$o_j^D, h_j^D = g(u_j, h_{j-1}^D) \tag{2}$$

где  $g(\bullet)$  — функция декодера, представленная однонаправленной LSTM сетью;  $u_j$  — векторное описание скопированного слова или отношения, декодированного на шаге  $j - 1$ ;  $w^u$  — матрица весов;  $c_j$  — вектор внимания.

Выход декодера  $o_j^D$  используется для декодирования элементов реляционной тройки. Если  $j$  принадлежит ряду 1,4,7... прогнозируется тип отношения, если 2, 5, 8... — субъект, 3, 6, 9... — объект.

Векторное представление  $v_j$  отношения, субъекта или объекта, получившее наибольшую вероятность, используется в качестве входных данных для следующего временного шага  $v_{j+1}$ . Такой подход называется авторегрессионным [15].

Старт генерации и ее завершение выполняется подобно другим задачам, использующим подход, основанный на преобразовании последовательностей. Вектор вложения  $v_j$  в формуле 2 инициализируется векторным представлением специального стартового слова «GO», а скрытое состояние  $h_0^D$  вектором предложения  $s = [h_n^E : h_n^E]$ . Модель выполняет генерацию до тех пор, пока не встретит элемент, сигнализирующий окончание генерации.

Для обучения модели CopyRL использовалась отрицательное логарифмическое правдоподобие, вычисляемое следующим образом:

$$L = \frac{1}{B \cdot T} \sum_{i=1}^B \sum_{j=1}^T -\log(p(y_i^j | y_i^{<j}, s, \theta)), \tag{3}$$

где  $T$  — максимальный временной шаг,  $B$  — размер мини-пакета,  $\theta$  — параметры всей модели.

Модель CopyRL имеет ряд недостатков. Во-первых, при генерации объекта необходимо исключать векторное представление соответствующее предсказанному на предыдущем шаге субъекту. В противном, модель опять его предскажет, как показали в исследовании [15]. Во-вторых, модель не способна извлекать элементы реляционных троек, состоящих из нескольких слов. В-третьих, функция потерь учитывает порядок декодирования реляционных троек. Это приводит к тому, модель штрафует за то, что предсказала реляционные тройки в отличном от заданного порядке, хотя в действительности должен наблюдаться инвариант к порядку.

Первые две проблемы позволяет решать модификация CopyMTL, предложенная в [16]. Их модель основана на многозадачном обучении, для прогнозирования элементов реляционной тройки с несколькими маркерами.

Проблема маскирования решается путем отображения  $o_{j-1}^D$  и  $o_j^E$  в объединенное пространство признаков через один дополнительный нелинейный слой:

$$q_j^E = \text{selu}([o_{j+1}^D : o_j^E] \times w^e) \times w^f. \quad (4)$$

Также CopyMTL способна выделять элементы отношений, состоящих из нескольких слов. Для этого в [16] вводится дополнительный блок маркировки последовательностей. Выходные сигналы из блока кодировщика поступают в блоки CopyRe-Decoder и Sequence-Labeling. Блок CopyRe-Decoder аналогичен модели CopyRe. Блок Sequence-Labeling в [16] представлен в виде CRF, который вычисляет наиболее вероятные метки для каждого слова. Используется схема BIO (Begin, Inside, Outside). На этапе вывода, полученные теги используются для постобработки декодированных объектов. Для предсказанного элемента

Правила декодирования выглядят следующим образом: «В» объект из нескольких слов, добавляются все следующие токены «I»; «I» — объект из нескольких слов, добавляются все слова с «I» после, и все слова до «В» перед включительно; «O» — элемент состоит из одного слова.

Оценки условных вероятностей тегов для предложения  $s$  можно представить в следующем виде:

$$p(\text{tags}^* | s) = \frac{e^{\text{score}(s, \text{tags}^*)}}{\sum_{\text{tags}} e^{\text{score}(s, \text{tags}^*)}}, \quad (5)$$

$$\text{score}(s, \text{tags}) = \sum_i \varphi_{i, \text{tag}_i} + b_{\text{tag}_{i-1} \rightarrow \text{tag}_i},$$

где  $\varphi_{i, \text{tag}_i}$  — оценка тегов для  $i$ -го входного слова, который получен из скрытого состояния Bi-LSTM на временном шаге  $i$ ;

$b_{\text{tag}_{i-1} \rightarrow \text{tag}_i}$  — оценка перехода от  $\text{tag}_{i-1}$  к  $\text{tag}_i$ .

Оценки условной вероятности получаются путем оптимизации следующей функции потерь:

$$L^E = -\log(p(\text{tags}^* | s)). \quad (6)$$

Таким образом итоговая модель обучается при помощи комбинированной функции потерь:

$$L = \lambda \cdot L^E + L^D, \quad (7)$$

где  $\lambda$  — коэффициент потери при маркировке.

Хотя CopyMTL решает проблему прогнозирования реляционных троек, содержащих элементы из нескольких слов, в механизме копирования это никак не учитывается — выбирается слово с максимальной вероятностью. Также данная модель не учитывает инвариант

порядка прогнозирования реляционных троек, что негативно влияет на обучение модели.

### Преобразование преобразования последовательности в множество

Узким местом методов seq-to-seq является авторегрессионное декодирование. Этот процесс позволяет генерировать элементы реляционных троек один за другим преобразуя задачу (1) в

$$P(Y | s, \theta) = p_L(n | s) \prod_{i=1}^n p(Y_i | s, Y_{j < i}, \theta). \quad (8)$$

Строгая зависимость индексов во времени не позволяет распараллелить процесс декодирования реляционных троек. Кроме того, она не позволяет обмениваться информацией между всеми тройками, учитывая только предыдущие. Еще одним недостатком представленных seq-to-seq моделей является чувствительность функции потерь к перестановкам. Реляционные тройки, по своей природе, не имеют внутреннего порядка между собой. Однако, для адаптации авторегрессионного декодера, вывод которого представляет собой последовательность, неупорядоченные тройки должны быть отсортированы в определенном порядке на этапе обучения. Поскольку перекрестная энтропия накладывает штраф за каждую тройку, предсказанную вне позиции. Решить эти проблемы позволяет модифицированная модель seq-to-seq, которую предложили в [17]. Авторы сформулировали совместную задачу извлечения сущностей и отношений как проблему предсказания множества. Их модель также имеет кодировщик-декодер архитектуру. В качестве кодировщика используется модель типа BERT. Декодер данной модели в отличие от seq-to-seq не является авторегрессионным. Это позволяет не только избежать изучения порядка извлечения нескольких реляционных троек, но также генерировать тройки на основе двунаправленной информации, а не слева направо.

Архитектура декодера состоит из  $N$  идентичных слоев трансформера. На каждом уровне преобразователя присутствует механизма self-attention для моделирования отношений между тройками и механизм cross-attention для объединения информации данного предложения. Декодер преобразует  $m$  тройных запросов в  $m$  выходных вложений, которые затем независимо декодируются в реляционные тройки при помощи сетей прямой связи.

Основная трудность обучения данной модели состоит в том, чтобы набрать предсказанные тройки по отношению к истинным. В работе [17] предлагают использовать функцию потерь предсказания (9), которая дает оптимальное соответствие между истинными и предсказанными тройками

$$\pi^* = \operatorname{argmin}_{\pi \in \Pi(m)} \sum_{i=1}^m C_{\text{match}}(Y_i, \hat{Y}_{\pi(i)}), \quad (9)$$

где  $\Pi(m)$  — пространство всех перестановок длины  $m$ ;  $C_{\text{match}}(Y_i, \hat{Y}_{\pi(i)})$  — стоимость парного сопоставления между истинной и предсказанной тройкой с индексом.

Подход seq-to-set оказался довольно эффективным, что показывает и другая его реализация [18]. Концептуально эта модель решает задачу похожим образом, однако ее блоки имеет отличное строение. Так, кодировщик дополняется сетью BiGRU, а декодер имеет пирамидальную структуру и дополняется регрессором, который постепенно уточняет предложения, чтобы более точно находить и классифицировать интервалы.

### Анализ существующих LLM подходов

Современные большие языковые модели, такие как GPT3, ChatGPT, GPT4, а также их отечественные аналоги GigaChat и YandexGPT являются одним из наиболее востребованных подходов для решения задач обработки текстов на естественном языке,

Исследователи [21] используют набор данных SemEval 2010 Task 8, сильно страдающий от дисбаланса классов и для категорий данных с меньшим количеством предложений генерируют 3 дополнительных предложения при помощи ChatGPT. Авторы также отмечают, что генерация при помощи ChatGPT могут возникать следующие проблемы: 1) потеря специальных символов, обозначающих начальную и конечную позицию элемента реляционной тройки, что вынуждает применять выравнивание; 2) сгенерированные данные могут не соответствовать ожидаемым методам дополнения, таким как трансляция, рекомбинация или замена объекта/субъекта; дополнительные данные могут иметь низкое качество. Учитывая эти проблемы, в работе [21] предлагается генерировать новые данные с использованием подсказки, описывающий формат выходных данных.

Авторы работы [19–21] исследуют, способна ли большая языковая модель самостоятельно извлекать временные отношения с минимальным внешним воздействием. Были предложены три различных запроса для оценки производительности с «нулевым выстрелом», эксперименты с которыми иллюстрируют, что производительность ChatGPT сильно зависит от контекста, играющего роль дополнительного знания об конкретном документе. Хотя, в [19–21] показывают, что ChatGPT еще довольно сильно отстает от современных контролируемых методов, он не подвержен сильному снижению производительности в мало-представленных классах.

Построение графов знаний не всегда может ограничиться поиском объекта субъекта и отношения. Особен-

но это заметно в научном мире. Где соединения определяются сложной комбинацией. Теоретически такого рода отношения можно смоделировать как  $n$ -кортежи, где  $n$  — количество сущностей, что для обычных методов, как правило, невозможно. Более того, один объект может иметь несколько одноименных значений. В статье [21] предлагают настраивать большую языковую модель GPT3 так, чтобы она принимала отрывок текста и писала точно отформатированное резюме, содержащееся в подсказке. Преимущество такого подхода — способность извлекать информацию как на уровне предложения, так и на уровне документа.

### Эксперимент по созданию вспомогательной системы

Авторами был проведен эксперимент, задача которого — определить, способна ли модель посредством переданного ей контекста установить связанные ли сущности конкретным отношением. В своем исследовании мы использовали русскоязычный публично доступный набор данных NEREL [22]. Он состоит из 746/94/93 документов Russian Wikinews, разделенных на обучающее, валидационное и тестовое множества соответственно. В общей сложности, в NEREL 56 тысяч аннотированных именованных объектов, разделенных на 29 типов, и 39 тысяч аннотированных отношений, разделенных на 49 типов.

В эксперименте использовалась русскоязычная LLM GigaChat, на вход которой подавался различный контекст. Ниже показан пример контекста, и как можно видеть, модель не только способна устанавливая взаимосвязь сущностей посредством некоторого, заведомо определенного отношения, но и преобразовывать ответ в требуемый формат.

Prompt:

{«role»: «user», «content»: «Тебе на вход будет передан текст. Необходимо выделить имена людей и их профессии в формат JSON.»},

{«role»: «user», «content»: «Президент Башкирии Муртаза Рахимов в очередной раз решил поменять главу своей администрации.»},

{«role»: «user», «content»: «Вчера он уволил Азамата Сагитова, который возглавил башкирскую администрацию год назад после вынужденной отставки Радия Хабирова, сейчас занимающего пост заместителя начальника управления президента РФ по внутренней политике.»}

Ответ модели:

{«content»: «{«Муртаза Рахимов»: «Президент Башкирии», «Азамат Сагитов»: «глава администрации», «Радий Хабиров»: «заместитель начальника управления президента РФ по внутренней политике»}»}



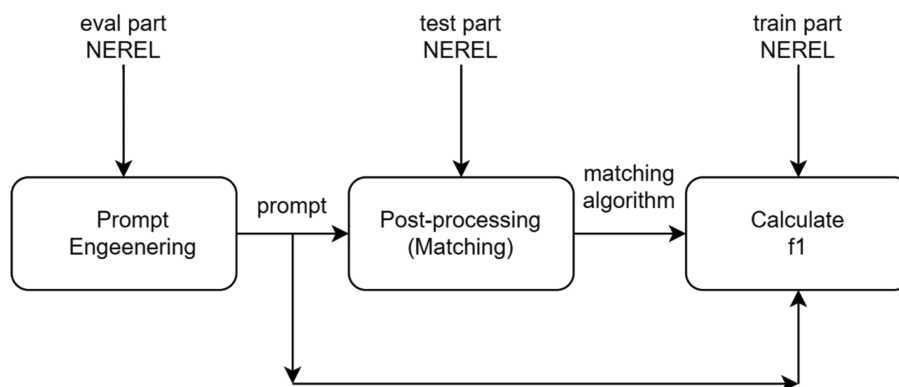


Рис. 1. Схема проведения эксперимента

Для численной оценки способности LLM устанавливать связь между сущностями, мы извлекли предложения в которых присутствовали реляционные тройки {PERSON; COUNTY; ORIGINS\_FROM}. И просили модель извлечь сущности, аффилированные отношением «родом из». Наша полная методология включала три стадии: 1 — подбор контекста для модели; 2 — обработка выходов модели, необходимая для матчинга ответов модели в разметку; 3 — оценка результатов матчинга. Для большей статистической значимости оценки мы меняли порядок разбиения данных, предложенный авторами NEREL. Для подборки контекста мы использовали валидационную часть, для подбора алгоритма постобработки тестовую, и финальное качество замеряли на тренировочной части. Общая схема проведения приведена на рисунке 1. Мы также удаляем примеры из замера, которые распознаны моделью как запрещенные, в соответствии с политикой, предоставляющей модель, компании.

В зависимости от выбора контекста и алгоритма постобработки наши оценки показывали качество от 0.5–0.65 f1 меры, что на наш взгляд, является достаточным для построения вспомогательной системы.

### Сравнительный анализ

По результатам описанных генеративных методов был проведен сравнительный анализ преимуществ и недостатков каждой технологии. Результаты сравнительного анализа отображены в таблице 1.

Метод prompt выделяется на фоне других подходов, так как не требует значительных объемов данных для обучения модели. Это свойство особенно важно в условиях ограниченных ресурсов или при использовании заранее обученных моделей, что позволяет сократить время на подготовку и обработку данных. В то время как методы seq-to-seq и seq-to-set требуют значительных объемов данных для тренировки и построения эффективных моделей, что может быть не всегда целесообразно при наличии ограничений по данным.

Таблица 1.

Сравнительный анализ генеративных методов

	seq-to-seq	seq-to-set	prompt
Точность	—	+	—
Низкие требования к базе данных	—	—	+
Высокий уровень влияния на работу системы	+	+	—
Низкий порог использования	—	—	+
Низкие временные затраты на реализацию	—	—	+
Автономность	+	+	—
Гибкость к количеству реляционных троек	+	—	+

Метод seq-to-seq показал большую гибкость в работе с реляционными тройками, что означает способность обрабатывать и учитывать большие объемы взаимосвязанных данных. Это свойство полезно в задачах, где важно учитывать сложные и динамичные структуры данных. В то время как методы seq-to-set и prompt менее эффективны в таких случаях, так как они более ориентированы на работу с отдельными элементами или запросами, а не на сложные взаимосвязи.

### Заключение

В работе был проведен аналитический обзор и сравнительный анализ существующих генеративных методов извлечения реляционных троек. Введена классификация генеративных методов, включающая три группы: классические методы преобразования последовательностей, методы преобразования последовательностей в множество, prompt-методы. Выделены ключевые преимущества и недостатки каждой группы генеративных подходов. Наибольшей перспективной, по мнению авторов, обладают подходы, основанные на использовании больших языковых моделей.

В следующих работах авторы планируют продолжить разработку методологии построения вспомогательной системы, реализующей комбинированную разметку, дополнив ее системой правил преобразования и сравне-

ния текстовых данных. Такая система в результате позволит воплощать самостоятельные системы извлечения реляционных троек в конкретных предметных областях.

## ЛИТЕРАТУРА

1. Kuzmenko A.V., Kireev V.S. A Review of Generative Methods for Extracting Relational Triples from Natural Language Texts // *Studies in Computational Intelligence*, 2024 Vol. 477 SCI, Q4 pp. 228–238
2. Кузьменко А. В., Киреев В. С. Классификация методов извлечения реляционных троек из текстов на естественном языке // *Материалы XXV Междунар. науч.-техн. конф. «Нейроинформатика-2023»*. — 2023. — С. 302–311.
3. Ambhore R. Governance of Unstructured Data: Managing Data Quality in Non-Traditional Data Sources // *International Journal of Research*. — 2024. — Vol. 11, No. 12. — P. 19–37.
4. Shang et al. Relational Triple Extraction: One Step is Enough // *Proc. 31st Int. Joint Conf. Artif. Intell. (IJCAI-22)*. — 2022. — P. 4360–4366.
5. Zenelko D., Aone C., Richardella A., et al. Kernel methods for relation extraction // *J. Mach. Learn. Res.* — 2003. — Vol. 3. — P. 1083–1106.
6. Chan S.Y., Roth D. Exploiting syntactico-semantic structures for relation extraction // *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics: Human Lang. Technol.* — 2011. — P. 551–560.
7. Zhong Z., Chen D.A. A frustratingly easy approach for entity and relation extraction // *Proc. 2021 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.* — 2021. DOI: 10.18653/v1/2021.naacl-main-5.
8. Miwa M., Sasaki Y. Modeling Joint Entity and Relation Extraction with Table Representation // *Proc. 2014 Conf. Empirical Methods in Natural Lang. Process. (EMNLP)*. — 2014. — P. 1858–1869.
9. Gupta P., Schütze H., Andrassy B. Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction // *Proc. COLING 2016, 26th Int. Conf. Comput. Linguistics: Tech. Papers*. — 2016. — P. 2537–2547.
10. Zhang M., Zhang Y., Fu G. End-to-End Neural Relation Extraction with Global Optimization // *Proc. 2017 Conf. Empirical Methods in Natural Lang. Process.* — 2017. — P. 1730–1740.
11. Miwa M., Bansal M. End-to-end Relation Extraction using LSTMs on Sequences and Tree Structures // *Proc. 54th Annu. Meet. Assoc. Comput. Linguistics*. — 2016. — Vol. 1. — P. 1105–1116.
12. Zheng S., Wang F., Bao H., et al. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme // *Proc. 55th Annu. Meet. Assoc. Comput. Linguistics*. — 2017. — Vol. 1. — P. 1227–1236.
13. Dai D., Xiao X., Lyu Y., et al. Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling // *Proc. AAAI Conf. Artif. Intell.* — 2019. — P. 6300–6308.
14. Wei Z., Su J., Wang Y., et al. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction // *Proc. 58th Annu. Meet. Assoc. Comput. Linguistics*. — 2020. — P. 1476–1488.
15. Zeng X., Zeng D., He S., et al. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism // *Proc. 56th Annu. Meet. Assoc. Comput. Linguistics*. — 2018. — Vol. 1. — P. 506–514.
16. Zeng D., Zhang H., Liu Q. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning // *Proc. AAAI Conf. Artif. Intell.* — 2020. — P. 9507–9514.
17. Sui D., Chen Y., Liu K., et al. Joint entity and relation extraction with set prediction networks // *IEEE Trans. Neural Networks Learn. Syst.* — 2023. DOI: 10.1109/TNNLS.2023.3264735.
18. Wen X., Zhou C., Tang H., et al. End-to-end entity detection with proposer and regressor // *Neural Process. Lett.* — 2023. DOI: 10.1007/s111063-023-11201-8.
19. Hu Y., Ameer I., Zou X. Zero-shot Clinical Entity Recognition using ChatGPT. — 2024. DOI: arXiv:2303.16416.
20. Yuan C., Xie Q., Ananiadou S. Zero-shot Temporal Relation Extraction with ChatGPT // *Proc. 22nd Workshop Biomed. Nat. Lang. Process. BioNLP Shared Tasks*. — 2023. DOI: 10.48550/arXiv.2304.05454.
21. Feng P., Wu H., Yang Z., et al. Leveraging Prompt and Top-K Predictions with ChatGPT Data Augmentation for Improved Relation Extraction // *Appl. Sci.* — 2023. — Vol. 13, № 23. Dagdelen J., Dunn A., Lee S., et al.
22. Artemova E., Malykh V., Tyers F., et al. NEREL: A Russian Dataset with Nested Named Entities, Relations and Events // *Proc. Recent Adv. Nat. Lang. Process.* — 2021. DOI: 10.26615/978-954-452-072-4\_100.